EFM32PG28 MCU Reference Manual

The EFM32PG28 family of microcontrollers is part of the Series 2 Gecko portfolio. EFM32PG28 MCUs are ideal for enabling energy-friendly embedded applications.

The highly efficient solution contains a 80 MHz Cortex-M33 with rich analog and communication peripherals to provide an industry-leading, energy efficient MCU for consumer and industrial applications.

EFM32PG28 applications include:

- Metering
- Industrial Automation
- Appliances
- Portable Medical Devices

KEY FEATURES

- 32-bit ARM Cortex®-M33 core with 80 MHz maximum operating frequency
- Up to 1024 kB of flash and 256 kB of RAM
- Robust peripheral set and up to 51 GPIO
- 16-channel ADC with options for 12, 16, or 20-bit resolution
- Best-in-class security with Secure Vault™
- AI/ML Hardware Accelerator

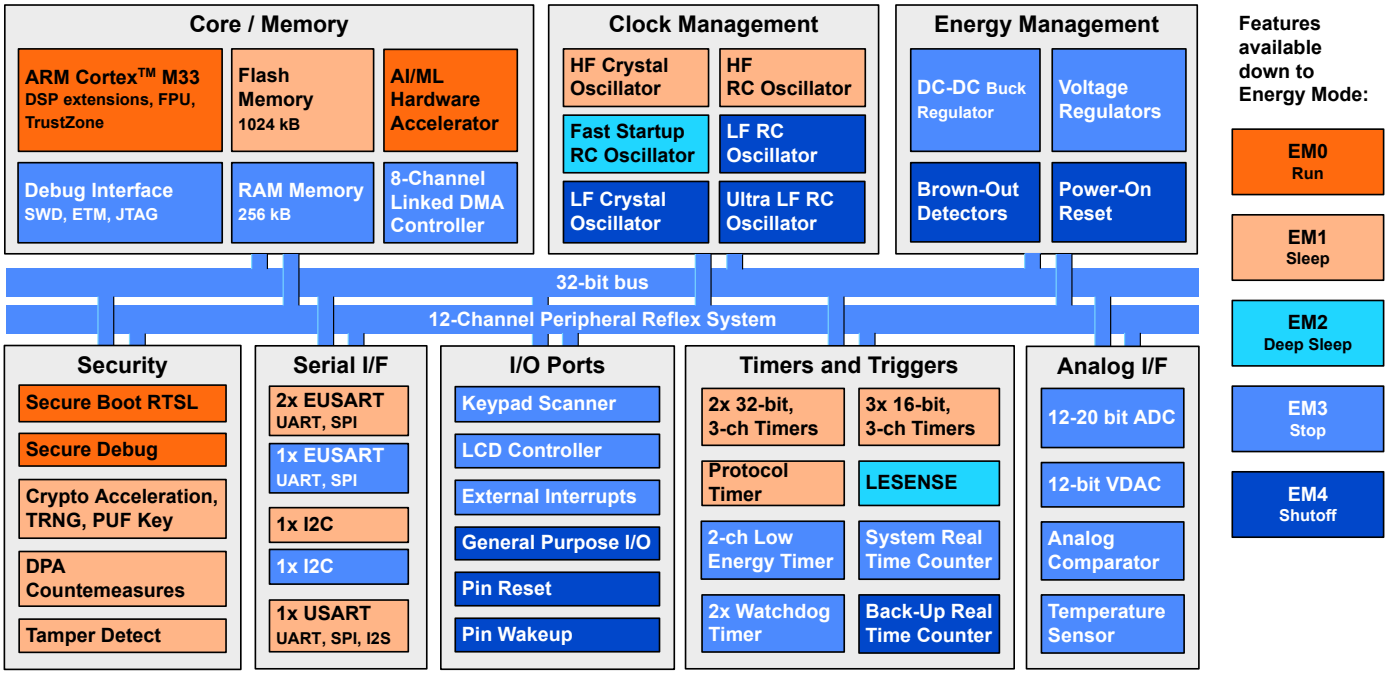


Table of Contents

1. About This Document

1.1 Introduction

This document contains reference material for the EFM32PG28 devices. All modules and peripherals in the EFM32PG28 devices are described in general terms. Not all modules are present in all devices and the feature set for each device might vary. Such differences, including pinout, are covered in the device data sheets.

1.2 Conventions

Register Names

Register names are given with a module name prefix followed by the short register name:

TIMERn_CTRL - Control Register

The "n" denotes the module number for modules which can exist in more than one instance.

Some registers are grouped which leads to a group name following the module prefix:

GPIO_Px_DOUT - Port Data Out Register

The "x" denotes the different ports.

Bit Fields

Registers contain one or more bit fields which can be 1 to 32 bits wide. Bit fields wider than 1 bit are given with start (x) and stop (y) bit [y:x].

Bit fields containing more than one bit are unsigned integers unless otherwise is specified.

Unspecified bit field settings must not be used, as this may lead to unpredictable behaviour.

Address

The address for each register can be found by adding the base address of the module found in the Memory Map (see Figure 4.1 System Address Space with Core and Code Space Listing on page 41), and the offset address for the register (found in module Register Map).

Access Type

The register access types used in the register descriptions are explained in Table 1.1 Register Access Types on page 30.

Table 1.1. Register Access Types

Access Type	Description
R	Read only. Writes are ignored
RW	Readable and writable
RW1	Readable and writable. Only writes to 1 have effect
W1	Read value undefined. Only writes to 1 have effect
W	Write only. Read value undefined.
RWH	Readable, writable, and updated by hardware
RW(nB), RWH(nB), etc.	"(nB)" suffix indicates that a bitfield explicitly does not support peripheral bit set/clear/toggle operations (see 4. Memory and Bus System)
RW(r), R(r), etc.	"(r)" suffix indicates that reading the register causes an action and may alter the register value.

Number format

0x prefix is used for hexadecimal numbers

0b prefix is used for binary numbers

Numbers without prefix are in decimal representation.

Reserved

Registers and bit fields marked with **reserved** are reserved for future use. These should be written to their reset value unless otherwise stated in the Register Description. Read values for reserved bits may be different in future or prior devices.

Reset Value

The reset value denotes the value after reset.

Registers denoted with X have unknown value out of reset and need to be initialized before use. Note that read-modify-write operations on these registers before they are initialized results in undefined register values.

Pin Connections

Pin connections are given with a module prefix followed by a short pin name:

CMU_CLKOUT1 (Clock management unit, clock output pin number 1)

The location for the pin names given in the module documentation can be found in the device-specific datasheet.

1.3 Related Documentation

Further documentation on the EFM32PG28 devices and the ARM Cortex®-M33 can be found at the Silicon Labs and ARM web pages:

www.silabs.com

www.arm.com

2. System Overview

2.1 Introduction

The EFM32 MCUs are the world's most energy friendly microcontrollers. With a unique combination of the powerful 32-bit ARM Cortex-M33, innovative low energy techniques, short wake-up time from energy saving modes, and a wide selection of peripherals, the EFM32PG23 microcontroller is well suited for any battery operated application as well as other systems requiring high performance and low-energy consumption.

2.2 Block Diagrams

The block diagram for the EFM32PG28 series is shown in (Figure 2.1 EFM32PG28 System-On-Chip Block Diagram on page 32).

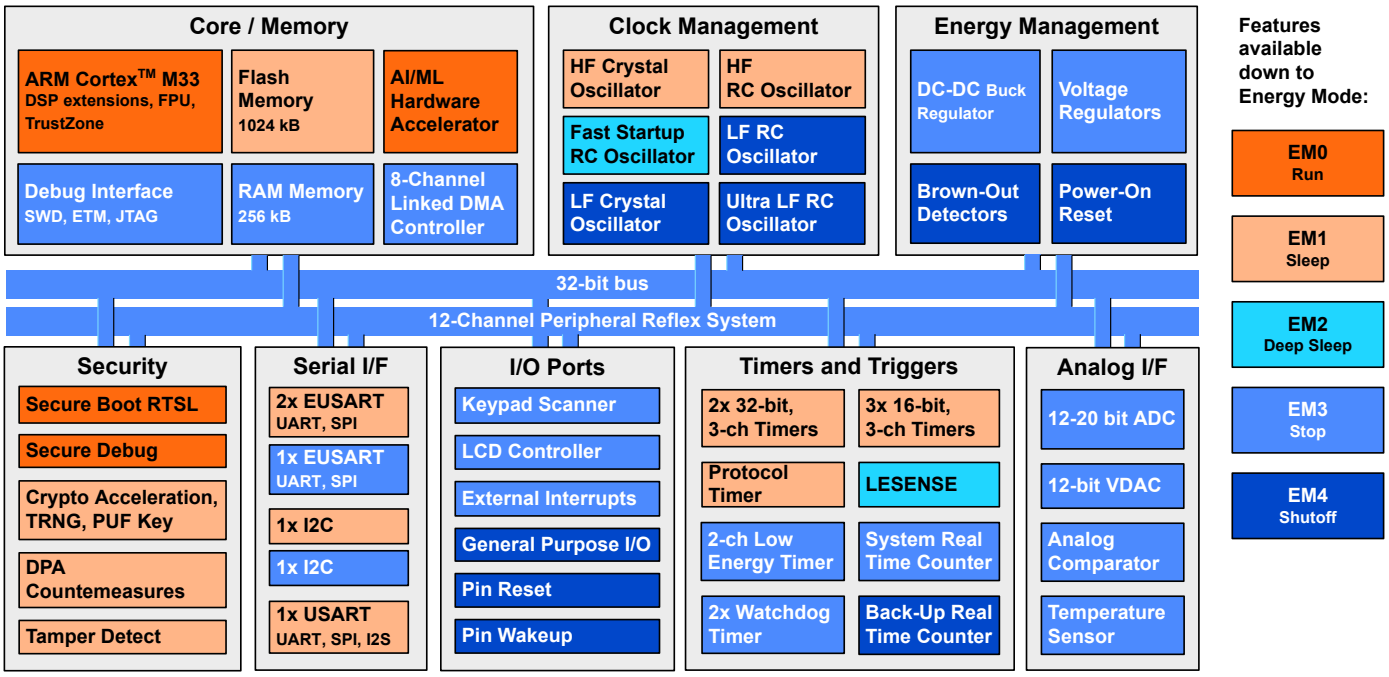


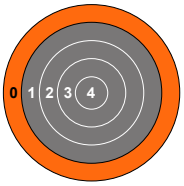
Figure 2.1. EFM32PG28 System-On-Chip Block Diagram

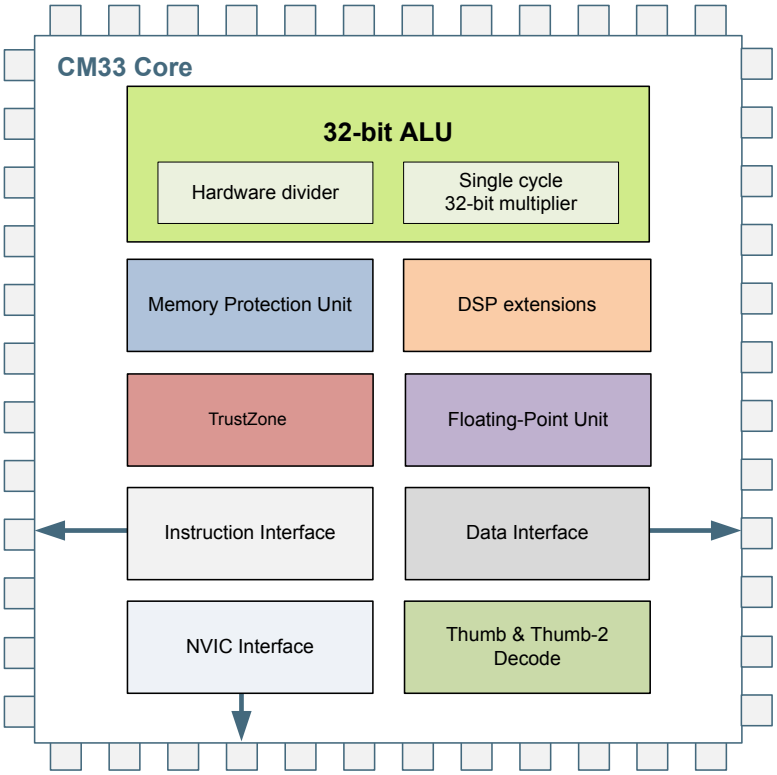
2.3 Features overview

- **ARM Cortex®-M33 CPU platform**
  - High Performance 32-bit processor @ up to 80 MHz
  - DSP instruction support and floating-point unit
  - Memory Protection Unit
  - Wake-up Interrupt Controller
- **Matrix Vector Processor (MVP)**
  - Tailored for AI / ML applications, complex floating-point matrixed multiplication and addition
  - Instruction set architecture with ALU, loop, and load/store instructions
  - ALU has 32-bit floating point input operands, supports real and complex numbers, single-cycle operation
  - Pipelined load/store unit with direct DMA bus managers supporting two simultaneous 32-bit reads and one 32-bit write per cycle
- **Flexible Energy Management System**
  - Five Energy Modes from EM0 to EM4 provide flexibility between higher performance and low power
  - Power routing configurations including DCDC control
  - Voltage Monitoring and Brown Out Detection
  - Automatic voltage scaling for additional energy savings
  - State Retention
- **Up to 1024 kB Flash**
- **Up to 256 kB RAM**
- **Up to 49 General Purpose I/O pins**
  - Configurable push-pull, open-drain, pull-up/down, input filter, slew rate
  - Configurable peripheral I/O locations
  - 16 asynchronous external interrupts
  - Output state retention and wake-up from Shutoff Mode
- **8 Channel DMA Controller**
  - Alternate/primary descriptors with scatter-gather/ping-pong operation
- **16 Channel Peripheral Reflex System (PRS)**
  - Autonomous inter-peripheral signaling enables smart operation in low energy modes
  - 12 asynchronous channels with configurable logic functionality
  - 4 synchronous channels for high-speed signalling between TIMER and IADC
- **General Purpose Cyclic Redundancy Check (GPCRC)**
  - Programmable 16-bit polynomial, fixed 32-bit polynomial
- **Communication interfaces**
  - 1 × Universal Synchronous/Asynchronous Receiver/Transmitter (USART)
    - UART/SPI/SmartCard (ISO 7816)/IrDA/I2S
    - Triple buffered full/half-duplex operation
    - Hardware flow control
    - 4-16 data bits
  - 3 × Enhanced Universal Synchronous/Asynchronous Receiver/Transmitter (EUSART)
    - UART/SPI/IrDA support
    - High-speed operation in EM0/1 using high-frequency clock source
    - One instance with low-energy operation in EM2 using 32.768 kHz clock source
    - Buffered full/half-duplex operation
    - Hardware flow control
    - 7/8/9 data bits
  - 2 × I^2C Interface (I2C) with SMBus support
    - Address recognition in EM3 Stop Mode

- **Timers/Counters**
  - 1 × 32-bit and 4 × 16-bit Timer/Counters (TIMER)
    - 3 Compare/Capture/PWM channels
    - Dead-Time Insertion
  - 24-bit Low Energy Timer (LETIMER)
  - 32-bit Real-Time Counter (SYSRTC)
  - 32-bit Ultra Low Energy Backup Real Time Counter (BURTC) for periodic wake-up from any Energy Mode
  - 16-bit Pulse Counter
    - Asynchronous pulse counting/quadrature decoding
  - 2 × Watchdog Timer (WDOG)
- **Ultra low power precision analog peripherals**
  - Incremental Analog to Digital Converter (IADC) with 12-bit resolution at 1 Msps and 16-bit resolution at 76.9 ksps
    - Single ended or differential operation
    - Conversion tailgating for predictable latency
  - 12-bit 500 ksps Digital to Analog Converter (VDAC)
    - 2 single ended channels/1 differential channel
  - 2 × Analog Comparator (ACMP)
    - Programmable speed/current
  - Analog Bus (ABUS) signal routing
  - Accurate die temperature sensor
  - External thermistor interface
- **Low-energy sensor interface**
  - Autonomous sensor monitoring in deep sleep mode
  - Wide range of supported sensors including LC sensors
- **Low-energy keypad scanner**
  - Up to 6 x 8 key switches supported
  - Autonomous keypad scanning in EM0 / EM1
  - Wake on key press from EM2 / EM3
- • **Integrated LCD Controller**
    - Up to 8 x 24 (192) or 4 x 28 (112) segments
    - Voltage boost, contrast and autonomous animation
    - Patented low-energy LCD driver
- **Ultra efficient Power-on Reset (POR) and Brown-Out Detector (BOD)**
- **Debug Interface**
  - 4-pin Joint Test Action Group (JTAG) interface
  - 2-pin serial-wire debug (SWD) interface
  - Embedded Trace (ETM) interface with 4 data lines
- **Security**
  - Secure Boot with Root of Trust and Secure Loader (RTSL)
    - Prevents malware injection and rollback
    - Ensures authentic firmware execution and OTA updates
  - Dedicated Secure Core
    - Delivers faster, more energy efficient hardware crypto with Differential Power Analysis (DPA) countermeasures for AES128/256, SHA-1, SHA-2 (up to 256-bit), ECC (up to 256-bit), ECDSA, ECDH and J-Pake
    - Provides isolation with the application core
    - Provides hardware cryptographic acceleration
    - True Random Number Generator (TRNG) compliant with NIST SP800-90 and AIS-31
  - ARM® TrustZone®
  - Secure Debug with lock/unlock
    - Allows authenticated access for enhanced Failure Analysis (FA)

3. System Processor





Quick Facts

What?

The EFM32PG28 features the industry leading Cortex®-M33 CPU from ARM.

Why?

The ARM Cortex®-M33 is designed for exceptionally short response time, high code density, and high 32-bit throughput while maintaining a strict cost and power consumption budget.

How?

Combined with the ultra low energy peripherals available in EFM32PG28 devices, the Cortex®-M33 processor's Harvard architecture, 3 stage pipeline, single cycle instructions, Thumb-2 instruction set support, and fast interrupt handling make it perfect for 8-bit, 16-bit, and 32-bit applications.

3.1 Introduction

The ARM Cortex®-M33 32-bit RISC processor provides outstanding computational performance and exceptional system response to interrupts while meeting low cost requirements and low power consumption.

The ARM Cortex®-M33 implemented is revision r0p4.

3.2 Features

- Harvard architecture
  - Separate data and program memory buses (No memory bottleneck as in a single bus system)
- 3-stage pipeline
- Thumb-2 instruction set
  - Enhanced levels of performance, energy efficiency, and code density
- Single cycle multiply and hardware divide instructions
  - 32-bit multiplication in a single cycle
  - Signed and unsigned divide operations between 2 and 11 cycles
- 1.5 DMIPS/MHz
- TrustZone
  - Independent Secure and Privileged states
  - Accelerated context switching
- 16 Region MPU
- 24-bit System Tick Timer for Real Time OS
- Excellent 32-bit migration choice for 8/16 bit architecture based designs
  - Simplified stack-based programmer's model is compatible with traditional ARM architecture and retains the programming simplicity of legacy 8-bit and 16-bit architectures
- Aligned or unaligned data storage and access
  - Contiguous storage of data requiring different byte lengths
  - Data access in a single core access cycle
- Integrated power modes
  - Sleep Now mode for immediate transfer to low power state
  - Sleep on Exit mode for entry into low power state after the servicing of an interrupt
  - Ability to extend power savings to other system components
- Optimized for low latency, nested interrupts

3.3 Functional Description

For a full functional description of the ARM Cortex®-M33 implementation in the EFM32PG28 family, the reader is referred to the ARM Cortex®-M33 documentation.

3.3.1 Interrupt Operation

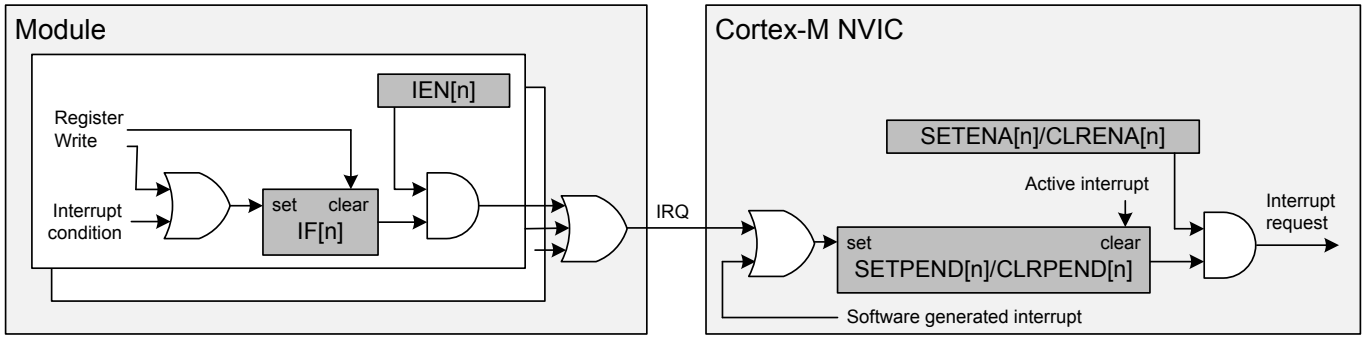


Figure 3.1. Interrupt Operation

The interrupt request (IRQ) lines are connected to the Cortex®-M33. Each of these lines (shown in 3.3.3 Interrupt Request Lines (IRQ)) is connected to one or more interrupt flags in one or more modules. The interrupt flags are set by hardware on an interrupt condition. It is also possible to set/clear the interrupt flags through the IF register interface. When setting or clearing an interrupt through the IF register use of the IF_SET or IF_CLR bit operation registers is required; directly writing the main interrupt flag register will have no effect.

Each interrupt flag is then qualified with its own interrupt enable bit (IEN register), before being OR'ed with the other interrupt flags to generate the IRQ. A high IRQ line will set the corresponding pending bit (can also be set/cleared with the SETPEND/CLRPEND bits in ISPRn/ICPRn) in the Cortex®-M33 NVIC. The pending bit is then qualified with an enable bit (set/cleared with SETENA/CLRENA bits in ISERn/ICERn) before generating an interrupt request to the core. Figure 3.1 Interrupt Operation on page 37 illustrates the interrupt system. For more information on how the interrupts are handled inside the Cortex®-M33, the reader is referred to the **ARM Cortex-M33 Processor Technical Reference Manual**.

3.3.1.1 Avoiding Extraneous Interrupts

There can be latencies in the system such that clearing an interrupt flag could take longer than leaving an Interrupt Service Routine (ISR). This can lead to the ISR being re-entered as the interrupt flag has yet to clear immediately after leaving the ISR. To avoid this, when clearing an interrupt flag at the end of an ISR, the user should execute ARM's Data Synchronization Barrier (DSB) instruction. Another approach is to clear the interrupt flag immediately after identifying the interrupt source and then service the interrupt as shown in the pseudo-code below. The ISR typically is sufficiently long to more than cover the few cycles it may take to clear the interrupt status, and also allows the status to be checked for further interrupts before exiting the ISR.

```
irqXServiceRoutine() {
   do {
      clearIrqXStatus();
      serviceIrqX();
   } while(irqXStatusIsActive());
}
```

3.3.2 TrustZone

The Cortex®-M33 implements ARM TrustZone which provides the ability to restrict access to peripherals and memory regions based on the CPU security attribute. TrustZone works in combination which the MPU which controls privileged/unprivileged execution of code to provide a full security solution. The Security Management Unit (SMU) is used to configure access restrictions in the various modes. Refer to 9. SMU - Security Management Unit for more information.

For information about TrustZone features in the core or information on TrustZone specific instructions please see the ARM Cortex-M33 Processor Technical Reference Manual provided by ARM

3.3.3 Interrupt Request Lines (IRQ)

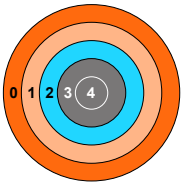
This table shows all IRQ's for the system processor. M33 High Speed interrupts are indicated by an '*'.

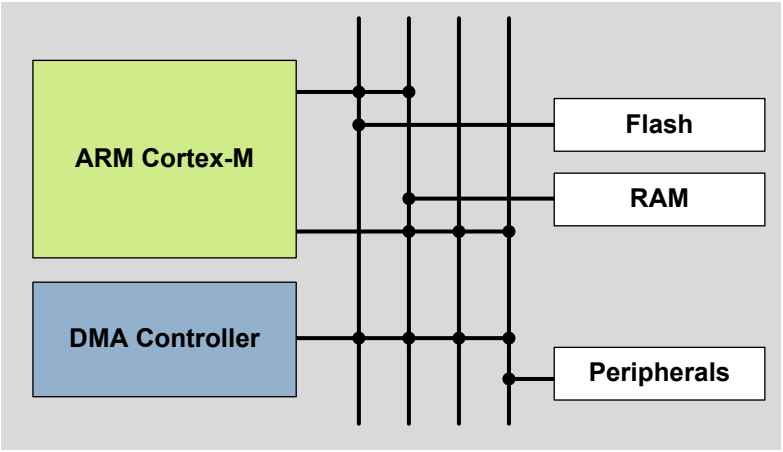
See the individual peripheral chapters for more information on interrupt function.

IRQ #	Name
0*	SMU_SECURE
1*	SMU_S_PRIVILEGED
2*	SMU_NS_PRIVILEGED
3*	EMU
4*	TIMER0
5*	TIMER1
6*	TIMER2
7*	TIMER3
8*	TIMER4
9*	USART0_RX
10*	USART0_TX
11*	EUSART0_RX
12*	EUSART0_TX
13*	EUSART1_RX
14*	EUSART1_TX
15*	EUSART2_RX
16*	EUSART2_TX
17*	ICACHE0
18*	BURTC
19*	LETIMER0
20*	SYSCFG
21*	MPAHBRAM
22*	LDMA
23*	LFXO
24*	LFRCO
25*	ULFRCO
26*	GPIO_ODD
27*	GPIO_EVEN
28*	I2C0
29*	I2C1
30*	EMUDG
39*	HOSTMAILBOX
41*	ACMP0
42*	ACMP1

IRQ #	Name
43*	WDOG0
44*	WDOG1
45*	HFXO0
46*	HFRCO0
47*	HFRCOEM23
48*	CMU
50*	IADC
51*	MSC
52*	DPLL0
53*	EMUEFP
54*	DCDC
55*	VDAC0
56*	PCNT0
57*	SW0
58*	SW1
59*	SW2
60*	SW3
61*	KERNEL0
62*	KERNEL1
63*	M33CTI0
64*	M33CTI1
65*	FPUEXH
66*	SETAMPERHOST
67*	SEMBRX
68*	SEMBTX
69*	LESENSE
70*	SYSRTC_APP
71*	SYSRTC_SEQ
72*	LCD
73*	KEYSCAN
76*	AHB2AHB0
77*	AHB2AHB1
78*	MVP

4. Memory and Bus System





Quick Facts

What?

A low latency memory system including low energy Flash and RAM with data retention which makes the low energy modes attractive.

Why?

RAM retention reduces the need for storing data in Flash and enables frequent use of the ultra low energy modes EM2 and EM3.

How?

Low energy and non-volatile Flash memory stores program and application data in all energy modes and can easily be reprogrammed in system. Low leakage RAM with data retention in EM0 to EM3 removes the data restore time penalty, and the DMA ensures fast autonomous transfers with predictable response time.

4.1 Introduction

The EFM32PG28 contains a set of AMBA buses which move data between peripherals, memory, and the CPU. All memories and register interfaces are memory mapped into a unified address space.

4.2 Functional Description

The internal memory segments of the Cortex®-M33 are mapped into the system memory map as shown by Figure 4.1 System Address Space with Core and Code Space Listing on page 41.

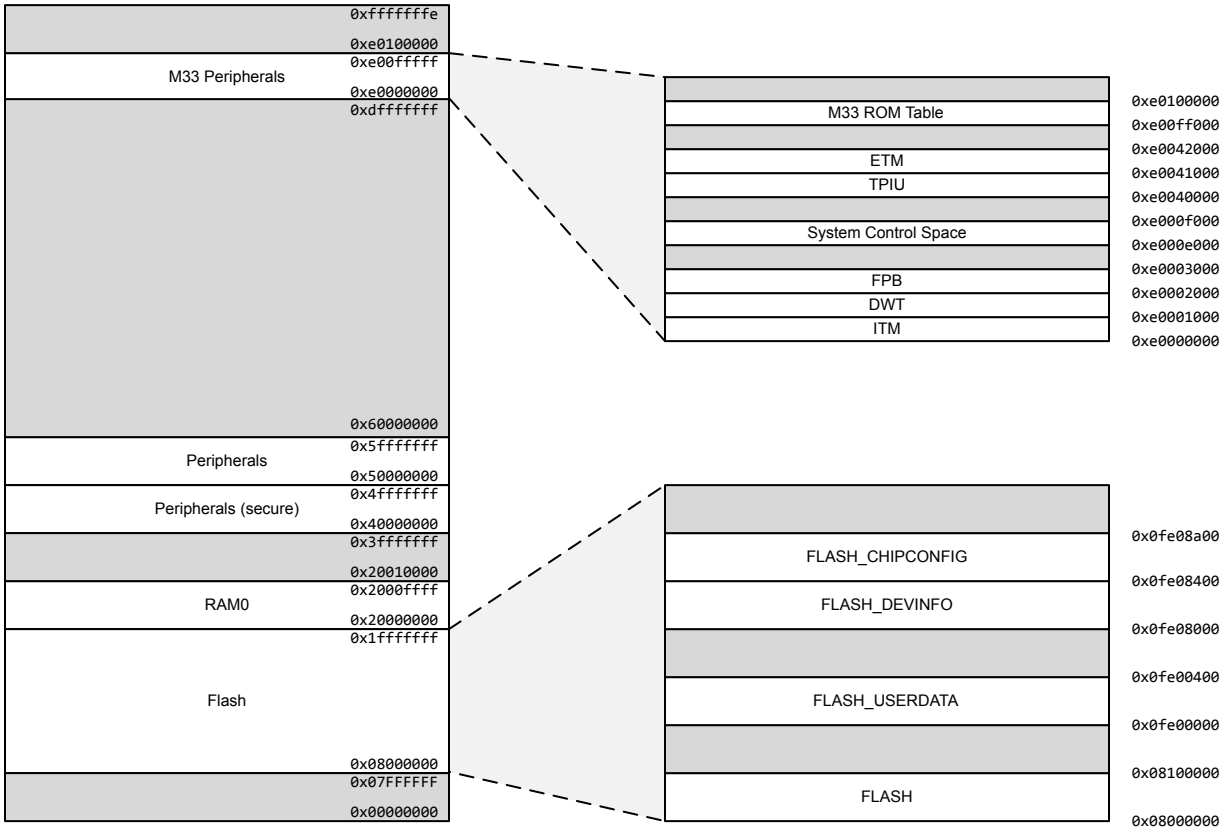


Figure 4.1. System Address Space with Core and Code Space Listing

Flash for the main program memory (CODE) is located at address 0x08000000 in the memory map of the EFM32PG28. Flash memory also contains a USERDATA area intended for user-defined data storage, the DEVINFO space with device characteristics and identifying information, and CHIPCONFIG with internal production test and calibration information.

SRAM for the main data memory (RAM) is located at address 0x20000000 in the memory map of the EFM32PG28. When running code located in RAM, the Cortex®-M33 uses the System bus interface to fetch instructions. This results in reduced performance as the Cortex®-M33 accesses stack, other data in SRAM and peripherals using the System bus interface.

The Sequencer RAM (SEQRAM) is located at address 0xA0000000 and is used by the Sequencer for both instructions and data. This RAM is also available for general use if not required by the RF subsystem.

4.2.1 Bus Matrix

A multilayer AMBA AHB bus matrix connects the manager bus interfaces to the AHB subordinates. The bus matrix allows several AHB subordinates to be accessed simultaneously. An AMBA APB interface is used for the peripherals, which are accessed through an AHB-to-APB bridge connected to the AHB bus matrix.

The CPU has two AHB bus managers (Code and System) so that it may retrieve instructions and data in parallel. The Code manager is used to access all memory below address 0x20000000 and the System manager access addresses 0x20000000 and above.

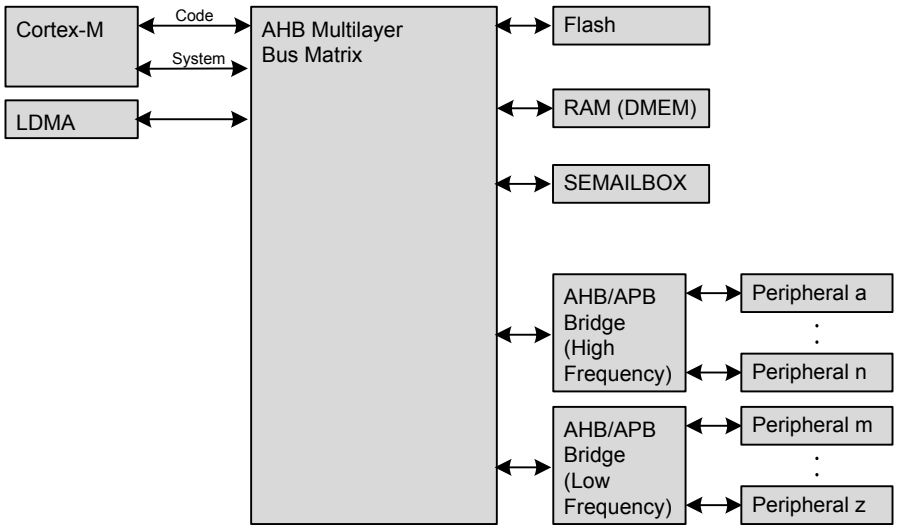


Figure 4.2. EFM32PG28 Bus System

4.2.1.1 Arbitration

The Bus Matrix uses a round-robin arbitration algorithm which enables high throughput and low latency, while starvation of simultaneous accesses to the same bus subordinate are eliminated. Round-robin does not assign a fixed priority to each bus manager. The arbiter does not insert any bus wait-states during peak interaction. However, one wait state is inserted for manager accesses occurring after a prolonged inactive time. This wait state allows for increased power efficiency during manager idle time.

4.2.1.2 Bus Faults

System accesses from the core can receive a bus fault in the following condition(s):

- The core attempts to access an address that is not assigned to any peripheral or other system device. These faults can be enabled or disabled by setting the ADDRFAULTEN bit in the SYSCFG_CTRL register.
- The core attempts to access a peripheral register that is LOCKED.
- The core attempts to access a peripheral or system device that has its clock disabled. This fault can be enabled or disabled by setting the ADDRFAULTEN bit in the SYSCFG_CTRL register.
- System RAM controller detects a 2bit ECC error. These faults can be enabled or disabled by setting the RAMECCERRFAULTEN bit in the SYSCFG_CTRL register.
- Registers with synchronization requirements may generate bus faults if accessed incorrectly. See 4.2.4.4 Peripheral Access Performance for more details on register access types. In particular the following actions can cause bus faults:
  - Config register written while peripheral enabled.
  - Sync register written while peripheral disabled.
  - LfSync register written while a previous write is pending.
  - Peripheral disabled while any LfSync write is pending.
  - Peripheral registers written or module re-enabled while DISABLING is set.
  - Peripheral registers other than write-only fields or SWRST read while RESETTING is set.
  - Peripheral registers other than read-only fields (including SWRST) written while RESETTING is set.

In addition to any condition-specific bus fault control bits, the bus fault interrupt itself can be enabled or disabled in the same way as all other internal core interrupts.

4.2.2 Flash

The Flash retains data in any state and typically stores the application code and special user data. The Flash memory is typically programmed through the debug interface, but can also be erased and written to from software.

- Up to 1024 kB of memory
- Page size of 8 KB (minimum erase unit)
- Lock registers for memory protection
- Data retention in any state

4.2.3 SRAM

The primary task of the SRAM memory is to store application data. Additionally, it is possible to execute instructions from SRAM, and the DMA may be set up to transfer data between the SRAM, Flash and peripherals.

The device contains several blocks of SRAM in data memory (DRAM) space. For more detailed information see 5. MSC - Memory System Controller .

- Up to 256 kB of memory (RAM)
- RAM blocks may be powered down when not in use
- Data retention of the entire memory or selected banks in EM2 and EM3

4.2.4 Peripherals

The peripherals are mapped into the peripheral memory segment, each with a fixed size address range shown in the 4.2.4.1 Peripheral Map

4.2.4.1 Peripheral Map

This table shows the address range for each peripheral. In addition it shows the lowest energy mode in which the peripheral is powered. Note that EM3 is defined as EM2 with all clocks disabled. Therefore all peripherals powered in EM2 are also powered in EM3 but may not function if they require a running clock.

See the individual peripheral chapters for more information on low power operation.

Address Range	Module Name	Power Domain
0x40000000 - 0x40003FFF	SCRATCHPAD	EM1
0x40004000 - 0x40007FFF	EMU	EM2 (PD0A)
0x40008000 - 0x4000BFFF	CMU	EM2 (PD0B)
0x40010000 - 0x40013FFF	HFRCO0	EM1
0x40018000 - 0x4001BFFF	FSRCO	EM2 (PD0A)
0x4001C000 - 0x4001FFFF	DPLL0	EM1
0x40020000 - 0x40023FFF	LFXO	EM4
0x40024000 - 0x40027FFF	LFRCO	EM4
0x40028000 - 0x4002BFFF	ULFRCO	EM4
0x40030000 - 0x40033FFF	MSC	EM1
0x40034000 - 0x40037FFF	ICACHE0	EM1
0x40038000 - 0x4003BFFF	PRS	EM2 (PD0E)
0x4003C000 - 0x4003FFFF	GPIO	EM2 (PD0B)
0x40040000 - 0x40043FFF	LDMA	EM1
0x40044000 - 0x40047FFF	LDMAXBAR	EM1
0x40048000 - 0x4004BFFF	TIMER0	EM1
0x4004C000 - 0x4004FFFF	TIMER1	EM1
0x40050000 - 0x40053FFF	TIMER2	EM1
0x40054000 - 0x40057FFF	TIMER3	EM1
0x40058000 - 0x4005BFFF	TIMER4	EM1
0x4005C000 - 0x4005FFFF	USART0	EM1
0x40064000 - 0x40067FFF	BURTC	EM4
0x40068000 - 0x4006BFFF	I2C1	EM1
0x40078000 - 0x4007BFFF	SYSCFG	EM1
0x4007C000 - 0x4007FFFF	SYSCFG	EM1
0x40080000 - 0x40083FFF	BURAM	EM4
0x40088000 - 0x4008BFFF	GPCRC	EM1
0x40094000 - 0x40097FFF	DCDC	EM2 (PD0B)
0x40098000 - 0x4009BFFF	HOSTMAILBOX	EM1
0x400A0000 - 0x400A3FFF	EUSART1	EM1
0x400A4000 - 0x400A7FFF	EUSART2	EM1
0x400A8000 - 0x400ABFFF	SYSRTC0	EM2 (PD0A)
0x400AC000 - 0x400AFFFF	LCD	EM2 (PD0A)

Address Range	Module Name	Power Domain
0x400B0000 - 0x400B3FFF	KEYSCAN	EM2 (PD0E)
0x400B4000 - 0x400B7FFF	DMEM	EM1
0x44008000 - 0x4400BFFF	SMU	EM1
0x4400C000 - 0x4400FFFF	SMU	EM1
0x49000000 - 0x49003FFF	LETIMER0	EM2 (PD0B)
0x49004000 - 0x49007FFF	IADC0	EM2 (PD0B)
0x49008000 - 0x4900BFFF	ACMP0	EM2 (PD0B)
0x4900C000 - 0x4900FFFF	ACMP1	EM2 (PD0B)
0x49024000 - 0x49027FFF	VDAC0	EM2 (PD0B)
0x49030000 - 0x49033FFF	PCNT0	EM2 (PD0B)
0x49038000 - 0x4903BFFF	LESENSE	EM2 (PD0B)
0x4A000000 - 0x4A003FFF	HFRCOEM23	EM2 (PD0C)
0x4A004000 - 0x4A007FFF	HFXO0	EM2 (PD0C)
0x4B000000 - 0x4B003FFF	I2C0	EM2, (PD0D)
0x4B004000 - 0x4B007FFF	WDOG0	EM2, (PD0D)
0x4B008000 - 0x4B00BFFF	WDOG1	EM2, (PD0D)
0x4B010000 - 0x4B013FFF	EUSART0	EM2, (PD0D)
0x4C000000 - 0x4C00007F	SEMAILBOX	EM1
0x4D000000 - 0x4D003FFF	MVP	EM1
0x50000000 - 0x50003FFF	SCRATCHPAD_NS	EM1
0x50004000 - 0x50007FFF	EMU_NS	EM2 (PD0A)
0x50008000 - 0x5000BFFF	CMU_NS	EM2 (PD0B)
0x50010000 - 0x50013FFF	HFRCO0_NS	EM1
0x50018000 - 0x5001BFFF	FSRCO_NS	EM2 (PD0A)
0x5001C000 - 0x5001FFFF	DPLL0_NS	EM1
0x50020000 - 0x50023FFF	LFXO_NS	EM4
0x50024000 - 0x50027FFF	LFRCO_NS	EM4
0x50028000 - 0x5002BFFF	ULFRCO_NS	EM4
0x50030000 - 0x50033FFF	MSC_NS	EM1
0x50034000 - 0x50037FFF	ICACHE0_NS	EM1
0x50038000 - 0x5003BFFF	PRS_NS	EM2 (PD0E)
0x5003C000 - 0x5003FFFF	GPIO_NS	EM2 (PD0B)
0x50040000 - 0x50043FFF	LDMA_NS	EM1
0x50044000 - 0x50047FFF	LDMAXBAR_NS	EM1
0x50048000 - 0x5004BFFF	TIMER0_NS	EM1
0x5004C000 - 0x5004FFFF	TIMER1_NS	EM1
0x50050000 - 0x50053FFF	TIMER2_NS	EM1

Address Range	Module Name	Power Domain
0x50054000 - 0x50057FFF	TIMER3_NS	EM1
0x50058000 - 0x5005BFFF	TIMER4_NS	EM1
0x5005C000 - 0x5005FFFF	USART0_NS	EM1
0x50064000 - 0x50067FFF	BURTC_NS	EM4
0x50068000 - 0x5006BFFF	I2C1_NS	EM1
0x50078000 - 0x5007BFFF	SYSCFG_NS	EM1
0x5007C000 - 0x5007FFFF	SYSCFG_NS	EM1
0x50080000 - 0x50083FFF	BURAM_NS	EM4
0x50088000 - 0x5008BFFF	GPCRC_NS	EM1
0x50094000 - 0x50097FFF	DCDC_NS	EM2 (PD0B)
0x50098000 - 0x5009BFFF	HOSTMAILBOX_NS	EM1
0x500A0000 - 0x500A3FFF	EUSART1_NS	EM1
0x500A4000 - 0x500A7FFF	EUSART2_NS	EM1
0x500A8000 - 0x500ABFFF	SYSRTC0_NS	EM2 (PD0A)
0x500AC000 - 0x500AFFFF	LCD_NS	EM2 (PD0A)
0x500B0000 - 0x500B3FFF	KEYSCAN_NS	EM2 (PD0E)
0x500B4000 - 0x500B7FFF	DMEM_NS	EM1
0x54008000 - 0x5400BFFF	SMU_NS	EM1
0x5400C000 - 0x5400FFFF	SMU_NS	EM1
0x59000000 - 0x59003FFF	LETIMER0_NS	EM2 (PD0B)
0x59004000 - 0x59007FFF	IADC0_NS	EM2 (PD0B)
0x59008000 - 0x5900BFFF	ACMP0_NS	EM2 (PD0B)
0x5900C000 - 0x5900FFFF	ACMP1_NS	EM2 (PD0B)
0x59024000 - 0x59027FFF	VDAC0_NS	EM2 (PD0B)
0x59030000 - 0x59033FFF	PCNT0_NS	EM2 (PD0B)
0x59038000 - 0x5903BFFF	LESENSE_NS	EM2 (PD0B)
0x5A000000 - 0x5A003FFF	HFRCOEM23_NS	EM2 (PD0C)
0x5A004000 - 0x5A007FFF	HFXO0_NS	EM2 (PD0C)
0x5B000000 - 0x5B003FFF	I2C0_NS	EM2, (PD0D)
0x5B004000 - 0x5B007FFF	WDOG0_NS	EM2, (PD0D)
0x5B008000 - 0x5B00BFFF	WDOG1_NS	EM2, (PD0D)
0x5B010000 - 0x5B013FFF	EUSART0_NS	EM2, (PD0D)
0x5C000000 - 0x5C00007F	SEMAILBOX_NS	EM1
0x5D000000 - 0x5D003FFF	MVP_NS	EM1

Note:

1. Peripherals listed as being in EM2 (PD0A) always remain powered in EM2 and EM3. Other EM2 power domains (PD0B, PD0C, etc.) are powered down in EM2 and EM3 if all peripherals on that domain are unused.

4.2.4.2 Peripheral non-word access behavior

When writing to peripheral registers, all accesses are treated as 32-bit accesses. This means that writes to a register need to be large enough to cover all bits of register, otherwise, any uncovered bits may become corrupted from the partial-word transfer. Thus, the safest practice is to always do 32-bit writes to peripheral registers.

When reading, there is generally no issue with partial word accesses, however, note that any read action (e.g. FIFO popping) will be triggered regardless of whether the actual FIFO bit-field was included in the transfer size.

4.2.4.3 Peripheral Bit Set and Clear

The EFM32PG28 supports bit set, bit clear, and bit toggle access to most peripheral registers. The bit set and bit clear functionality (also called Bit Access) enables modification of bit fields without the need to perform a read-modify-write. Also, the operation is contained within a single bus access. Bit access registers and their addresses are shown in the register map for each peripheral. Peripherals with no _SET, _CLR, or _TGL registers in the register map to not support these functions.

Each register with Bit Set functionality will have a _SET register. Whenever a bit in the SET register is written to a 1 the corresponding bit in its target register is set. The SET register is located at TARGET + 0x1000 where TARGET is the address of the target register and has the same name as the target register with '_SET' appended.

Each register with Bit Clear functionality will have a CLR register. Whenever a bit in the CLR register is written to a 1 the corresponding bit in its target register is cleared. The CLR register is located at TARGET + 0x2000 where TARGET is the address of the target register and has the same name as the target register with '_CLR' appended.

Each register with Bit Toggle functionality will have a TGL register. Whenever a bit in the TGL register is written to a 1 the corresponding bit in its target register is inverted. The TGL register is located at TARGET + 0x3000 where TARGET is the address of the target register and has the same name as the target register with '_TGL' appended.

Note: It is possible to combine bit clear and bit set operations in order to arbitrarily modify multi-bit register fields without affecting other fields in the same register. In this case, care should be taken to ensure that the field does not have intermediate values that can lead to erroneous behavior. For example, if bit clear and bit set operations are used to change an analog tuning register field from 0x2 to 0x4 by clearing bit 1 and then setting bit 2, the field would take on a value of zero for short time. If the analog module is active at the time, this could lead to undesired behavior.

4.2.4.4 Peripheral Access Performance

The Cortex®-M33, DMA Controller, and peripherals run on clocks which can be pre-scaled separately. Clocks and pre-scaling are described in more detail in 7. CMU - Clock Management Unit. This section describes the access performance for a peripheral register based on its frequency relative to the CPUCLK frequency and its access type. For this discussion, PERCLK refers to a selected peripheral's clock frequency and CPUCLK refers to the core's clock frequency.

The type of each register in a peripheral is indicated in the 'Access' column of the peripherals register table. Register types are: ENABLE, CONFIG, SYNC, LFSYNC, and INTFLAG. If not type is listed then the register is a Generic register.

4.2.4.4.1 Generic Registers

Registers with no type listed are generic registers. They may be read or written to at any time. Access will not stall the CPU.

4.2.4.4.2 CONFIG Registers

CONFIG Registers contain configuration that does not change during peripheral operation.

CONFIG registers may only be written when a peripheral is disabled. Writing to a CONFIG register when a peripheral is enabled will result in a BUSFAULT. CONFIG register writes will not stall the CPU.

CONFIG registers may be read at any time. Reads will not stall the CPU.

4.2.4.4.3 SYNC Registers

SYNC registers are used to communicate with running high-speed peripherals where PERCLK is expected to be either higher or marginally slower (within an order of magnitude) than CPUCLK. For example a timer running at 78 MHz when the core is at 39 MHz or at 9.75 MHz when the core is 78 MHz. In this case CPU stalls of several PERCLK cycles do not significantly impact overall system performance in most systems.

SYNC registers may only be written to when the peripheral is enabled. Writing to a SYNC register when a peripheral is disabled will result in a BUSFAULT. A write will take several (2 - 3) PERCLK cycles to complete (take effect) during which time the entire module will be in a pending state. If a SYNC register is written to while the peripheral is already in a pending state, the CPU is stalled until the previous write finishes. If a SYNC register is written to while the peripheral is not in a pending state, the CPU is not stalled.

SYNC registers may be read at any time. If a SYNC register is read while the peripheral is disabled, the CPU is not stalled. If a SYNC register is read while the peripheral is enabled, the CPU will be stalled for several (2 -3) PERCLK cycles while up to date values are retrieved from the peripheral.

4.2.4.4.4 LFSYNC Registers

LFSYNC registers are used to communicate with running low frequency peripherals where PERCLK is expected to be much lower than the CPU clock and synchronization delays may be long. For example, a LETIMER running at 32 kHz when the core is at 78 MHz. In this case CPU stalls of several PERCLK cycles represent a significant blockage of the CPU and need to be avoided whenever possible. LFSYNC registers accommodate this by allowing the CPU to write the register and continue to do other work while the value is synchronized.

LFSYNC registers may be written at any time. A write will take several (3 -4) PERCLK cycles to complete during which the register will be in a pending state. If a LFSYNC register is written to while it is in a pending state, a BUSFAULT will occur. Each LFSYNC register has a status bit indicating if it is currently pending.

LFSYNC registers may be read at any time. The CPU is never stalled on a read. If a LFSYNC register is read while pending, the pending (recently written) data will be returned even though it has not yet taken effect. Software may use the busy status bit to determine if the read value has been applied to the hardware.

4.2.4.4.5 ENABLE Registers

ENABLE registers contain the enable bit for a peripheral.

ENABLE registers may be written at any time. When the peripheral is enabled it takes some time for the enable to take effect during which time the module is pending. Peripherals will be in the pending state for a few (2 - 3) PERCLK cycles when first enabled. Since the clock source for the peripheral may not be running before the peripheral is enabled, the start up time for the clock source may increase the pending time. See 7. CMU - Clock Management Unit for more information on on-demand clock sources.

When EN is cleared to 0, the peripheral will be disabled. The DISABLING status bit will be set to 1 until the operation is complete. During disablement, the module will wait for any SYNCBUSY status to clear, reset the core peripheral function, and de-assert the peripheral clock. Entry into low energy modes EM2 and EM3 will be delayed while a peripheral is disabling.

While DISABLING is set, writing to any register in the module, including attempts to re-enable with EN, will cause a bus fault condition. Any register in the module can be read while DISABLING.

4.2.4.4.6 SWRST Registers

SWRST registers are available in some blocks to reset the module back to the intial condition, similar to a power-on reset.

SWRST registers have a SWRST bit, which will reset the peripheral when written to 1. These registers also contain a status bit, RESETTING, which indicates that a reset is in progress.

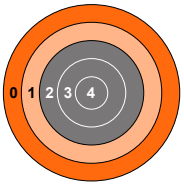
If a peripheral is resetting due to SWRST, entry into low energy modes EM2 and EM3 will be delayed. Writing to the SWRST bit or any other writeable register while a reset is in progress will generate a bus fault condition.

4.2.4.4.7 INTFLAG Registers

INTFLAG registers contain interrupt flags. To set or clear an interrupt flag, the _SET or _CLR register alias must be used. Writing directly to the INTFLAG register will have no effect.

Note that for an interrupt to occur when a flag is set the IRQ must be enabled in the NVIC.

5. MSC - Memory System Controller



01000101011011100110010101110010
01100111011110010010000001001101
01101001011000110111001001101111
00100000011100100111010101101100
01100101011100110010000001110100
01101000011001010010000001110111
01101111011100100110110001100100
00100000011011110110011000100000
01101100011011110111011100101101
01100101011011100110010101110010
01100111011110010010000001101101
01101001011000110111001001101111
01100011011011110110111001110100
01110010011011110110110001101100
01100101011100100010000001100100
01100101011100110110100101100111
01101110001000010100010101101110

Quick Facts

What?

The user can perform flash memory read, read configuration, and write operations through the Memory System Controller (MSC). SRAM operation may be configured though System Configuration (SYSCFG).

Why?

The MSC allows the application code and user data to be stored in non-volatile flash memory. Certain memory system functions, such as program memory wait-states and flash lock bits are configured from the MSC peripheral register interface, giving the developer the ability to dynamically customize the memory system performance, security level, energy consumption and error handling capabilities to the requirements at hand.

How?

The MSC integrates a low-energy flash IP with a charge pump, enabling minimum energy consumption while eliminating the need for external programming voltage to erase the memory. An easy to use write and erase interface is supported by an internal, fixed-frequency oscillator and autonomous flash timing and control reduces software complexity while not using other timer resources.

A highly efficient low energy instruction cache reduces the number of flash reads significantly, thus saving energy. Performance is also improved when wait-states are used, since many of the wait-states are eliminated. Built-in performance counters can be used to measure the efficiency of the instruction cache.

Instruction prefetcher improves program execution performance by reducing the number of wait-state cycles needed.

5.1 Introduction

The Memory System Controller (MSC) is the program memory unit of the EFM32PG28 microcontroller. The flash memory is readable and writable from both the Cortex®-M33 and DMA. The flash memory is divided into two blocks: the main block and the information block. Program code is normally written to the main block. The information block is available for special user data. There is also a read-only page in the information block containing system and device calibration data. Flash read and write operations are supported in energy modes EM0 and EM1.

5.2 Features

- AHB read interface
  - Scalable access performance to optimize the Cortex®-M33 code interface
    - Advanced energy optimization functionality
      - Conditional branch target prefetch suppression
      - Cortex®-M33 unfolding of if-then (IT) blocks
      - Instruction Cache
      - Instruction Prefetch
  - DMA read support in EM0 and EM1
- Command and status interface
  - Flash write and erase
    - Accessible from Cortex®-M33 in EM0
    - DMA write support in EM0 and EM1
  - Core clock independent flash timing
    - Internal oscillator and internal timers for precise and autonomous flash timing
      - General purpose timers are not occupied during flash erase and write operations
      - No special time scaling registers needed
  - Configurable interrupt erase abort
    - Improved interrupt predictability
  - Memory and bus fault control
- Security features
  - Lockable debug access
  - Page lock registers
  - SW Mass erase and User Data lock bits
- End-of-write and end-of-erase interrupts

5.3 Functional Description

The size of the main flash block is device dependent. The largest size available is 1024 kB (128 pages). The information block has 1 kB available for user data. The information block also contains chip configuration data located in a reserved area. The main block is mapped to address 0x08000000 and the information block is mapped to address 0x0FE00000. Table 5.1 MSC Flash Memory Mapping on page 50 outlines how the flash is mapped in the memory space. All flash memory is organized into 8 kB pages.

Table 5.1. MSC Flash Memory Mapping

Block	Page	Base address	Write/Erase by...	Software Readable?	Purpose/Name	Size
Main	0	0x08000000	Software, debug	Yes	User code and data	16 kB - 1024 kB
	1	0x08002000	Software, debug	Yes		
	...		Software, debug	Yes		
	127[1]	0x080FE000	Software, debug	Yes		
Information	N/A	0x0FE00000	Software, debug	Yes	User Data (UD)	1 kB
	N/A	0x0FE08000	-	Yes	Device Information (DI)	1 kB

Note:

1. 128 pages for largest device.

5.3.1 RAM Configuration

The SYSCFG and MPAHBRAM modules contain controls for configuring the various RAM blocks on the device. Options include enabling EM2/EM3 data retention, ECC, and RAM port priorities. For a complete description see 5.6 SYSCFG - System Configuration.

5.3.2 Instruction Cache

The instruction cache improves the speed and power consumption of the Cortex®-M33 by providing fast, low-power access to recently executed instructions. For detailed information see 5.5 ICACHE - Instruction Cache

5.3.3 Device Information (DI) Page

This read-only page holds calibration data from the production test, several unique device IDs, and other part specific information. For a complete description see 5.4 DEVINFO - Device Info Page.

5.3.4 User Data (UD) Page Description

This is the user data page in the information block. The page can be erased and written by software when MISCLOCKWORD.UDLOCKBIT is 0.

5.3.5 Bootloader

The EFM32PG28 supports use of the Gecko Bootloader detailed in *UG489: Silicon Labs Gecko Bootloader User's Guide for GSDK 4.0 and Higher* (https://www.silabs.com/support/resources).To enable bootloader functionality, the second stage of the bootloader must be configured and programmed into the beginning of flash. The first stage of the bootloader is provided by the SE and is not user accessible. For more details on SE bootloader support, see the SE peripheral documentation.

5.3.6 Post-reset Behavior

Calibration values are automatically written to registers by the MSC before application code start-up. The values can also be read from the DI page by software. Other information such as the device ID and production date is also stored in the DI page and is readable from software.

As part of the reset, hardware performs repeated flash reads to determine when flash is fully powered up and available for use by the CPU. PWRUPCKBDFAILCOUNT in MSC_STATUS contains the number of failed reads during the last reset.

5.3.7 Flash Startup

Flash wakeup on demand is supported when waking from EM2/3 to EM0. Set bit FLASHPWRUPONDEMAND of register EMU_CTRL to enable the power up on demand. When enabled, flash will not be powered up until accessed. In this case it is possible for the MCU to wake, execute out of RAM or cache, and return to sleep mode without ever powering on the flash. Software can force the flash to power up by writing PWRUP in MSC_CMD. When flash is powered via MSC_CMD the MSC_IF.PWRUPF interrupt flag will be set when power up is complete and the CPU will be interrupted if MSC_IEN.PWRUPF is set.

5.3.8 Flash EM0 / EM1 Power Down

It is also possible to put the flash in a power-saving sleep mode when the system is in EM0 or EM1. Flash power down can be configured to happen on entering EM1 or with an immediate manual operation.

During EM0, software can instruct the flash to go to power down mode with the MSC_CMD.PWROFF command. Any system IRQ or flash read will wake the flash. The MSC_CMD.PWRUP command is used to power the flash back up in the absence of a wake event.

The MSC_PWRCTRL register allows the flash to be configured to automatically enter sleep mode on entering EM1 sleep with the bits PWROFFONEM1ENTRY and PWROFFONEM1PENTRY, respectively. If the flash is configured to sleep during one of these states, it may sometimes be powered back up without processor intervention in EM0 (for example, if DMA reads flash in EM1). By default, the flash will remain powered on after such access. If the PWROFFENTRYAGAIN bit is set, it instructs the flash to re-enter the power down state if no further access is seen during the timeout period defined by PWROFFDLY. Flash must be idle for PWROFFDLY * 64 bus clocks before it will enter sleep again.

5.3.9 Wait-states

Since the CPU may be clocked faster than the flash can respond, it is necessary to configure wait-states for flash accesses at higher CPU clock speeds. See the device Datasheet for information on the maximum allowed frequency for each wait-state setting. To configure the flash wait-states set the MODE field in MSC_READCTRL.

When changing wait states, care should be taken that the system is never in an invalid state. To ensure this, MODE should be changed after the clock is changed when reducing clock speed and before the clock is changed when increasing clock speed.

5.3.10 Cortex®-M33 If-Then Block Folding

The Cortex®-M33 offers a mechanism known as if-then block folding. This is a form of speculative prefetching where small if-then blocks are collapsed in the prefetch buffer if the condition evaluates to false. The instructions in the block then appear to execute in zero cycles. With this scheme, performance is optimized at the cost of higher energy consumption as the processor fetches more instructions from memory than it actually executes. To disable the mode, write a 1 to the DISFOLD bit in the NVIC Auxiliary Control Register; see the Cortex®-M33 Technical Reference Manual for details. Normally, it is expected that this feature is most efficient when operating with 0 wait-states. Folding is enabled by default.

5.3.11 Line Buffering (Prefetch)

The MSC reads a 2-word line from flash on any flash access. The data being accessed is returned immediately and the other word locally cached so that it can be provided immediately if accessed. This has the effect of pre-fetching the second word when the first is read, resulting in fewer wait-states when executing sequential code. This feature may be disabled by setting DOUTBUFEN in MSC_READCTRL.

5.3.12 Erase and Write Operations

To erase a page first set WREN in MSC_WRITECTRL and load any address in the page to be erased into the MSC_ADDRB register. Next check INVADDR, LOCKED, and WREADY in MSC_STATUS to ensure that the address is valid, not locked, and the MSC is ready to modify flash. Writing ERASEPAGE in MSC_WRITEMD will execute the page erase operation. ERASE in MSC_IF will be set when the page erase is complete. If ERASE in MSC_IEN is set, the end of a page erase will also trigger an interrupt. Finally, clear WREN to disable flash operations.

In addition to a page erase, a mass erase will clear the entire contents of the main flash array. A mass erase can be initiated by the Secure Engine. User Data page contents are not included in a mass erase.

To perform a programming operation, set WREN and load the address to be programmed into the MSC_ADDRB register. Next check INVADDR, LOCKED, WREADY, and WDATAREADY in MSC_STATUS to ensure that the address is valid, not locked, the MSC is ready to modify flash, and the write data buffer is clear. Writing data to MSC_WDATA will begin the programming operation. If a burst write is being performed, the next data word can be programmed to MSC_WDATA as soon as WDATAREADY is set. WRITE in MSC_IF will be set when the programming operation is complete. If WRITE in MSC_IEN is set, the end of the program operation will also trigger an interrupt. Finally, clear WREN to disable flash operations.

If data is written to the MSC_WDATA register faster than it can be processed, WDATAOV in MSC_IF will be set. If WDATAOV in MSC_IEN is set an interrupt will also be fired.

The MSC_ADDRB register only has to be written once when writing to sequential words. After each word is written, ADDRB is incremented automatically by 4. The INVADDR bit of the MSC_STATUS register is set if the loaded address is outside the flash. The LOCKED bit of the MSC_STATUS register is set if the page addressed is locked. Any attempts to erase or write to the page are ignored if INVADDR or the LOCKED bits of the MSC_STATUS register are set.

Write and erase operations may be aborted by software. To abort an erase, set the ERASEABORT bit in the MSC_WRITECMD register. To abort a write, set WRITEEND in MSC_WRITECMD

For a DMA write, CLEARWDATA in MSC_WRITECMD to assert a DMA request and transfer the first word. Alternately the first word may be programmed manually into MSC_WDATA by code.

By default, if any interrupt occurs during an erase operation, the erase is aborted. This feature may be disabled by clearing IRQERASEABORT in MSC_WRITECTRL. When an erase is aborted due to an interrupt, ERASEABORTED in MSC_STAUTS is set by hardware.

Software may observe the status of the MSC via the MSC_STATUS register. When a flash operation is in progress, BUSY will be set. If a flash operation has been requested but not yet started, PENDING will be set. This may occur if a subsystem is performing MSC operations. When the write buffer underflows, TIMEOUT will be set. Buffer underflow is a normal part of the write procedure since it will occur once the last word has been written and no more data is available.

The flash memory is organized into 64-bit wide double-words. Each 64-bit double-word can be written only twice between erase cycles. The lower and upper 32-bit words may be written sequentially in any order, or one at a time. Each flash bit is 1 after erase. Writing a 0 will clear the bit. Writing a 1 will not change the bit value.

While it is possible to write twice to the lower or upper 32-bit word of the 64-bit double word, then the other 32-bit word cannot be used. In this case, it is permitted to write to either the lower or upper 32-bit word twice between each erase, so long as no bit is ever cleared more than once.

Note: The ERASEPAGE bit in WRITECMD and the WDATA register cannot safely be written from code in flash. It is recommended to place a small code section in RAM to set these bits and wait for the operation to complete. Also note that DMA transfers to or from any other address in flash while a write or erase operation is in progress will produce unpredictable results.

Note: During a write or erase, flash read accesses will be stalled, effectively halting code execution from flash. Code execution continues upon write/erase completion. Code residing in RAM or ICACHE may be executed during a write/erase operation.

5.3.12.1 Low-Power Write

To limit maximum current, the programming operations can be slowed down. Set LPWRITE in MSC_WRITECTRL to double the write time and halve the write current.

5.3.12.2 Flash Lock

The ability to program or erase individual flash pages may be disabled using the MSC_PAGELOCKn registers. The bits in these registers may only be set to 1 by software on the device and are cleared when the device is reset. This means that once locked, a page may not be unlocked until a reset occurs. Users wishing to lock accesses to flash should implement code to write to the MSC_PAGELOCKn registers immediately after a reset. Any page locked in this way cannot be written to or erased.

User Data can be locked by setting MSC_MISCLOCKWORD.UDLOCK to 1. Mass erase is enabled out of reset, however if firmware sets MELOCKBIT in the MSC_MISCLOCKWORD register, then mass erase can only be issued by the SE.

5.4 DEVINFO - Device Info Page

The Device Info Page holds factory programmed information about the device. It contains the following data:

- Calibration values for reconfiguring the device
- Unique ID's
- OPN identifiers (family, feature set, flash size, etc.)

5.4.1 DEVINFO Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	DEVINFO_INFO	R	DI Information
0x004	DEVINFO_PART	R	Part Info
0x008	DEVINFO_MEMINFO	R	Memory Info
0x00C	DEVINFO_MSIZE	R	Memory Size
0x010	DEVINFO_PKGINFO	R	Misc Device Info
0x014	DEVINFO_CUSTOMINFO	R	Custom Part Info
0x018	DEVINFO_SWFIX	R	SW Fix Register
0x01C	DEVINFO_SWCAPA0	R	Software Restriction
0x020	DEVINFO_SWCAPA1	R	Software Restriction
0x028	DEVINFO_EXTINFO	R	External Component Info
0x040	DEVINFO_EUI48L	R	EUI 48 Low
0x044	DEVINFO_EUI48H	R	EUI 48 High
0x048	DEVINFO_EUI64L	R	EUI64 Low
0x04C	DEVINFO_EUI64H	R	EUI64 High
0x050	DEVINFO_CALTEMP	R	Calibration Temperature
0x054	DEVINFO_EMUTEMP	R	EMU Temp
0x058	DEVINFO_HFRCODPLLCALn	R	HFRCODPLL Calibration
0x0A0	DEVINFO_HFRCOEM23CALn	R	HFRCOEM23 Calibration
0x130	DEVINFO_MODULENAME0	R	Module Name Information
0x134	DEVINFO_MODULENAME1	R	Module Name Information
0x138	DEVINFO_MODULENAME2	R	Module Name Information
0x13C	DEVINFO_MODULENAME3	R	Module Name Information
0x140	DEVINFO_MODULENAME4	R	Module Name Information
0x144	DEVINFO_MODULENAME5	R	Module Name Information
0x148	DEVINFO_MODULENAME6	R	Module Name Information
0x14C	DEVINFO_MODULEINFO	R	Module Information
0x150	DEVINFO_MODXOCAL	R	Module External Oscillator Calibration Information
0x17C	DEVINFO_HFXOCAL	R	High Frequency Crystal Oscillator Calibration Data
0x180	DEVINFO_IADC0GAIN0	R	IADC Gain Calibration
0x184	DEVINFO_IADC0GAIN1	R	IADC Gain Calibration
0x188	DEVINFO_IADC0OFFSETCAL0	R	IADC Offset Calibration
0x18C	DEVINFO_IADC0NORMALOFF-SETCAL0	R	IADC Offset Calibration
0x190	DEVINFO_IADC0NORMALOFF-SETCAL1	R	IADC Offset Calibration

Offset	Name	Type	Description
0x194	DEVINFO_IADC0HISPDOFF-SETCAL0	R	IADC Offset Calibration
0x198	DEVINFO_IADC0HISPDOFF-SETCAL1	R	IADC Offset Calibration
0x1FC	DEVINFO_LEGACY	R	Legacy Device Info
0x25C	DEVINFO_RTHERM	R	

5.4.2 DEVINFO Register Description

5.4.2.1 DEVINFO_INFO - DI Information

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xC								0x0								0x0															
Access	R								R								R															
Name	DEVINFOREV								PRODREV								CRC															

Bit	Name	Reset	Access	Description
31:24	DEVINFOREV	0xC	R	**DI Page Version**
	DEVINFO layout revision as unsigned integer (initially 1)			
23:16	PRODREV	0x0	R	**Production Revision**
	Production revision as unsigned integer			
15:0	CRC	0x0	R	**CRC**
	CRC of DI-page (CRC-16-CCITT)			

5.4.2.2 DEVINFO_PART - Part Info

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset			0x0								0x0						0x0															
Access			R								R						R															
Name			FAMILY								FAMILYNUM						DEVICENUM															

Bit	Name	Reset	Access	Description
31:30	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
29:24	FAMILY	0x0	R	**Device Family**
	Encoded portion of the Device Family			

Value	Mode	Description
0	FG	Flex Gecko
3	ZG	Z-Wave Gecko
5	PG	Pearl Gecko
8	SG	Sidewalk Gecko

23:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:16	FAMILYNUM	0x0	R	**Device Family**
	Numeric portion of the Device Family			
15:0	DEVICENUM	0x0	R	**Device Number**
	Device Number. The device number is one letter and 3 digits. NUMBER = (alpha-'A')*1000 + numeric. 0 = A000; 1123 = B123			

5.4.2.3 DEVINFO_MEMINFO - Memory Info

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																0x0								0x0							
Access	R																R								R							
Name	DILEN																UDPAGESIZE								FLASHPAGESIZE							

Bit	Name	Reset	Access	Description
31:16	DILEN	0x0	R	**Length of DI Page**
	Length of DI area (number of 32-bit words included in CRC)			
15:8	UDPAGESIZE	0x0	R	**User Data Page Size**
	User Data page size			
7:0	FLASHPAGESIZE	0x0	R	**Flash Page Size**
	Flash page size in bytes coded as 2^((MEMINFO.PAGESIZE +10) & 0xFF. For example, the value of 0xFF = 512 bytes			

5.4.2.4 DEVINFO_MSIZE - Memory Size

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset						0x0											0x0															
Access						R											R															
Name						SRAM											FLASH															

Bit	Name	Reset	Access	Description
31:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26:16	SRAM	0x0	R	**Sram Size**
	Ram size, kbyte count as unsighed integer (eg 16)			
15:0	FLASH	0x0	R	**Flash Size**
	Flash size, kbyte count as unsigned integer (eg. 128)			

5.4.2.5 DEVINFO_PKGINFO - Misc Device Info

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0								0x0								0x0							
Access									R								R								R							
Name									PINCOUNT								PKGTYPE								TEMPGRADE							

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:16	PINCOUNT	0x0	R	**Pin Count**
	Device pin count as unsigned integer (eg. 48)			
15:8	PKGTYPE	0x0	R	**Package Type**
	Package identifier as character			

Value	Mode	Description
74	WLCSP	WLCSP package
76	BGA	BGA package
77	QFN	QFN package
81	QFP	QFP package

Bit	Name	Reset	Access	Description
7:0	TEMPGRADE	0x0	R	**Temperature Grade**
	Temperature Grade of product as unsigned integer enumeration			

Value	Mode	Description
0	N40TO85	-40 to 85 degC
1	N40TO125	-40 to 125 degC
2	N40TO105	-40 to 105 degC
3	N0TO70	0 to 70 degC

5.4.2.6 DEVINFO_CUSTOMINFO - Custom Part Info

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	PARTNO																															

Bit	Name	Reset	Access	Description
31:16	PARTNO	0x0	R	**Part Number**
	Custom part identifier as unsigned integer (eg. 903). 65535 for standard product			
15:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

5.4.2.7 DEVINFO_SWFIX - SW Fix Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFFFF																															
Access	R																															
Name	RSV																															

Bit	Name	Reset	Access	Description
31:0	RSV	0xFFFFFFFF	R	**Reserved**
	Reserved for future use			

5.4.2.8 DEVINFO_SWCAPA0 - Software Restriction

Offset	Bit Position																											
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset						0x0					0x0				0x0				0x0				0x0				0x0				0x0	
Access						R					R				R				R				R				R				R	
Name						ZWAVE					SRI				CONNECT				BTSMART				RF4CE				THREAD				ZIGBEE	

Bit	Name	Reset	Access	Description
31:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26:24	ZWAVE	0x0	R	**Z-Wave Capability**
	Z-Wave Stack Capability			

Value	Mode	Description
0	LEVEL0	Z-Wave stack capability not available
1	LEVEL1	Z-Wave Gateway
2	LEVEL2	Z-Wave End Device
3	LEVEL3	Z-Wave Sensor
4	LEVEL4	Z-Wave Lighting

Bit	Name	Reset	Access	Description
23:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:20	SRI	0x0	R	**RAIL Capability**
	RAIL capability not available			

Value	Mode	Description
0	LEVEL0	RAIL capability not available
1	LEVEL1	RAIL enabled
2	LEVEL2	N/A
3	LEVEL3	N/A

Bit	Name	Reset	Access	Description
19:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17:16	CONNECT	0x0	R	**Connect Capability**
	Connect stack capability level			

Value	Mode	Description
0	LEVEL0	Connect stack capability not available
1	LEVEL1	Connect enabled
2	LEVEL2	N/A
3	LEVEL3	N/A

Bit	Name	Reset	Access	Description
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:12	BTSMART	0x0	R	**Bluetooth Smart Capability**
	Bluetooth SMART stack capability level			
	Value	Mode		Description
	0	LEVEL0		Bluetooth SMART stack capability not available
	1	LEVEL1		Bluetooth SMART enabled
	2	LEVEL2		N/A
	3	LEVEL3		N/A
11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	RF4CE	0x0	R	**RF4CE Capability**
	RF4CE stack capability level			
	Value	Mode		Description
	0	LEVEL0		Thread stack capability not available
	1	LEVEL1		Thread stack enabled
	2	LEVEL2		N/A
	3	LEVEL3		N/A
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	THREAD	0x0	R	**Thread Capability**
	Thread stack capability level			
	Value	Mode		Description
	0	LEVEL0		RF4CE stack capability not available
	1	LEVEL1		RF4CE stack enabled
	2	LEVEL2		N/A
	3	LEVEL3		N/A
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	ZIGBEE	0x0	R	**Zigbee Capability**
	ZigBee stack capability level			
	Value	Mode		Description
	0	LEVEL0		ZigBee stack capability not available
	1	LEVEL1		GreenPower only
	2	LEVEL2		ZigBee and GreenPower
	3	LEVEL3		ZigBee Only

5.4.2.9 DEVINFO_SWCAPA1 - Software Restriction

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													R	R	R	R
Name																													XOUT	GWEN	NCPEN	RFMCUEN

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	XOUT	0x0	R	**XOUT**
	XOUT feature available			
2	GWEN	0x0	R	**Gateway**
	Gateway enabled part			
1	NCPEN	0x0	R	**NCP**
	Network co-processor enabled part. NCP only if RFMCUEN = 0			
0	RFMCUEN	0x0	R	**RF-MCU**
	RF-MCU enabled part. RF-MCU only if NCPEN = 0			

5.4.2.10 DEVINFO_EXTINFO - External Component Info

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0								0x0								0x0							
Access									R								R								R							
Name									REV								CONNECTION								TYPE							

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:16	REV	0x0	R	**Revision**
	MCM Revision			
15:8	CONNECTION	0x0	R	**Connection**
	Connection protocol to external interface			

Value	Mode	Description
0	SPI	SPI control interface
255	NONE	No interface

Bit	Name	Reset	Access	Description
7:0	TYPE	0x0	R	**Type**
	External Component			

Value	Mode	Description
255	NONE	

5.4.2.11 DEVINFO_EUI48L - EUI 48 Low

Offset	Bit Position																													
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0								0x0																							
Access	R								R																							
Name	OUI48L								UNIQUEID																							

Bit	Name	Reset	Access	Description
31:24	OUI48L	0x0	R	**OUI48L**
	Lower Octet of EUI48 Organizationally Unique Identifier			
23:0	UNIQUEID	0x0	R	**Unique ID**
	Unique identifier			

5.4.2.12 DEVINFO_EUI48H - EUI 48 High

Offset	Bit Position																													
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFF																0x0															
Access	R																R															
Name	RESERVED																OUI48H															

Bit	Name	Reset	Access	Description
31:16	RESERVED	0xFFFF	R	**RESERVED**
	Reserved			
15:0	OUI48H	0x0	R	**OUI48H**
	Upper two Octets of EUI48 OUI			

5.4.2.13 DEVINFO_EUI64L - EUI64 Low

Offset	Bit Position																													
0x048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	UNIQUEL																															

Bit	Name	Reset	Access	Description
31:0	UNIQUEL	0x0	R	**UNIQUEL**
	Lower 32 bits of EUI64 Unique Identifier			

5.4.2.14 DEVINFO_EUI64H - EUI64 High

Offset	Bit Position																													
0x04C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																								0x0							
Access	R																								R							
Name	OUI64																								UNIQUEH							

Bit	Name	Reset	Access	Description
31:8	OUI64	0x0	R	**OUI64**
	24-bit OUI identifier			
7:0	UNIQUEH	0x0	R	**UNIQUEH**
	Upper 8 bits of EUI64 unique identifier			

5.4.2.15 DEVINFO_CALTEMP - Calibration Temperature

Offset	Bit Position																													
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									TEMP							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	TEMP	0x0	R	**Cal Temp**
	Calibration temperature as an unsigned int in DegC. (0x19 = 25DegC)			

5.4.2.16 DEVINFO_EMUTEMP - EMU Temp

Offset	Bit Position																													
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																										0x0						
Access																										R						
Name																										EMUTEMPROOM						

Bit	Name	Reset	Access	Description
31:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10:2	EMUTEMPROOM	0x0	R	**Emu Room Temperature**
	EMU_TEMP temperature reading at room (calibration) temperature.			
1:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

5.4.2.17 DEVINFO_HFRCODPLLCALn - HFRCODPLL Calibration

Offset	Bit Position																													
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0				0x0		0x0		0x0			0x0					0x0		0x0							0x0						
Access	R				R		R		R			R					R		R							R						
Name	IREFTC				CMPSEL		CLKDIV		CMPBIAS			FREQRANGE					LDOHP		FINETUNING							TUNING						

Bit	Name	Reset	Access	Description
31:28	IREFTC	0x0	R	
	Tempco Trim			
27:26	CMPSEL	0x0	R	
	Comparator Load Select			
25:24	CLKDIV	0x0	R	
	Locally Divide HFRCO Clock Output			
23:21	CMPBIAS	0x0	R	
	Comparator Bias Current			
20:16	FREQRANGE	0x0	R	
	Frequency Range			
15	LDOHP	0x0	R	
	LDO High Power Mode			
14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:8	FINETUNING	0x0	R	
	Fine Tuning Value			
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:0	TUNING	0x0	R	
	Tuning Value			

5.4.2.18 DEVINFO_HFRCOEM23CALn - HFRCOEM23 Calibration

Offset	Bit Position																													
0x0A0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0				0x0		0x0		0x0			0x0					0x0		0x0							0x0						
Access	R				R		R		R			R					R		R							R						
Name	IREFTC				CMPSEL		CLKDIV		CMPBIAS			FREQRANGE					LDOHP		FINETUNING							TUNING						

Bit	Name	Reset	Access	Description
31:28	IREFTC	0x0	R	
	Tempco Trim			
27:26	CMPSEL	0x0	R	
	Comparator Load Select			
25:24	CLKDIV	0x0	R	
	Locally Divide HFRCO Clock Output			
23:21	CMPBIAS	0x0	R	
	Comparator Bias Current			
20:16	FREQRANGE	0x0	R	
	Frequency Range			
15	LDOHP	0x0	R	
	LDO High Power Mode			
14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:8	FINETUNING	0x0	R	
	Fine Tuning Value			
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:0	TUNING	0x0	R	
	Tuning Value			

5.4.2.19 DEVINFO_MODULENAME0 - Module Name Information

Offset	Bit Position																													
0x130	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFF								0xFF								0xFF								0xFF							
Access	R								R								R								R							
Name	MODCHAR4								MODCHAR3								MODCHAR2								MODCHAR1							

Bit	Name	Reset	Access	Description
31:24	MODCHAR4	0xFF	R	
	Fourth character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
23:16	MODCHAR3	0xFF	R	
	Third character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
15:8	MODCHAR2	0xFF	R	
	Second character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
7:0	MODCHAR1	0xFF	R	
	First character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			

5.4.2.20 DEVINFO_MODULENAME1 - Module Name Information

Offset	Bit Position																													
0x134	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFF								0xFF								0xFF								0xFF							
Access	R								R								R								R							
Name	MODCHAR8								MODCHAR7								MODCHAR6								MODCHAR5							

Bit	Name	Reset	Access	Description
31:24	MODCHAR8	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
23:16	MODCHAR7	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
15:8	MODCHAR6	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
7:0	MODCHAR5	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			

5.4.2.21 DEVINFO_MODULENAME2 - Module Name Information

Offset	Bit Position																													
0x138	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFF								0xFF								0xFF								0xFF							
Access	R								R								R								R							
Name	MODCHAR12								MODCHAR11								MODCHAR10								MODCHAR9							

Bit	Name	Reset	Access	Description
31:24	MODCHAR12	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
23:16	MODCHAR11	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
15:8	MODCHAR10	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
7:0	MODCHAR9	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			

5.4.2.22 DEVINFO_MODULENAME3 - Module Name Information

Offset	Bit Position																													
0x13C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFF								0xFF								0xFF								0xFF							
Access	R								R								R								R							
Name	MODCHAR16								MODCHAR15								MODCHAR14								MODCHAR13							

Bit	Name	Reset	Access	Description
31:24	MODCHAR16	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
23:16	MODCHAR15	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
15:8	MODCHAR14	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
7:0	MODCHAR13	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			

5.4.2.23 DEVINFO_MODULENAME4 - Module Name Information

Offset	Bit Position																													
0x140	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFF								0xFF								0xFF								0xFF							
Access	R								R								R								R							
Name	MODCHAR20								MODCHAR19								MODCHAR18								MODCHAR17							

Bit	Name	Reset	Access	Description
31:24	MODCHAR20	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
23:16	MODCHAR19	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
15:8	MODCHAR18	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
7:0	MODCHAR17	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			

5.4.2.24 DEVINFO_MODULENAME5 - Module Name Information

Offset	Bit Position																													
0x144	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFF								0xFF								0xFF								0xFF							
Access	R								R								R								R							
Name	MODCHAR24								MODCHAR23								MODCHAR22								MODCHAR21							

Bit	Name	Reset	Access	Description
31:24	MODCHAR24	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
23:16	MODCHAR23	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
15:8	MODCHAR22	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			
7:0	MODCHAR21	0xFF	R	
	Character of Module Name, 0xFF = unwritten, 0x00 = character not used in name			

5.4.2.25 DEVINFO_MODULENAME6 - Module Name Information

Offset	Bit Position																													
0x148	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFF																0xFF								0xFF							
Access	R																R								R							
Name	RSV																MODCHAR26								MODCHAR25							

Bit	Name	Reset	Access	Description
31:16	RSV	0xFFFF	R	
	Reserved for future use			
15:8	MODCHAR26	0xFF	R	
	Last possible character of module name, 0xFF = unwritten, 0x00 = character not used in name			
7:0	MODCHAR25	0xFF	R	
	0xFF = unwritten, 0x00 = character not used in name			

5.4.2.26 DEVINFO_MODULEINFO - Module Information

Offset	Bit Position																													
0x14C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1	0x1	0x1	0x1FF									0x1	0x1	0x1	0x1	0x1	0x7F							0x7			0x1F				
Access	R	R	R	R									R	R	R	R	R	R							R			R				
Name	EXTVALID	PHYLIMITED	PADCDC	MODNUMBERMSB									HFXOCALVAL	LFXOCALVAL	EXPRESS	LFXO	TYPE	MODNUMBER							ANTENNA			HWREV				

Bit	Name	Reset	Access	Description
31	EXTVALID	0x1	R	
	EXTINFO entry used			
	Value	Mode		Description
	0	EXTUSED		
	1	EXTUNUSED		
30	PHYLIMITED	0x1	R	
	PHY Limited			
	Value	Mode		Description
	0	LIMITED		
	1	UNLIMITED		
29	PADCDC	0x1	R	
	PAVDD Connection			
	Value	Mode		Description
	0	VDCDC		
	1	OTHER		
28:20	MODNUMBERMSB	0x1FF	R	
	Counter allowing unique identification of module per lookup when combined with MODNUMBER			
19	HFXOCALVAL	0x1	R	
	HFXO Factory Calibrated			
	Value	Mode		Description
	0	VALID		
	1	NOTVALID		
18	LFXOCALVAL	0x1	R	

Bit	Name	Reset	Access	Description
	LFXO Factory Calibrated			
	Value	Mode		Description
	0	VALID		
	1	NOTVALID		
17	EXPRESS	0x1	R	
	Blue Gecko Express			
	Value	Mode		Description
	0	SUPPORTED		
	1	NONE		
16	LFXO	0x1	R	
	Module has LFXO			
	Value	Mode		Description
	0	NONE		
	1	PRESENT		
15	TYPE	0x1	R	
	Module Type			
	Value	Mode		Description
	0	PCB		
	1	SIP		
14:8	MODNUMBER	0x7F	R	
	Counter allowing unique identification of module per lookup when combined with MODNUMBER MSB			
7:5	ANTENNA	0x7	R	
	Module Antenna Type			
	Value	Mode		Description
	0	BUILTIN		None
	1	CONNECTOR		
	2	RFPAD		
	3	INVERTEDF		
4:0	HWREV	0x1F	R	
	Module Hardware Revision. Starting from 0			

5.4.2.27 DEVINFO_MODXOCAL - Module External Oscillator Calibration Information

Offset	Bit Position																													
0x150	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset										0x7F							0xFF								0xFF							
Access										R							R								R							
Name										LFXOCAPTUNE							HFXOCTUNEXOANA								HFXOCTUNEXIANA							

Bit	Name	Reset	Access	Description
31:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:16	LFXOCAPTUNE	0x7F	R	
	LFXO Cap Tuning			
15:8	HFXOCTUNEXOANA	0xFF	R	
	Tuning capacitance on XO			
7:0	HFXOCTUNEXIANA	0xFF	R	
	Tuning capacitance on XI			

5.4.2.28 DEVINFO_HFXOCAL - High Frequency Crystal Oscillator Calibration Data

Offset	Bit Position																															
0x17C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFFFF																								0x0				0x0			
Access	R																								R				R			
Name	RESERVED																								VTRTRIMANA				SHUNTBIASANA			

Bit	Name	Reset	Access	Description
31:8	RESERVED	0xFFFFFF	R	**New BitField**
	Reserved			
7:4	VTRTRIMANA	0x0	R	
	BUFOUT Reference Trim			
3:0	SHUNTBIASANA	0x0	R	
	Shunt Regulator Bias Current			

Value	Mode	Description
0	I20UA	
1	I30UA	
2	I40UA	
3	I50UA	
4	I60UA	
5	I70UA	
6	I80UA	
7	I90UA	
8	I100UA	
9	I110UA	
10	I120UA	
11	I130UA	
12	I140UA	
13	I150UA	
14	I160UA	
15	I170UA	

5.4.2.29 DEVINFO_IADC0GAIN0 - IADC Gain Calibration

Offset	Bit Position																													
0x180	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																0x0															
Access	R																R															
Name	GAINCANA2																GAINCANA1															

Bit	Name	Reset	Access	Description
31:16	GAINCANA2	0x0	R	
	Input Gain = 2x			
15:0	GAINCANA1	0x0	R	
	Input Gain = 1x and 0.5x			

5.4.2.30 DEVINFO_IADC0GAIN1 - IADC Gain Calibration

Offset	Bit Position																													
0x184	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																0x0															
Access	R																R															
Name	GAINCANA4																GAINCANA3															

Bit	Name	Reset	Access	Description
31:16	GAINCANA4	0x0	R	
	Input Gain = 4x			
15:0	GAINCANA3	0x0	R	
	Input Gain = 3x			

5.4.2.31 DEVINFO_IADC0OFFSETCAL0 - IADC Offset Calibration

Offset	Bit Position																													
0x188	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																0x0															
Access	R																R															
Name	OFFSETANA1HIACC																OFFSETANABASE															

Bit	Name	Reset	Access	Description
31:16	OFFSETANA1HIACC	0x0	R	
	High-accuracy OSR adjustment term			
15:0	OFFSETANABASE	0x0	R	
	Base analog offset term			

5.4.2.32 DEVINFO_IADC0NORMALOFFSETCAL0 - IADC Offset Calibration

Offset	Bit Position																													
0x18C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																0x0															
Access	R																R															
Name	OFFSETANA2NORM																OFFSETANA1NORM															

Bit	Name	Reset	Access	Description
31:16	OFFSETANA2NORM	0x0	R	
	Normal mode offset gain adjustment term			
15:0	OFFSETANA1NORM	0x0	R	
	Normal mode analog offset term at OSR=2x, gain = 1x			

5.4.2.33 DEVINFO_IADC0NORMALOFFSETCAL1 - IADC Offset Calibration

Offset	Bit Position																													
0x190	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	R															
Name																	OFFSETANA3NORM															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	OFFSETANA3NORM	0x0	R	
	Normal mode offset term for OSR>=4x			

5.4.2.34 DEVINFO_IADC0HISPDOFFSETCAL0 - IADC Offset Calibration

Offset	Bit Position																													
0x194	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset								0x0									0x0															
Access								R									R															
Name								OFFSETANA2HISPD									OFFSETANA1HISPD															

Bit	Name	Reset	Access	Description
31:16	OFFSETANA2HISPD	0x0	R	
	High speed mode offset gain adjustment term			
15:0	OFFSETANA1HISPD	0x0	R	
	High speed mode analog offset term at OSR=2x, gain = 1x			

5.4.2.35 DEVINFO_IADC0HISPDOFFSETCAL1 - IADC Offset Calibration

Offset	Bit Position																													
0x198	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									OFFSETANA3HISPD							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	OFFSETANA3HISPD	0x0	R	
	High-speed mode offset term for OSR>=4x			

5.4.2.36 DEVINFO_LEGACY - Legacy Device Info

Offset	Bit Position																													
0x1FC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset												0x80																				
Access												R																				
Name												DEVICEFAMILY																				

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:16	DEVICEFAMILY Device Family	0x80	R	**Device Family**

Value	Mode	Description
16	EFR32MG1P	EFR32 Mighty Gecko Family Series 1 Device Config 1
17	EFR32MG1B	EFR32 Mighty Gecko Family Series 1 Device Config 1
18	EFR32MG1V	EFR32 Mighty Gecko Family Series 1 Device Config 1
19	EFR32BG1P	EFR32 Blue Gecko Family Series 1 Device Config 1
20	EFR32BG1B	EFR32 Blue Gecko Family Series 1 Device Config 1
21	EFR32BG1V	EFR32 Blue Gecko Family Series 1 Device Config 1
25	EFR32FG1P	EFR32 Flex Gecko Family Series 1 Device Config 1
26	EFR32FG1B	EFR32 Flex Gecko Family Series 1 Device Config 1
27	EFR32FG1V	EFR32 Flex Gecko Family Series 1 Device Config 1
28	EFR32MG12P	EFR32 Mighty Gecko Family Series 1 Device Config 2
29	EFR32MG12B	EFR32 Mighty Gecko Family Series 1 Device Config 2
30	EFR32MG12V	EFR32 Mighty Gecko Family Series 1 Device Config 2
31	EFR32BG12P	EFR32 Blue Gecko Family Series 1 Device Config 2
32	EFR32BG12B	EFR32 Blue Gecko Family Series 1 Device Config 2
33	EFR32BG12V	EFR32 Blue Gecko Family Series 1 Device Config 2
37	EFR32FG12P	EFR32 Flex Gecko Family Series 1 Device Config 2
38	EFR32FG12B	EFR32 Flex Gecko Family Series 1 Device Config 2
39	EFR32FG12V	EFR32 Flex Gecko Family Series 1 Device Config 2
40	EFR32MG13P	EFR32 Mighty Gecko Family Series 13 Device Config 3
41	EFR32MG13B	EFR32 Mighty Gecko Family Series 13 Device Config 3
42	EFR32MG13V	EFR32 Mighty Gecko Family Series 1 Device Config 3
43	EFR32BG13P	EFR32 Blue Gecko Family Series 1 Device Config 3

Bit	Name	Reset	Access	Description
	44	EFR32BG13B		EFR32 Blue Gecko Family Series 1 Device Config 3
	45	EFR32BG13V		EFR32 Blue Gecko Family Series 1 Device Config 3
	49	EFR32FG13P		EFR32 Flex Gecko Family Series 1 Device Config 3
	50	EFR32FG13B		EFR32 Flex Gecko Family Series 1 Device Config 3
	51	EFR32FG13V		EFR32 Flex Gecko Family Series 1 Device Config 3
	52	EFR32MG14P		EFR32 Mighty Gecko Family Series 1 Device Config 4
	53	EFR32MG14B		EFR32 Mighty Gecko Family Series 1 Device Config 4
	54	EFR32MG14V		EFR32 Mighty Gecko Family Series 1 Device Config 4
	55	EFR32BG14P		EFR32 Blue Gecko Family Series 1 Device Config 4
	56	EFR32BG14B		EFR32 Blue Gecko Family Series 1 Device Config 4
	57	EFR32BG14V		EFR32 Blue Gecko Family Series 1 Device Config 4
	61	EFR32FG14P		EFR32 Flex Gecko Family Series 1 Device Config 4
	62	EFR32FG14B		EFR32 Flex Gecko Family Series 1 Device Config 4
	63	EFR32FG14V		EFR32 Flex Gecko Family Series 1 Device Config 4
	71	EFM32G		EFM32 Gecko Device Family
	72	EFM32GG		EFM32 Giant Gecko Device Family
	73	EFM32TG		EFM32 Tiny Gecko Device Family
	74	EFM32LG		EFM32 Leopard Gecko Device Family
	75	EFM32WG		EFM32 Wonder Gecko Device Family
	76	EFM32ZG		EFM32 Zero Gecko Device Family
	77	EFM32HG		EFM32 Happy Gecko Device Family
	81	EFM32PG1B		EFM32 Pearl Gecko Device Family Series 1 Device Config 1
	83	EFM32JG1B		EFM32 Jade Gecko Device Family Series 1 Device Config 1
	85	EFM32PG12B		EFM32 Pearl Gecko Device Family Series 1 Device Config 2
	87	EFM32JG12B		EFM32 Jade Gecko Device Family Series 1 Device Config 2
	89	EFM32PG13B		EFM32 Pearl Gecko Device Family Series 1 Device Config 3
	91	EFM32JG13B		EFM32 Jade Gecko Device Family Series 1 Device Config 3
	100	EFM32GG11B		EFM32 Giant Gecko Device Family Series 1 Device Config 1
	103	EFM32TG11B		EFM32 Giant Gecko Device Family Series 1 Device Config 1
	120	EZR32LG		EZR32 Leopard Gecko Device Family
	121	EZR32WG		EZR32 Wonder Gecko Device Family
	122	EZR32HG		EZR32 Happy Gecko Device Family
	128	SERIES2V0		DI page is encoded with the series 2 layout. Check alternate location.
15:0	*Reserved*			*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*

5.4.2.37 DEVINFO_RTHERM -

Offset	Bit Position																													
0x25C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									RTHERM							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	RTHERM	0x0	R	
	Calibrated Thermistor Resistor			

5.5 ICACHE - Instruction Cache

The ICACHE provides fast access to recently executed instructions, improving both speed and power consumption of code execution. The instruction cache is enabled by default, but can be disabled by setting CACHEDIS in ICACHE_CTRL. When enabled, the instruction cache typically reduces the number of flash reads significantly, thus saving energy. In most cases, a cache hit-rate of more than 70 % is achievable. When a 32-bit instruction fetch hits in the cache, the data is returned to the processor in one clock cycle, bypassing the flash access wait-states. The cache content is retained in EM2 and EM3.

The instruction cache is connected directly to the CODE bus on the ARM core and functions as a memory access filter between the processor and the memory system, as illustrated in Figure 5.1 Instruction Cache Block Diagram on page 85. The cache consists of an access filter, lookup logic, SRAM, and three performance counters. The access filter checks if a transfer is an instruction fetch located in a cacheable region. If it is the cache lookup logic and SRAM is enabled. Otherwise, the cache is bypassed and the access is forwarded to the memory system. If lookup is enabled data is either returned from the cache (hit) or fetch from the memory system and cached (miss).

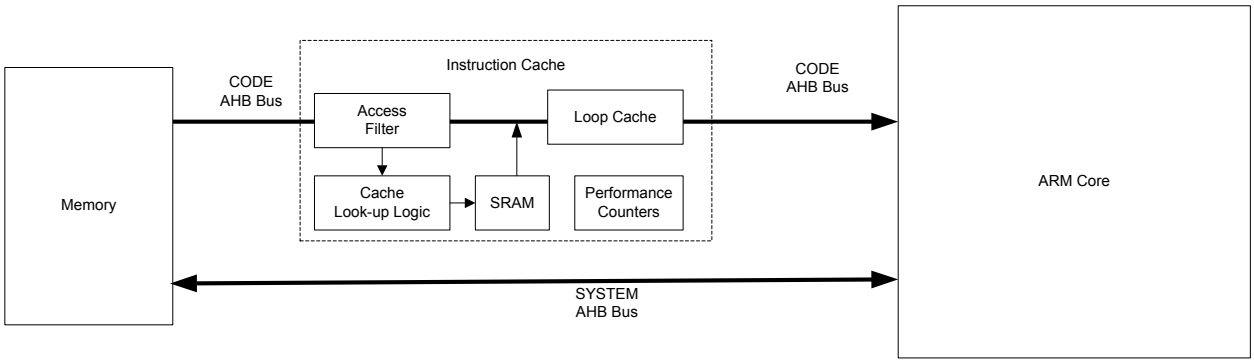


Figure 5.1. Instruction Cache Block Diagram

Note that while all access to code spaces use the CODE bus only instruction fetches are cached. Data accesses to the CODE region are passed through the ICACHE.

5.5.1 Cache Operation

It is highly recommended to keep the cache enabled. To improve cache-efficiency, sections of code with very low cache hit rate should not be cached. This is achieved by placing these code sections in non-cacheable MPU regions and setting USEMPU in ICACHE_CTRL. When USEMPU is set, instruction fetches to non-cacheable MPU regions will not be looked up or saved in cache. This feature may also be used to avoid instructions from low-power memory taking up space from more power-hungry memory. For more information on the MPU see the ARM Cortex®-M33 MPU documentation.

The optional loop-cache is optimized to store smaller code-loops efficiently. The loop-cache is enabled when LPLEVEL in ICACHE_LPMODE is set to ADVANCED or MINACTIVITY. The difference between the two settings is that when MINACTIVITY is selected loop-cache outputs may be gated off to reduce power at the cost of more wait-states due to loop-cache misses. Having LPLEVEL set to BASIC disables the loop-cache functionality completely. NESTFACTOR in ICACHE _LPMODE is used to decide when to stick with the currently detected loop rather than start tracking a new loop. Optimal value will depend on the actual code running, meaning that this setting may be tuned for optimal performance.

By default, the instruction cache is automatically invalidated when the contents of the flash are changed (i.e. written or erased). In many cases, however, the application only makes changes to data in the flash, not code. In this case, the automatic invalidate feature can be disabled by setting AUTOFLUSHDIS in ICACHE_CTRL. The cache can also be manually invalidated by writing 1 to FLUSH in ICACHE_CMD.

In the event that a parity error in the cache is detected, the RAMERROR flag will be set in ICACHE_IF. The data is automatically reloaded when this occurs so no action is required by software. This flag is informational only and can be used to detect the rate of corruption events. If RAMERROR in ICACHE_IEN is set, an interrupt will be triggered.

The cache is automatically flushed whenever a bus fault occurs. If this occurs during performance counting the counts will be effected.

5.5.2 Performance Measurement

To measure the hit-rate of a code-section, the built-in performance counters can be used. Before the section, start the performance counters by setting STARTPC in ICACHE_CMD register. This starts the performance counters, counting from 0. At the end of the section, stop the performance counters by setting STOPPC in ICACHE_CMD. The number of cache hits and cache misses for that section can then be read from PCHITS and PCMISSES. The cache hit-ratio can be calculated as PCHITS / (PCHITS + PCMISSES). PCAHITS contains the loopcache hits only. Any hits in PCAHITS are also counted in PCHITS. The loopcache hit-ratio can be calculated as PCAHITS / (PCHITS + PCMISSES). When PCHITS/PCAHITS/PCMISSES overflow, the HITOF/AHITOF/MISSOF interrupt flags are set respectively. These flags must be cleared by software. The range of the performance counters can be extended by increasing a counter in the interrupt routine. The performance counters only count when a cache lookup is performed. Access to non-cacheable regions, data fetches, and access made while the ICACHE is disabled do not increment PCMISSES.

Software may check the if the performance counters are running using PCRUNNING in ICACHE_STATUS.

5.5.3 ICACHE Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	ICACHE_IPVERSION	R	IP Version
0x004	ICACHE_CTRL	RW	Control Register
0x008	ICACHE_PCHITS	RH	Performance Counter Hits
0x00C	ICACHE_PCMISSES	RH	Performance Counter Misses
0x010	ICACHE_PCAHITS	RH	Performance Counter Advanced Hits
0x014	ICACHE_STATUS	RH	Status Register
0x018	ICACHE_CMD	W	Command Register
0x01C	ICACHE_LPMODE	RW	Low Power Mode
0x020	ICACHE_IF	RWH INTFLAG	Interrupt Flag
0x024	ICACHE_IEN	RW	Interrupt Enable
0x1000	ICACHE_IPVERSION_SET	R	IP Version
0x1004	ICACHE_CTRL_SET	RW	Control Register
0x1008	ICACHE_PCHITS_SET	RH	Performance Counter Hits
0x100C	ICACHE_PCMISSES_SET	RH	Performance Counter Misses
0x1010	ICACHE_PCAHITS_SET	RH	Performance Counter Advanced Hits
0x1014	ICACHE_STATUS_SET	RH	Status Register
0x1018	ICACHE_CMD_SET	W	Command Register
0x101C	ICACHE_LPMODE_SET	RW	Low Power Mode
0x1020	ICACHE_IF_SET	RWH INTFLAG	Interrupt Flag
0x1024	ICACHE_IEN_SET	RW	Interrupt Enable
0x2000	ICACHE_IPVERSION_CLR	R	IP Version
0x2004	ICACHE_CTRL_CLR	RW	Control Register
0x2008	ICACHE_PCHITS_CLR	RH	Performance Counter Hits
0x200C	ICACHE_PCMISSES_CLR	RH	Performance Counter Misses
0x2010	ICACHE_PCAHITS_CLR	RH	Performance Counter Advanced Hits
0x2014	ICACHE_STATUS_CLR	RH	Status Register
0x2018	ICACHE_CMD_CLR	W	Command Register
0x201C	ICACHE_LPMODE_CLR	RW	Low Power Mode
0x2020	ICACHE_IF_CLR	RWH INTFLAG	Interrupt Flag
0x2024	ICACHE_IEN_CLR	RW	Interrupt Enable
0x3000	ICACHE_IPVERSION_TGL	R	IP Version
0x3004	ICACHE_CTRL_TGL	RW	Control Register
0x3008	ICACHE_PCHITS_TGL	RH	Performance Counter Hits
0x300C	ICACHE_PCMISSES_TGL	RH	Performance Counter Misses
0x3010	ICACHE_PCAHITS_TGL	RH	Performance Counter Advanced Hits

Offset	Name	Type	Description
0x3014	ICACHE_STATUS_TGL	RH	Status Register
0x3018	ICACHE_CMD_TGL	W	Command Register
0x301C	ICACHE_LPMODE_TGL	RW	Low Power Mode
0x3020	ICACHE_IF_TGL	RWH INTFLAG	Interrupt Flag
0x3024	ICACHE_IEN_TGL	RW	Interrupt Enable

5.5.4 ICACHE Register Description

5.5.4.1 ICACHE_IPVERSION - IP Version

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x0	R	**IP version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

5.5.4.2 ICACHE_CTRL - Control Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														AUTOFLUSHDIS	USEMPU	CACHEDIS

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	AUTOFLUSHDIS	0x0	RW	**Automatic Flushing Disable**
	Disables automatic flushing based on Internal Flash write/erase			
1	USEMPU	0x0	RW	**Use MPU**
	Use MPU to select non/cacheable regions			
0	CACHEDIS	0x0	RW	**Cache Disable**
	Disables caching for all regions			

5.5.4.3 ICACHE_PCHITS - Performance Counter Hits

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	PCHITS																															

Bit	Name	Reset	Access	Description
31:0	PCHITS	0x0	R	**Performance Counter Hits**
	Hit counter value			

5.5.4.4 ICACHE_PCMISSES - Performance Counter Misses

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	PCMISSES																															

Bit	Name	Reset	Access	Description
31:0	PCMISSES	0x0	R	**Performance Counter Misses**
	Miss counter value			

5.5.4.5 ICACHE_PCAHITS - Performance Counter Advanced Hits

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	PCAHITS																															

Bit	Name	Reset	Access	Description
31:0	PCAHITS	0x0	R	**Performance Counter Advanced Hits**
	Hit counter value for hits due to Advanced Buffering mode. These hits are also represented in PCHITS.			

5.5.4.6 ICACHE_STATUS - Status Register

Offset	Bit Position																														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																R
Name																																PCRUNNING

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	PCRUNNING	0x0	R	**PC Running**
	Performance Counters are running			

5.5.4.7 ICACHE_CMD - Command Register

Offset	Bit Position																														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														W(nB)	W(nB)	W(nB)
Name																														STOPPC	STARTPC	FLUSH

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	STOPPC	0x0	W(nB)	**Stop Performance Counters**
	Stops the Performance Counters			
1	STARTPC	0x0	W(nB)	**Start Performance Counters**
	Starts the Performance Counters			
0	FLUSH	0x0	W(nB)	**Flush**
	Clears Cached Data			

5.5.4.8 ICACHE_LPMODE - Low Power Mode

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x2						0x3	
Access																									RW						RW	
Name																									NESTFACTOR						LPLEVEL	

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:4	NESTFACTOR	0x2	RW	**Low Power Nest Factor** Parameter used in the advanced buffering mode to control its estimation when a branch access is likely to be accssed in the near future. In general, a higher number will improve performance in code with deeply nested loops.
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	LPLEVEL	0x3	RW	**Low Power Level** Controls the low-power level of the cache. In general, the default setting is best for most applications.

Value	Mode	Description
0	BASIC	Base instruction cache functionality
1	ADVANCED	Advanced buffering mode, where the cache uses the fetch pattern to predict highly accessed data and store it in low-energy memory
3	MINACTIVITY	Minimum activity mode, which allows the cache to minimize activity in logic that it predicts has a low probability being used. This mode can introduce wait-states into the instruction fetch stream when the cache exits one of its low-activity states. The number of wait-states introduced is small, but users running with 0-wait-state memory and wishing to reduce the variability that the cache might introduce with additional wait-states may wish to lower the cache low-power level. Note, this mode includes the advanced buffering mode functionality.

5.5.4.9 ICACHE_IF - Interrupt Flag

Offset	Bit Position																														
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0						0x0	0x0	0x0
Access																								RW						RW	RW	RW
Name																								RAMERROR						AHITOF	MISSOF	HITOF

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	RAMERROR	0x0	RW	**RAM error Interrupt Flag**
	RAM parity error detected			
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	AHITOF	0x0	RW	**Advanced Hit Overflow Interrupt Flag**
	Advanced hit performance counter has overflowed			
1	MISSOF	0x0	RW	**Miss Overflow Interrupt Flag**
	Miss performance counter has overflowed			
0	HITOF	0x0	RW	**Hit Overflow Interrupt Flag**
	Hit performance counter has overflowed			

5.5.4.10 ICACHE_IEN - Interrupt Enable

Offset	Bit Position																														
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0						0x0	0x0	0x0
Access																								RW						RW	RW	RW
Name																								RAMERROR						AHITOF	MISSOF	HITOF

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	RAMERROR	0x0	RW	**RAM error Interrupt Enable**
	Enable RAMERROR interrupt			
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	AHITOF	0x0	RW	**Advanced Hit Overflow Interrupt Enable**
	Enable AHITOF interrupt			
1	MISSOF	0x0	RW	**Miss Overflow Interrupt Enable**
	Enable MISSOF interrupt			
0	HITOF	0x0	RW	**Hit Overflow Interrupt Enable**
	Enable HITOF interrupt			

5.6 SYSCFG - System Configuration

The system has one main SRAM block, DMEM0.

5.6.1 RAM Retention

DMEM0 is broken into 16 KB banks. By default all banks are retained in EM2/EM3. Sleep mode current can be significantly reduced by fully powering down banks that do not need to be retained. To select the amount of RAM to be powered down in EM2/EM3, set RAMRETNCTRL in SYSCFG_DMEM0RETNCTRL to the desired value.

5.6.2 ECC

DMEM0 supports one bit correction and two bit detection ECC.

To enable error detection for DMEM0, set ECCEN in MPAHBRAM_CTRL. To enable auto-correction of one bit errors in DMEM0, set ECCWEN in MPAHBRAM_CTRL.

When ECC error events in DMEM0 are detected, the corresponding bits in MPAHBRAM_IF are set. Errors arising from a specific port 'x' will be indicated by the AHBxERR1B or AHBxERR2B flags. When a flag is set, an interrupt will be triggered if the corresponding interrupt enable bit is set in MPAHBRAM_IEN. When an error occurs, the address of the detected error is written to MPAHBRAM_ECCERRADDRx for the respective port 'x'. The address registers are sticky and will not be loaded with a new address until they are cleared through MPAHBRAM_CMD.CLEARECCADDRx. If multiple ECC errors occur without ECCERRADDRn being cleared, the Px bit in MPAHBRAM_ECCMERRIND will be set. These status bits are also sticky, and are cleared with MPAHBRAM_CMD.CLEARECCADDRx.

Upon a two bit ECC error in DMEM, MPAHBRAM can also issue a bus fault. To enable this, set the ECCERRFAULTEN bit in MPAHBRAM_CTRL.

The recommend procedure for initializing ECC RAM is to first enable ECC, then write zeros to all locations. This will clear the RAM and initialize the syndrome. If the ECC RAM is not written as described, then any reads to uninitialized RAM locations will result in an ECC error.

Note: The RAM ECC feature must be enabled to achieve good long term reliability. The long term reliability of the RAM is only specified with ECC enabled.

5.6.3 Software Interrupts

The SYSCFG block also provides some software interrupts that can be used to communicate between software tasks. To trigger a software interrupt set the corresponding bit in SYSCFG_IF.

5.6.4 Bus faults

By default, two bit ECC errors and reads to unmapped addresses trigger a BusFault. These bus fault sources can be disabled by clearing RAMECCERRFAULTEN and ADDRFAULTEN in SYSCFG_CTRL.

5.6.5 SYSCFG Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x004	SYSCFG_IPVERSION	R	IP Version ID
0x008	SYSCFG_IF	RWH INTFLAG	Interrupt Flag
0x00C	SYSCFG_IEN	RW	Interrupt Enable
0x014	SYSCFG_CHIPREVHW	RWH	Chip Revision, Hard-Wired
0x018	SYSCFG_CHIPREV	RW	Chip Revision
0x024	SYSCFG_CFGSYSTIC	RW	SysTick Clock Source
0x200	SYSCFG_CTRL	RW	Control
0x208	SYSCFG_DMEM0RETNCTRL	RW	DMEM0 Retention Control
0x30C	SYSCFG_RAMBIASCONF	RW	RAM Bias Configuration
0x418	SYSCFG_ICACHERAM-RETNCTRL	RW	HOST ICACHERAM Retention Control
0x41C	SYSCFG_DMEM0PORTMAP-SEL	RW	DMEM0 Port Remap Selection
0x600	SYSCFG_ROOTDATA0	RW	Data Register 0
0x604	SYSCFG_ROOTDATA1	RW	Data Register 1
0x608	SYSCFG_ROOTLOCKSTATUS	RH	Lock Status
0x60C	SYSCFG_ROOTSESWVER-SION	RW	SE SW Version
0x1004	SYSCFG_IPVERSION_SET	R	IP Version ID
0x1008	SYSCFG_IF_SET	RWH INTFLAG	Interrupt Flag
0x100C	SYSCFG_IEN_SET	RW	Interrupt Enable
0x1014	SYSCFG_CHIPREVHW_SET	RWH	Chip Revision, Hard-Wired
0x1018	SYSCFG_CHIPREV_SET	RW	Chip Revision
0x1024	SYSCFG_CFGSYSTIC_SET	RW	SysTick Clock Source
0x1200	SYSCFG_CTRL_SET	RW	Control
0x1208	SYSCFG_DMEM0RETNCTRL_SET	RW	DMEM0 Retention Control
0x130C	SYSCFG_RAMBIASCONF_SET	RW	RAM Bias Configuration
0x1418	SYSCFG_ICACHERAM-RETNCTRL_SET	RW	HOST ICACHERAM Retention Control
0x141C	SYSCFG_DMEM0PORTMAP-SEL_SET	RW	DMEM0 Port Remap Selection
0x1600	SYSCFG_ROOTDATA0_SET	RW	Data Register 0
0x1604	SYSCFG_ROOTDATA1_SET	RW	Data Register 1
0x1608	SYSCFG_ROOTLOCKSTA-TUS_SET	RH	Lock Status
0x160C	SYSCFG_ROOTSESWVER-SION_SET	RW	SE SW Version

Offset	Name	Type	Description
0x2004	SYSCFG_IPVERSION_CLR	R	IP Version ID
0x2008	SYSCFG_IF_CLR	RWH INTFLAG	Interrupt Flag
0x200C	SYSCFG_IEN_CLR	RW	Interrupt Enable
0x2014	SYSCFG_CHIPREVHW_CLR	RWH	Chip Revision, Hard-Wired
0x2018	SYSCFG_CHIPREV_CLR	RW	Chip Revision
0x2024	SYSCFG_CFGSYSTIC_CLR	RW	SysTick Clock Source
0x2200	SYSCFG_CTRL_CLR	RW	Control
0x2208	SYSCFG_DMEM0RETNCTRL_CLR	RW	DMEM0 Retention Control
0x230C	SYSCFG_RAMBIASCONF_CLR	RW	RAM Bias Configuration
0x2418	SYSCFG_ICACHERAM-RETNCTRL_CLR	RW	HOST ICACHERAM Retention Control
0x241C	SYSCFG_DMEM0PORTMAP-SEL_CLR	RW	DMEM0 Port Remap Selection
0x2600	SYSCFG_ROOTDATA0_CLR	RW	Data Register 0
0x2604	SYSCFG_ROOTDATA1_CLR	RW	Data Register 1
0x2608	SYSCFG_ROOTLOCKSTA-TUS_CLR	RH	Lock Status
0x260C	SYSCFG_ROOTSESWVER-SION_CLR	RW	SE SW Version
0x3004	SYSCFG_IPVERSION_TGL	R	IP Version ID
0x3008	SYSCFG_IF_TGL	RWH INTFLAG	Interrupt Flag
0x300C	SYSCFG_IEN_TGL	RW	Interrupt Enable
0x3014	SYSCFG_CHIPREVHW_TGL	RWH	Chip Revision, Hard-Wired
0x3018	SYSCFG_CHIPREV_TGL	RW	Chip Revision
0x3024	SYSCFG_CFGSYSTIC_TGL	RW	SysTick Clock Source
0x3200	SYSCFG_CTRL_TGL	RW	Control
0x3208	SYSCFG_DMEM0RETNCTRL_TGL	RW	DMEM0 Retention Control
0x330C	SYSCFG_RAMBIASCONF_TGL	RW	RAM Bias Configuration
0x3418	SYSCFG_ICACHERAM-RETNCTRL_TGL	RW	HOST ICACHERAM Retention Control
0x341C	SYSCFG_DMEM0PORTMAP-SEL_TGL	RW	DMEM0 Port Remap Selection
0x3600	SYSCFG_ROOTDATA0_TGL	RW	Data Register 0
0x3604	SYSCFG_ROOTDATA1_TGL	RW	Data Register 1
0x3608	SYSCFG_ROOTLOCKSTA-TUS_TGL	RH	Lock Status
0x360C	SYSCFG_ROOTSESWVER-SION_TGL	RW	SE SW Version

5.6.6 SYSCFG Register Description

5.6.6.1 SYSCFG_IPVERSION - IP Version ID

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x7																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x7	R	**New BitField**
	ID indicating version of IP			

5.6.6.2 SYSCFG_IF - Interrupt Flag

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset			0x0	0x0			0x0	0x0							0x0	0x0			0x0	0x0	0x0	0x0	0x0	0x0					0x0	0x0	0x0	0x0
Access			RW	RW			RW	RW							RW	RW			RW	RW	RW	RW	RW	RW					RW	RW	RW	RW
Name			FRCRAMERR2B	FRCRAMERR1B			SEQRAMERR2B	SEQRAMERR1B							SRW2HOSTBUSERRIF	HOST2SRWBUSERRIF			FPIXC	FPIDC	FPOFC	FPUFC	FPDZC	FPIOC					SW3	SW2	SW1	SW0

Bit	Name	Reset	Access	Description
31:30	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
29	FRCRAMERR2B	0x0	RW	**FRCRAM Error 2-bit Interrupt Flag**
	FRCRAM 2-bit ECC Error Interrupt flag.			
28	FRCRAMERR1B	0x0	RW	**FRCRAM Error 1-bit Interrupt Flag**
	FRCRAM 1-bit ECC Error Interrupt flag.			
27:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25	SEQRAMERR2B	0x0	RW	**SEQRAM Error 2-bit Interrupt Flag**
	SEQRAM 2-bit ECC Error Interrupt flag.			
24	SEQRAMERR1B	0x0	RW	**SEQRAM Error 1-bit Interrupt Flag**
	SEQRAM 1-bit ECC Error Interrupt flag.			
23:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	SRW2HOSTBUSERRIF	0x0	RW	**SRW2HOSTBUSERRIF Interrupt Flag**
	AHB SRW to AHB HOST Bus Error Interrupt Flag			
16	HOST2SRWBUSERRIF	0x0	RW	**HOST2SRWBUSERRIF Interrupt Flag**
	AHB Host to AHB SRW Bus Error Interrupt Flag			
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	FPIXC	0x0	RW	**FPU Inexact interrupt flag**
	Set upon FPU inexact exception			
12	FPIDC	0x0	RW	**FPU Input denormal interrupt flag**
	Set upon FPU input denormal exception			
11	FPOFC	0x0	RW	**FPU Overflow interrupt flag**
	Set upon FPU overflow exception			
10	FPUFC	0x0	RW	**FPU Underflow interrupt flag**

Bit	Name	Reset	Access	Description
	Set upon FPU underflow exception			
9	FPDZC	0x0	RW	**FPU Divide by zero interrupt flag**
	Set upon FPU divide by zero operation			
8	FPIOC	0x0	RW	**FPU Invalid Operation interrupt flag**
	Set upon FPU invalid operation			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	SW3	0x0	RW	**Software Interrupt Flag**
	Software interrupts			
2	SW2	0x0	RW	**Software Interrupt Flag**
	Software interrupts			
1	SW1	0x0	RW	**Software Interrupt Flag**
	Software interrupts			
0	SW0	0x0	RW	**Software Interrupt Flag**
	Software interrupts			

5.6.6.3 SYSCFG_IEN - Interrupt Enable

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset			0x0	0x0			0x0	0x0							0x0	0x0			0x0	0x0	0x0	0x0	0x0	0x0					0x0	0x0	0x0	0x0
Access			RW	RW			RW	RW							RW	RW			RW	RW	RW	RW	RW	RW					RW	RW	RW	RW
Name			FRCRAMERR2B	FRCRAMERR1B			SEQRAMERR2B	SEQRAMERR1B							SRW2HOSTBUSERRIEN	HOST2SRWBUSERRIEN			FPIXC	FPIDC	FPOFC	FPUFC	FPDZC	FPIOC					SW3	SW2	SW1	SW0

Bit	Name	Reset	Access	Description
31:30	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
29	FRCRAMERR2B	0x0	RW	**FRCRAM Error 2-bit Interrupt Enable**
	Set to enable the FRCRAM2ERR2BIF Interrupt			
28	FRCRAMERR1B	0x0	RW	**FRCRAM Error 1-bit Interrupt Enable**
	Set to enable the FRCRAM2ERR1BIF Interrupt			
27:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25	SEQRAMERR2B	0x0	RW	**SEQRAM Error 2-bit Interrupt Enable**
	Set to enable the SEQRAM2ERR2BIF Interrupt			
24	SEQRAMERR1B	0x0	RW	**SEQRAM Error 1-bit Interrupt Enable**
	Set to enable the SEQRAM2ERR1BIF Interrupt			
23:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	SRW2HOSTBUSER-RIEN	0x0	RW	**SRW2HOSTBUSERRIEN Interrupt Enable**
	Set to enable the SRW2HOSTBUSERR Interrupt			
16	HOST2SRWBUSER-RIEN	0x0	RW	**HOST2SRWBUSERRIEN Interrupt Enable**
	Set to enable the HOST2SRWBUSERR Interrupt			
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	FPIXC	0x0	RW	**FPU Inexact Interrupt Enable**
	Set to enable the FPIXCIF Interrupt			
12	FPIDC	0x0	RW	**FPU Input denormal Interrupt Enable**
	Set to enable the FPIDCIF Interrupt			
11	FPOFC	0x0	RW	**FPU Overflow Interrupt Enable**

Bit	Name	Reset	Access	Description
	Set to enable the FPOFCIF Interrupt			
10	FPUFC	0x0	RW	**FPU Underflow Interrupt Enable**
	Set to enable the FPUFCIF Interrupt			
9	FPDZC	0x0	RW	**FPU Divide by zero Interrupt Enable**
	Set to enable the FPDZCIF Interrupt			
8	FPIOC	0x0	RW	**FPU Invalid Operation Interrupt Enable**
	Set to enable the FPIOCIF Interrupt			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	SW3	0x0	RW	**Software Interrupt Enable**
	Set to enable the Software Interrupts			
2	SW2	0x0	RW	**Software Interrupt Enable**
	Set to enable the Software Interrupts			
1	SW1	0x0	RW	**Software Interrupt Enable**
	Set to enable the Software Interrupts			
0	SW0	0x0	RW	**Software Interrupt Enable**
	Set to enable the Software Interrupts			

5.6.6.4 SYSCFG_CHIPREVHW - Chip Revision, Hard-Wired

Offset	Bit Position																															
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x1				0x0				0x12											
Access													R				R				RW											
Name													MAJOR				MINOR				PARTNUMBER											

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	MAJOR	0x1	R	**Hardwired Chip Major value**
	Stores the Hardwired Chip Revision Minor signal value upper 4bits, and reflects tie-block value that can change from revision to revision.			
15:12	MINOR	0x0	R	**Hardwired Chip Minor value**
	Stores the Hardwired Chip Revision Minor signal value lower 4 bits.			
11:0	PARTNUMBER	0x12	RW	**Hardwired Chip Part Number value**
	Hardwired Chip Part Number signal value			

5.6.6.5 SYSCFG_CHIPREV - Chip Revision

Offset	Bit Position																															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0				0x0				0x0											
Access													RW				RW				RW											
Name													MAJOR				MINOR				PARTNUMBER											

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	MAJOR	0x0	RW	**Chip Revision Major value**
	Upper 4bits of Chip Revision Minor			
15:12	MINOR	0x0	RW	**Chip Revision Minor value**
	Lower 4bits of Chip Revision Minor			
11:0	PARTNUMBER	0x0	RW	**Chip Part Number value**

5.6.6.6 SYSCFG_CFGSYSTIC - SysTick Clock Source

Offset	Bit Position																															
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																SYSTICEXTCLKEN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	SYSTICEXTCLKEN	0x0	RW	**SysTick External Clock Enable**
	Set 1 to use an external clock as the M33 System Tick.			

5.6.6.7 SYSCFG_CTRL - Control

Offset	Bit Position																													
0x200	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x1				0x1	0x1
Access																											RW				RW	RW
Name																											RAMECCERRFAULTEN				CLKDISFAULTEN	ADDRFAULTEN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	RAMECCERRFAULTEN	0x1	RW	**Two bit ECC error bus fault response ena**
	When this bit is set, busfaults are generated if 2-bit ECC error occurs.			
4:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	CLKDISFAULTEN	0x1	RW	**Disabled Clkbus Bus Fault Enable**
	When this bit is set, busfaults are generated on accesses to peripherals with disabled bus clock			
0	ADDRFAULTEN	0x1	RW	**Invalid Address Bus Fault Response Enabl**
	When this bit is set, busfaults are generated on accesses to unmapped parts of system and code address space			

5.6.6.8 SYSCFG_DMEM0RETNCTRL - DMEM0 Retention Control

Offset	Bit Position																															
0x208	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									RAMRETNCTRL							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	RAMRETNCTRL	0x0	RW	**DMEM0 blockset retention control**

DMEM0 RAM blockset retention controls in EM23 with full access in EM01.Each bank has a bit to select to Retain or Powerdown

Value	Mode	Description
0	ALLON	None of the RAM blocks powered down
32768	BLK15	Power down RAM block 15 (address range 0x2003C000-0x20040000)
49152	BLK14TO15	Power down RAM blocks 14 and above (address range 0x20038000-0x20040000)
57344	BLK13TO15	Power down RAM blocks 13 and above (address range 0x20034000-0x20040000)
61440	BLK12TO15	Power down RAM blocks 12 and above (address range 0x20030000-0x20040000)
63488	BLK11TO15	Power down RAM blocks 11 and above (address range 0x2002C000-0x20040000)
64512	BLK10TO15	Power down RAM blocks 10 and above (address range 0x20028000-0x20040000)
65024	BLK9TO15	Power down RAM blocks 9 and above (address range 0x20024000-0x20040000)
65280	BLK8TO15	Power down RAM blocks 8 and above (address range 0x20020000-0x20040000)
65408	BLK7TO15	Power down RAM blocks 7 and above (address range 0x2001C000-0x20040000)
65472	BLK6TO15	Power down RAM blocks 6 and above (address range 0x20018000-0x20040000)
65504	BLK5TO15	Power down RAM blocks 5 and above (address range 0x20014000-0x20040000)
65520	BLK4TO15	Power down RAM blocks 4 and above (address range 0x20010000-0x20040000)

Bit	Name	Reset	Access	Description
	65528	BLK3TO15		Power down RAM blocks 3 and above (address range 0x2000C000-0x20040000)
	65532	BLK2TO15		Power down RAM blocks 2 and above (address range 0x20008000-0x20040000)
	65534	BLK1TO15		Power down RAM blocks 1 and above (address range 0x20004000-0x20040000)
	65535	ALLOFF		Power down All RAM blocks

5.6.6.9 SYSCFG_RAMBIASCONF - RAM Bias Configuration

Offset	Bit Position																															
0x30C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x2		
Access																														RW		
Name																														RAMBIASCTRL		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	RAMBIASCTRL	0x2	RW	**RAM Bias Control**

RAM blockset Bias control. Voltage Source Bias control setting

Value	Mode	Description
0	NO	None
1	VSB100	Voltage Source Bias 100mV
2	VSB200	Voltage Source Bias 200mV
4	VSB300	Voltage Source Bias 300mV
8	VSB400	Voltage Source Bias 400mV

5.6.6.10 SYSCFG_ICACHERAMRETNCTRL - HOST ICACHERAM Retention Control

Offset	Bit Position																													
0x418	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																RAMRETNCTRL

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	RAMRETNCTRL	0x0	RW	**ICACHERAM Retention control**

Host ICACHE RAM power-down in EM23 with full access in EM01.

Value	Mode	Description
0	ALLON	None of the Host ICACHE RAM blocks powered down
1	ALLOFF	Power down all Host ICACHE RAM blocks

5.6.6.11 SYSCFG_DMEM0PORTMAPSEL - DMEM0 Port Remap Selection

Offset	Bit Position																											
0x41C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x1		0x3		0x2		0x1		0x0		0x0		0x1		0x1	
Access																	RW		RW		RW		RW		RW		RW		RW		RW	
Name																	MVPAHBDATA2PORTSEL		MVPAHBDATA1PORTSEL		MVPAHBDATA0PORTSEL		SRWECA1PORTSEL		SRWECA0PORTSEL		AHBSRWPORTSEL		SRWAESPORTSEL		LDMAPORTSEL	

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:14	MVPAHBDATA2PORTSEL	0x1	RW	**MVPAHBDATA2 portmap selection**
	MVPWDMA address remap selection, default configured to MPAHBRAM1			
13:12	MVPAHBDATA1PORTSEL	0x3	RW	**MVPAHBDATA1 portmap selection**
	MVPRDMA1 address remap selection, default configured to MPAHBRAM3			
11:10	MVPAHBDATA0PORTSEL	0x2	RW	**MVPAHBDATA0 portmap selection**
	MVPRDMA0 address remap selection, default configured to MPAHBRAM2			
9:8	SRWECA1PORTSEL	0x1	RW	**SRWECA1 portmap selection**
	SRWECA1 address remap selection, default configured to MPAHBRAM1			
7:6	SRWECA0PORTSEL	0x0	RW	**SRWECA0 portmap selection**
	SRWECA0 address remap selection, default configured to MPAHBRAM0			
5:4	AHBSRWPORTSEL	0x0	RW	**AHBSRW portmap selection**
	AHBSRW address remap selection, default configured to MPAHBRAM0			
3:2	SRWAESPORTSEL	0x1	RW	**SRWAES portmap selection**
	SRWAES address remap selection, default configured to MPAHBRAM1			
1:0	LDMAPORTSEL	0x1	RW	**LDMA portmap selection**
	LDMA address remap selection, default configured to MPAHBRAM1			

5.6.6.12 SYSCFG_ROOTDATA0 - Data Register 0

Offset	Bit Position																													
0x600	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	DATA																															

Bit	Name	Reset	Access	Description
31:0	DATA	0x0	RW	**Data**
	Generic data space for user to pass to root, e.g., address of struct in mem			

5.6.6.13 SYSCFG_ROOTDATA1 - Data Register 1

Offset	Bit Position																													
0x604	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	DATA																															

Bit	Name	Reset	Access	Description
31:0	DATA	0x0	RW	**Data**
	Generic data space for user to pass to root, e.g., address of struct in mem			

5.6.6.14 SYSCFG_ROOTLOCKSTATUS - Lock Status

Offset	Bit Position																														
0x608	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0											0x1	0x1	0x1	0x1	0x1								0x1						0x1	0x1	0x1
Access	R											R	R	R	R	R								R						R	R	R
Name	EFUSEUNLOCKED											USERSPNIDLOCK	USERSPIDLOCK	USERNIDLOCK	USERDBGLOCK	USERDBGAPLOCK								ROOTDBGLOCK						MFRLOCK	REGLOCK	BUSLOCK

Bit	Name	Reset	Access	Description
31	EFUSEUNLOCKED	0x0	R	**E-Fuse Unlocked**
	E-Fuse Unlocked when 1; Locked when 0.			
30:21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20	USERSPNIDLOCK	0x1	R	**User Secure Non-invasive Debug Lock**
	Locked when 1; unlocked when 0.			
19	USERSPIDLOCK	0x1	R	**User Secure Invasive Debug Lock**
	Locked when 1; unlocked when 0.			
18	USERNIDLOCK	0x1	R	**User Non-invasive Debug Lock**
	Locked when 1; unlocked when 0.			
17	USERDBGLOCK	0x1	R	**User Invasive Debug Lock**
	Locked when 1; unlocked when 0.			
16	USERDBGAPLOCK	0x1	R	**User Debug Access Port Lock**
	Locked when 1; unlocked when 0.			
15:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	ROOTDBGLOCK	0x1	R	**Root Debug Lock**
	Locked when 1; unlocked when 0.			
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	MFRLOCK	0x1	R	**Manufacture Lock**
	Locked when 1; unlocked when 0.			
1	REGLOCK	0x1	R	**Register Lock**
	Locked when 1; unlocked when 0.			
0	BUSLOCK	0x1	R	**Bus Lock**
	Locked when 1; unlocked when 0.			

5.6.6.15 SYSCFG_ROOTSESWVERSION - SE SW Version

Offset	Bit Position																													
0x60C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	SWVERSION																															

Bit	Name	Reset	Access	Description
31:0	SWVERSION	0x0	RW	**SW Version**
	Location for SE to write the Firmware version and host to read it			

5.6.7 MPAHBRAM Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	MPAHBRAM_IPVERSION	R	IP Version ID
0x004	MPAHBRAM_CMD	W	Command Register
0x008	MPAHBRAM_CTRL	RW	Control Register
0x00C	MPAHBRAM_ECCERRADDR0	RH	ECC Error Address 0
0x010	MPAHBRAM_ECCERRADDR1	RH	ECC Error Address 1
0x014	MPAHBRAM_ECCERRADDR2	RH	ECC Error Address 2
0x018	MPAHBRAM_ECCERRADDR3	RH	ECC Error Address 3
0x01C	MPAHBRAM_ECCMERRIND	RH	Multiple ECC Error Indication
0x020	MPAHBRAM_IF	RWH INTFLAG	Interrupt Flags
0x024	MPAHBRAM_IEN	RW	Interrupt Enable
0x1000	MPAHBRAM_IPVERSION_SET	R	IP Version ID
0x1004	MPAHBRAM_CMD_SET	W	Command Register
0x1008	MPAHBRAM_CTRL_SET	RW	Control Register
0x100C	MPAHBRAM_ECCER-RADDR0_SET	RH	ECC Error Address 0
0x1010	MPAHBRAM_ECCER-RADDR1_SET	RH	ECC Error Address 1
0x1014	MPAHBRAM_ECCER-RADDR2_SET	RH	ECC Error Address 2
0x1018	MPAHBRAM_ECCER-RADDR3_SET	RH	ECC Error Address 3
0x101C	MPAHBRAM_ECCMER-RIND_SET	RH	Multiple ECC Error Indication
0x1020	MPAHBRAM_IF_SET	RWH INTFLAG	Interrupt Flags
0x1024	MPAHBRAM_IEN_SET	RW	Interrupt Enable
0x2000	MPAHBRAM_IPVERSION_CLR	R	IP Version ID
0x2004	MPAHBRAM_CMD_CLR	W	Command Register
0x2008	MPAHBRAM_CTRL_CLR	RW	Control Register
0x200C	MPAHBRAM_ECCER-RADDR0_CLR	RH	ECC Error Address 0
0x2010	MPAHBRAM_ECCER-RADDR1_CLR	RH	ECC Error Address 1
0x2014	MPAHBRAM_ECCER-RADDR2_CLR	RH	ECC Error Address 2
0x2018	MPAHBRAM_ECCER-RADDR3_CLR	RH	ECC Error Address 3
0x201C	MPAHBRAM_ECCMER-RIND_CLR	RH	Multiple ECC Error Indication
0x2020	MPAHBRAM_IF_CLR	RWH INTFLAG	Interrupt Flags

Offset	Name	Type	Description
0x2024	MPAHBRAM_IEN_CLR	RW	Interrupt Enable
0x3000	MPAHBRAM_IPVERSION_TGL	R	IP Version ID
0x3004	MPAHBRAM_CMD_TGL	W	Command Register
0x3008	MPAHBRAM_CTRL_TGL	RW	Control Register
0x300C	MPAHBRAM_ECCER-RADDR0_TGL	RH	ECC Error Address 0
0x3010	MPAHBRAM_ECCER-RADDR1_TGL	RH	ECC Error Address 1
0x3014	MPAHBRAM_ECCER-RADDR2_TGL	RH	ECC Error Address 2
0x3018	MPAHBRAM_ECCER-RADDR3_TGL	RH	ECC Error Address 3
0x301C	MPAHBRAM_ECCMER-RIND_TGL	RH	Multiple ECC Error Indication
0x3020	MPAHBRAM_IF_TGL	RWH INTFLAG	Interrupt Flags
0x3024	MPAHBRAM_IEN_TGL	RW	Interrupt Enable

5.6.8 MPAHBRAM Register Description

5.6.8.1 MPAHBRAM_IPVERSION - IP Version ID

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x2	
Access																															R	
Name																															IPVERSION	

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	IPVERSION	0x2	R	**New BitField**

5.6.8.2 MPAHBRAM_CMD - Command Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													W(nB)	W(nB)	W(nB)	W(nB)
Name																													CLEARECCADDR3	CLEARECCADDR2	CLEARECCADDR1	CLEARECCADDR0

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	CLEARECCADDR3	0x0	W(nB)	**Clear ECCERRADDR3**
	Write 1 to clear ECCERRADDR3 and ECCMERRIND.P3.			
2	CLEARECCADDR2	0x0	W(nB)	**Clear ECCERRADDR2**
	Write 1 to clear ECCERRADDR2 and ECCMERRIND.P2.			
1	CLEARECCADDR1	0x0	W(nB)	**Clear ECCERRADDR1**
	Write 1 to clear ECCERRADDR1 and ECCMERRIND.P1.			
0	CLEARECCADDR0	0x0	W(nB)	**Clear ECCERRADDR0**
	Write 1 to clear ECCERRADDR0 and ECCMERRIND.P0.			

5.6.8.3 MPAHBRAM_CTRL - Control Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																										0x1		0x0		0x0	0x0	0x0
Access																										RW		RW		RW	RW	RW
Name																										ADDRFAULTEN		AHBPORTPRIORITY		ECCERRFAULTEN	ECCWEN	ECCEN

Bit	Name	Reset	Access	Description
31:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	ADDRFAULTEN	0x1	RW	**Address fault bus fault enable**
	Enable bus fault upon address fault			
5:3	AHBPORTPRIORITY	0x0	RW	**AHB port arbitration priority**

Value	Mode	Description
0	NONE	No AHB port have raised priority.
1	PORT0	AHB port 0 has raised priority.
2	PORT1	AHB port 1 has raised priority.
3	PORT2	AHB port 2 has raised priority.
4	PORT3	AHB port 3 has raised priority.

2	ECCERRFAULTEN	0x0	RW	**ECC Error bus fault enable**
	Enable bus fault upon 2 bit ECC error			
1	ECCWEN	0x0	RW	**Enable ECC syndrome writes**
0	ECCEN	0x0	RW	**Enable ECC functionality**

5.6.8.4 MPAHBRAM_ECCERRADDR0 - ECC Error Address 0

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	ADDR																															

Bit	Name	Reset	Access	Description
31:0	ADDR	0x0	R	**ECC Error Address**
	Last captured ECC error address on AHB port 0. Cleared by CMD.CLEARECCADDR0			

5.6.8.5 MPAHBRAM_ECCERRADDR1 - ECC Error Address 1

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	ADDR																															

Bit	Name	Reset	Access	Description
31:0	ADDR	0x0	R	**ECC Error Address**
	Last captured ECC error address on AHB port 1. Cleared by CMD.CLEARECCADDR1			

5.6.8.6 MPAHBRAM_ECCERRADDR2 - ECC Error Address 2

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	ADDR																															

Bit	Name	Reset	Access	Description
31:0	ADDR	0x0	R	**ECC Error Address**
	Last captured ECC error address on AHB port 2. Cleared by CMD.CLEARECCADDR2			

5.6.8.7 MPAHBRAM_ECCERRADDR3 - ECC Error Address 3

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	ADDR																															

Bit	Name	Reset	Access	Description
31:0	ADDR	0x0	R	**ECC Error Address**
	Last captured ECC error address on AHB port 3. Cleared by CMD.CLEARECCADDR3			

5.6.8.8 MPAHBRAM_ECCMERRIND - Multiple ECC Error Indication

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													R	R	R	R
Name																													P3	P2	P1	P0

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	P3	0x0	R	**Multiple ECC errors on AHB port 2**
	Multiple ECC error indication on AHB port 3. Cleared by CMD.CLEARECCADDR3.			
2	P2	0x0	R	**Multiple ECC errors on AHB port 2**
	Multiple ECC error indication on AHB port 2. Cleared by CMD.CLEARECCADDR2.			
1	P1	0x0	R	**Multiple ECC errors on AHB port 1**
	Multiple ECC error indication on AHB port 1. Cleared by CMD.CLEARECCADDR1.			
0	P0	0x0	R	**Multiple ECC errors on AHB port 0**
	Multiple ECC error indication on AHB port 0. Cleared by CMD.CLEARECCADDR0.			

5.6.8.9 MPAHBRAM_IF - Interrupt Flags

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																									RW	RW	RW	RW	RW	RW	RW	RW
Name																									AHB3ERR2B	AHB2ERR2B	AHB1ERR2B	AHB0ERR2B	AHB3ERR1B	AHB2ERR1B	AHB1ERR1B	AHB0ERR1B

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	AHB3ERR2B	0x0	RW	**AHB3 2-bit ECC Error Interrupt Flag**
6	AHB2ERR2B	0x0	RW	**AHB2 2-bit ECC Error Interrupt Flag**
5	AHB1ERR2B	0x0	RW	**AHB1 2-bit ECC Error Interrupt Flag**
4	AHB0ERR2B	0x0	RW	**AHB0 2-bit ECC Error Interrupt Flag**
3	AHB3ERR1B	0x0	RW	**AHB3 1-bit ECC Error Interrupt Flag**
2	AHB2ERR1B	0x0	RW	**AHB2 1-bit ECC Error Interrupt Flag**
1	AHB1ERR1B	0x0	RW	**AHB1 1-bit ECC Error Interrupt Flag**
0	AHB0ERR1B	0x0	RW	**AHB0 1-bit ECC Error Interrupt Flag**

5.6.8.10 MPAHBRAM_IEN - Interrupt Enable

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																									RW	RW	RW	RW	RW	RW	RW	RW
Name																									AHB3ERR2B	AHB2ERR2B	AHB1ERR2B	AHB0ERR2B	AHB3ERR1B	AHB2ERR1B	AHB1ERR1B	AHB0ERR1B

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	AHB3ERR2B	0x0	RW	**AHB3 2-bit ECC Error Interrupt Enable**
6	AHB2ERR2B	0x0	RW	**AHB2 2-bit ECC Error Interrupt Enable**
5	AHB1ERR2B	0x0	RW	**AHB1 2-bit ECC Error Interrupt Enable**
4	AHB0ERR2B	0x0	RW	**AHB0 2-bit ECC Error Interrupt Enable**
3	AHB3ERR1B	0x0	RW	**AHB3 1-bit ECC Error Interrupt Enable**
2	AHB2ERR1B	0x0	RW	**AHB2 1-bit ECC Error Interrupt Enable**
1	AHB1ERR1B	0x0	RW	**AHB1 1-bit ECC Error Interrupt Enable**
0	AHB0ERR1B	0x0	RW	**AHB0 1-bit ECC Error Interrupt Enable**

5.7 MSC Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	MSC_IPVERSION	R	IP Version ID
0x004	MSC_READCTRL	RW	Read Control Register
0x008	MSC_RDATACTRL	RW	Read Data Control Register
0x00C	MSC_WRITECTRL	RW	Write Control Register
0x010	MSC_WRITECMD	W	Write Command Register
0x014	MSC_ADDRB	RWH	Page Erase/Write Address Buffer
0x018	MSC_WDATA	RWH	Write Data Register
0x01C	MSC_STATUS	RH	Status Register
0x020	MSC_IF	RWH	Interrupt Flag Register
0x024	MSC_IEN	RW	Interrupt Enable Register
0x034	MSC_USERDATASIZE	R	User Data Region Size Register
0x038	MSC_CMD	W	Command Register
0x03C	MSC_LOCK	W	Configuration Lock Register
0x040	MSC_MISCLOCKWORD	RW	Mass Erase and User Data Page Lock Word
0x050	MSC_PWRCTRL	RW	Power Control Register
0x120	MSC_PAGELOCK0	RW	Main Space Page 0-31 Lock Word
0x124	MSC_PAGELOCK1	RW	Main Space Page 32-63 Lock Word
0x128	MSC_PAGELOCK2	RW	Main Space Page 64-95 Lock Word
0x12C	MSC_PAGELOCK3	RW	Main Space Page 96-127 Lock Word
0x1000	MSC_IPVERSION_SET	R	IP Version ID
0x1004	MSC_READCTRL_SET	RW	Read Control Register
0x1008	MSC_RDATACTRL_SET	RW	Read Data Control Register
0x100C	MSC_WRITECTRL_SET	RW	Write Control Register
0x1010	MSC_WRITECMD_SET	W	Write Command Register
0x1014	MSC_ADDRB_SET	RWH	Page Erase/Write Address Buffer
0x1018	MSC_WDATA_SET	RWH	Write Data Register
0x101C	MSC_STATUS_SET	RH	Status Register
0x1020	MSC_IF_SET	RWH	Interrupt Flag Register
0x1024	MSC_IEN_SET	RW	Interrupt Enable Register
0x1034	MSC_USERDATASIZE_SET	R	User Data Region Size Register
0x1038	MSC_CMD_SET	W	Command Register
0x103C	MSC_LOCK_SET	W	Configuration Lock Register
0x1040	MSC_MISCLOCKWORD_SET	RW	Mass Erase and User Data Page Lock Word
0x1050	MSC_PWRCTRL_SET	RW	Power Control Register
0x1120	MSC_PAGELOCK0_SET	RW	Main Space Page 0-31 Lock Word

Offset	Name	Type	Description
0x1124	MSC_PAGELOCK1_SET	RW	Main Space Page 32-63 Lock Word
0x1128	MSC_PAGELOCK2_SET	RW	Main Space Page 64-95 Lock Word
0x112C	MSC_PAGELOCK3_SET	RW	Main Space Page 96-127 Lock Word
0x2000	MSC_IPVERSION_CLR	R	IP Version ID
0x2004	MSC_READCTRL_CLR	RW	Read Control Register
0x2008	MSC_RDATACTRL_CLR	RW	Read Data Control Register
0x200C	MSC_WRITECTRL_CLR	RW	Write Control Register
0x2010	MSC_WRITECMD_CLR	W	Write Command Register
0x2014	MSC_ADDRB_CLR	RWH	Page Erase/Write Address Buffer
0x2018	MSC_WDATA_CLR	RWH	Write Data Register
0x201C	MSC_STATUS_CLR	RH	Status Register
0x2020	MSC_IF_CLR	RWH	Interrupt Flag Register
0x2024	MSC_IEN_CLR	RW	Interrupt Enable Register
0x2034	MSC_USERDATASIZE_CLR	R	User Data Region Size Register
0x2038	MSC_CMD_CLR	W	Command Register
0x203C	MSC_LOCK_CLR	W	Configuration Lock Register
0x2040	MSC_MISCLOCKWORD_CLR	RW	Mass Erase and User Data Page Lock Word
0x2050	MSC_PWRCTRL_CLR	RW	Power Control Register
0x2120	MSC_PAGELOCK0_CLR	RW	Main Space Page 0-31 Lock Word
0x2124	MSC_PAGELOCK1_CLR	RW	Main Space Page 32-63 Lock Word
0x2128	MSC_PAGELOCK2_CLR	RW	Main Space Page 64-95 Lock Word
0x212C	MSC_PAGELOCK3_CLR	RW	Main Space Page 96-127 Lock Word
0x3000	MSC_IPVERSION_TGL	R	IP Version ID
0x3004	MSC_READCTRL_TGL	RW	Read Control Register
0x3008	MSC_RDATACTRL_TGL	RW	Read Data Control Register
0x300C	MSC_WRITECTRL_TGL	RW	Write Control Register
0x3010	MSC_WRITECMD_TGL	W	Write Command Register
0x3014	MSC_ADDRB_TGL	RWH	Page Erase/Write Address Buffer
0x3018	MSC_WDATA_TGL	RWH	Write Data Register
0x301C	MSC_STATUS_TGL	RH	Status Register
0x3020	MSC_IF_TGL	RWH	Interrupt Flag Register
0x3024	MSC_IEN_TGL	RW	Interrupt Enable Register
0x3034	MSC_USERDATASIZE_TGL	R	User Data Region Size Register
0x3038	MSC_CMD_TGL	W	Command Register
0x303C	MSC_LOCK_TGL	W	Configuration Lock Register
0x3040	MSC_MISCLOCKWORD_TGL	RW	Mass Erase and User Data Page Lock Word
0x3050	MSC_PWRCTRL_TGL	RW	Power Control Register

Offset	Name	Type	Description
0x3120	MSC_PAGELOCK0_TGL	RW	Main Space Page 0-31 Lock Word
0x3124	MSC_PAGELOCK1_TGL	RW	Main Space Page 32-63 Lock Word
0x3128	MSC_PAGELOCK2_TGL	RW	Main Space Page 64-95 Lock Word
0x312C	MSC_PAGELOCK3_TGL	RW	Main Space Page 96-127 Lock Word

5.8 MSC Register Description

5.8.1 MSC_IPVERSION - IP Version ID

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x7																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x7	R	**IP Version ID**
	IP Version ID			

5.8.2 MSC_READCTRL - Read Control Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x2																				
Access											RW																				
Name											MODE																				

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:20	MODE	0x2	RW	**Read Mode**

When changing to a higher frequency, this register must be set to a large number of wait states before the core clock is switched to the higher frequency. When changing to a lower frequency, this register should be set to a lower number of wait states after the frequency transition has been completed. The maximum frequency for each wait state setting is listed in the datasheet.

Value	Mode	Description
0	WS0	Zero wait-states inserted in fetch or read transfers
1	WS1	One wait-state inserted for each fetch or read transfer. See Flash Wait-States table for details
2	WS2	Two wait-states inserted for eatch fetch or read transfer. See Flash Wait-States table for details
3	WS3	Three wait-states inserted for eatch fetch or read transfer. See Flash Wait-States table for details

19:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

5.8.3 MSC_RDATACTRL - Read Data Control Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																				0x1											0x0	
Access																				RW											RW	
Name																				DOUTBUFEN											AFDIS	

Bit	Name	Reset	Access	Description
31:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	DOUTBUFEN	0x1	RW	**Flash dout pipeline buffer enable**
	Flag to enable or bypass flash data output buffer			
11:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	AFDIS	0x0	RW	**Automatic Invalidate Disable**
	When this bit is set the cache is not automatically invalidated when a write or page erase is performed.			
0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

5.8.4 MSC_WRITECTRL - Write Control Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0																		0x0		0x0	0x0
Access											RW																		RW		RW	RW
Name											RANGECOUNT																		LPWRITE		IRQERASEABORT	WREN

Bit	Name	Reset	Access	Description
31:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25:16	RANGECOUNT	0x0	RW	**EraseRange Count**
	Apply only with EraseRange command. Define number of pages to be erased.			
15:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	LPWRITE	0x0	RW	**Low-Power Write**
	When set, write times might double while reducing current consumption			
2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	IRQERASEABORT	0x0	RW	**Abort Page Erase on Interrupt**
	When this bit is set to 1, any Cortex-M33 interrupt aborts any current page erase operation. Executing that interrupt vector from Flash will halt the CPU.			
0	WREN	0x0	RW	**Enable Write/Erase Controller**
	When this bit is set, the MSC write and erase functionality is enabled			

5.8.5 MSC_WRITECMD - Write Command Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																				0x0				0x0			0x0	0x0		0x0	0x0	
Access																				W				W			W	W		W	W	
Name																				CLEARWDATA				ERASEMAIN0			ERASEABORT	ERASERANGE		WRITEEND	ERASEPAGE	

Bit	Name	Reset	Access	Description
31:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	CLEARWDATA	0x0	W	**Clear WDATA state**
	Will set WDATAREADY and DMA request. Should only be used when no write is active.			
11:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	ERASEMAIN0	0x0	W	**Mass erase region 0**
	Initiate mass erase of flash memory. If MELOCKBIT in MSC_MISCLOCKWORD is set, user firmware cannot initiate mass erase, and only the SE may initiate mass erase.			
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	ERASEABORT	0x0	W	**Abort erase sequence**
	Writing to this bit will abort an ongoing erase sequence.			
4	ERASERANGE	0x0	W	**Erase range of pages**
	Erase a range of user defined pages started from the MSC_ADDRB register. The WREN bit in the MSC_WRITECTRL register must be set in order to use this command.			
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	WRITEEND	0x0	W	**End Write Mode**
	Write 1 to abort a write command.			
1	ERASEPAGE	0x0	W	**Erase Page**
	Erase any user defined page selected by the MSC_ADDRB register. The WREN bit in the MSC_WRITECTRL register must be set in order to use this command.			
0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

5.8.6 MSC_ADDRB - Page Erase/Write Address Buffer

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	ADDRB																															

Bit	Name	Reset	Access	Description
31:0	ADDRB	0x0	RW	**Page Erase or Write Address Buffer**
	This register holds the system address for the erase or write operation. Address should be word aligned address.The MSB bit is not ignored for ADDRB			

5.8.7 MSC_WDATA - Write Data Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	DATAW																															

Bit	Name	Reset	Access	Description
31:0	DATAW	0x0	RW	**Write Data**
	The data to be written to the address in MSC_ADDRB. This register must be written when the WDATAREADY bit of MSC_STATUS is set.This register does not support write mask.			

5.8.8 MSC_STATUS - Status Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0				0x1			0x0								0x0									0x0	0x0	0x0	0x0	0x1	0x0	0x0	0x0
Access	R				R			R								R									R	R	R	R	R	R	R	R
Name	PWRUPCKBDFAILCOUNT				WREADY			PWRON								REGLOCK									RANGEPARTIAL	TIMEOUT	PENDING	ERASEABORTED	WDATAREADY	INVADDR	LOCKED	BUSY

Bit	Name	Reset	Access	Description
31:28	PWRUPCKBDFAIL-COUNT	0x0	R	**Flash power up checkerboard pattern chec**
	This field tells how many times checkboard pattern check fail occured after a reset sequence.			
27	WREADY	0x1	R	**Flash Write Ready**
	When this bit is set, flash has completed the power up sequence and is ready for write/erase commands.			
26:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24	PWRON	0x0	R	**Flash Power On Status**
	When this bit is set, flash is powered on. If zero, flash is powered off and reads from flash return indeterminate data.			
23:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	REGLOCK	0x0	R	**Register Lock Status**
	Indicates the current status of register lock			

Value	Mode	Description
0	UNLOCKED	
1	LOCKED	

15:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	RANGEPARTIAL	0x0	R	**EraseRange with skipped locked pages**
6	TIMEOUT	0x0	R	**Write Command Timeout**
	When this bit is set, it indicates that the last write command has completed due to a write buffer timeout. This bit is cleared automatically when a new write command is initiated.			
5	PENDING	0x0	R	**Write Command In Queue**
	When this bit is set, a flash operation has been requested but not yet started. New commands are ignored when PENDING is set.			
4	ERASEABORTED	0x0	R	**Erase Operation Aborted**

Bit	Name	Reset	Access	Description
	When MSC_WRITECTRL_IRQERASEABORT = 1, this bit is set because an interrupt has aborted the erase operation in progress.			
3	WDATAREADY	0x1	R	**WDATA Write Ready**
	When this bit is set, the content of MSC_WDATA is read by MSC Flash Write Controller and the register may be updated with the next 32-bit word to be written to flash. This bit is cleared when writing to MSC_WDATA.			
2	INVADDR	0x0	R	**Invalid Write Address or Erase Page**
	When this bit is set, software has attempted to load an invalid (unmapped) address into the MSC_ADDRB register.			
1	LOCKED	0x0	R	**Access Locked**
	When set, the last erase or write was aborted due to erase/write access constraints.			
0	BUSY	0x0	R	**Erase/Write Busy**
	When set, an erase or write operation is in progress and new commands are ignored.			

5.8.9 MSC_IF - Interrupt Flag Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0	0x0						0x0	0x0	0x0
Access																							RW	RW						RW	RW	RW
Name																							PWROFF	PWRUPF						WDATAOV	WRITE	ERASE

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	PWROFF	0x0	RW	**Flash Power Off Sequence Complete Flag**
	Set after MSC_CMD.PWROFF received, flash powered off complete			
8	PWRUPF	0x0	RW	**Flash Power Up Sequence Complete Flag**
	Set after MSC_CMD.PWRUP received, flash powered up complete and ready for read/write			
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	WDATAOV	0x0	RW	**Host write buffer overflow**
	If set, flash controller write buffer overflow detected			
1	WRITE	0x0	RW	**Host Write Done Interrupt Read Flag**
	Set when a write is done			
0	ERASE	0x0	RW	**Host Erase Done Interrupt Read Flag**
	Set when erase is done			

5.8.10 MSC_IEN - Interrupt Enable Register

Offset	Bit Position																														
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0	0x0						0x0	0x0	0x0
Access																							RW	RW						RW	RW	RW
Name																							PWROFF	PWRUPF						WDATAOV	WRITE	ERASE

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	PWROFF interrupt enable	0x0	RW	**Flash Power Off Seq done irq enable**
8	PWRUPF interrupt enable	0x0	RW	**Flash Power Up Seq done irq enable**
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	WDATAOV interrupt enable	0x0	RW	**write data buffer overflow irq enable**
1	WRITE interrupt enable	0x0	RW	**Write Done Interrupt enable**
0	ERASE interrupt enable	0x0	RW	**Erase Done Interrupt enable**

5.8.11 MSC_USERDATASIZE - User Data Region Size Register

Offset	Bit Position																													
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x4				
Access																											R				
Name																											USERDATASIZE				

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:0	USERDATASIZE	0x4	R	**User Data Size**
	This field determines user data region size. SIZE = 256B * USERDATASIZE.			

5.8.12 MSC_CMD - Command Register

Offset	Bit Position																													
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0				0x0
Access																												W				W
Name																												PWROFF				PWRUP

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	PWROFF	0x0	W	**Flash power off/sleep command**
	Write to this bit to power down the Flash. User code should execute from RAM afterwards. Read from flash after flash being powered down will cause undetermined behavior. To power up, either set CMD.PWRUP bit or try read from flash.			
3:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	PWRUP	0x0	W	**Flash Power Up Command**
	Write to this bit to power up the Flash. IRQ PWRUPF will be fired when power up sequence completed.			

5.8.13 MSC_LOCK - Configuration Lock Register

Offset	Bit Position																															
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W							
Name																									LOCKKEY							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x0	W	**Configuration Lock**

Write any other value than the unlock code to lock access to MSC_RDATACTRL, MSC_READCTRL, and MSC_WRITECTRL. Write the unlock code to enable access. When reading the register, bit 0 is set when the lock is enabled.

Value	Mode	Description
0	LOCK	
7025	UNLOCK	

5.8.14 MSC_MISCLOCKWORD - Mass Erase and User Data Page Lock Word

Offset	Bit Position																															
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x1				0x1
Access																												RW				RW
Name																												UDLOCKBIT				MELOCKBIT

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	UDLOCKBIT	0x1	RW	**User Data Lock**

Zero means host can write/erase to the user data area. Host is only allowed to write one. Root and debug can clear this bit.

3:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	MELOCKBIT	0x1	RW	**Mass Erase Lock**

Zero means host can mass erase the main space. Host is only allowed to write one. Root and debug can clear this bit.

5.8.15 MSC_PWRCTRL - Power Control Register

Offset	Bit Position																													
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset												0x10																0x0			0x1	0x0
Access												RW																RW			RW	RW
Name												PWROFFDLY																PWROFFENTRYAGAIN			PWROFFONEM1PENTRY	PWROFFONEM1ENTRY

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:16	PWROFFDLY	0x10	RW	**Power down delay**
	Defines delay cycles before flash enters sleep mode. Works together with PWROFFENTRYAGAIN bit. The power off delay is 64 * PWROFFDLY bus clock cycles.			
15:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	PWROFFENTRYAGAIN	0x0	RW	**POWER down flash again in EM1/EM1p**
	If enabled, flash will enter sleep mode again when POWEROFFONEM1ENTRY/POWEROFFONEM1PENTRY is set and no flash activities occur for the time determined by PWROFFDLY.			
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	PWROFFONEM1PENTRY	0x1	RW	**Power down Flash macro when enter EM1P**
	If enabled, flash will be in sleep mode when entering EM1P (radio-only sleep).			
0	PWROFFONEM1ENTRY	0x0	RW	**Power down Flash macro when enter EM1**
	If enabled, flash will be in sleep mode when entering EM1.			

5.8.16 MSC_PAGELOCK0 - Main Space Page 0-31 Lock Word

Offset	Bit Position																													
0x120	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	LOCKBIT																															

Bit	Name	Reset	Access	Description
31:0	LOCKBIT	0x0	RW	**page lock bit**
	Zero means the corresponding page is allowed to write/erase. change to one will prevent corresponding page from write/erase. bit[0] for main space page 0, and bit[1] for page 1... bit[31] for page 31. Reset to zero. Host is only allowed to write one. Root and Debug are allowed to clear this register			

5.8.17 MSC_PAGELOCK1 - Main Space Page 32-63 Lock Word

Offset	Bit Position																													
0x124	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	LOCKBIT																															

Bit	Name	Reset	Access	Description
31:0	LOCKBIT	0x0	RW	**page lock bit**
	Zero means the corresponding page is allowed to write/erase. change to one will prevent corresponding page from write/erase. bit[0] for main space page 32, and bit[1] for page 33... bit[31] for page 63. Reset to zero. Host is only allowed to write one. Root and Debug are allowed to clear this register			

5.8.18 MSC_PAGELOCK2 - Main Space Page 64-95 Lock Word

Offset	Bit Position																															
0x128	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	LOCKBIT																															

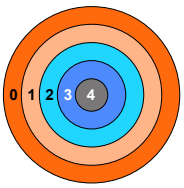
Bit	Name	Reset	Access	Description
31:0	LOCKBIT	0x0	RW	**page lock bit**
	Zero means the corresponding page is allowed to write/erase. change to one will prevent corresponding page from write/erase. bit[0] for main space page 64, and bit[1] for page 65... bit[31] for page 95. Reset to zero. Host is only allowed to write one. Root and Debug are allowed to clear this register			

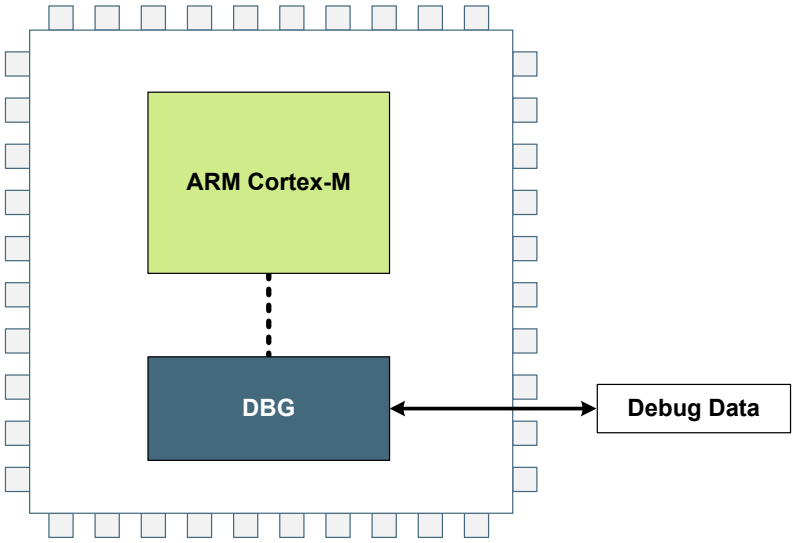
5.8.19 MSC_PAGELOCK3 - Main Space Page 96-127 Lock Word

Offset	Bit Position																															
0x12C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	LOCKBIT																															

Bit	Name	Reset	Access	Description
31:0	LOCKBIT	0x0	RW	**page lock bit**
	Zero means the corresponding page is allowed to write/erase. change to one will prevent corresponding page from write/erase. bit[0] for main space page 96, and bit[1] for page 97... bit[31] for page 127. Reset to zero. Host is only allowed to write one. Root and Debug are allowed to clear this register			

6. DBG - Debug Interface





Quick Facts

What?

The Debug Interface is used to program and debug EFM32PG28 devices.

Why?

The Debug Interface makes it easy to re-program and update the system in the field, and allows debugging with minimal I/O pin usage.

How?

The Cortex®-M33 supports advanced debugging features. EFM32PG28 devices can use a minimum of two port pins for debugging or programming. The internal and external state of the system can be examined with debug extensions supporting instruction or data access break and watch points.

6.1 Introduction

The EFM32PG28 devices include hardware debug support through a 2-pin serial-wire debug (SWD) interface or a 4-pin Joint Test Action Group (JTAG) interface, as well as an Embedded Trace Module (ETM) for data/instruction tracing. In addition, there is also a Serial Wire Viewer pin which can be used to output profiling information, data trace and software-generated messages.

For more technical information about the debug interface the reader is referred to:

- ARM Cortex®-M33 Technical Reference Manual
- ARM CoreSight Components Technical Reference Manual
- ARM Debug Interface v5 Architecture Specification
- IEEE Standard for Test Access Port and Boundary-Scan Architecture, IEEE 1149.1-2013

6.2 Features

- Debug Access Port Serial Wire JTAG (DAPSWJ)
  - Implements the ADIv5 debug interface
- ARM Trustzone
  - Enables secure debugging
- Breakpoint unit (BPU)
  - Implement up to 8 hardware breakpoints
- Data Watch point and Trace (DWT) unit
  - Implement up to 4 watch points, trigger resources and system profiling
- Instrumentation Trace Macrocell (ITM)
  - Application-driven trace source that supports printf style debugging
- Embedded Trace Macrocell v3.5 (ETM)
  - Real time instruction and data trace information of the processor
- Cross Trigger Interface (CTI)
  - Issues synchronous triggers based on system events
  - Can be used to generate IRQs or route to PRS signalling

6.3 Functional Description

There are debug and trace pins available on the device. Operation of these pins is described in the following sections.

6.3.1 Debug Pins

The following pins are the debug connections for the device:

- Serial Wire Clock Input and Test Clock Input (SWCLKTCK) (SWCLK) : This pin is enabled after power-up and has a built-in pull-down.
- Serial Wire Data Input/Output and Test Mode Select Input (SWDIOTMS) (SWDIO) : This pin is enabled after power-up and has a built-in pull-up.
- Test Data Output (TDO): This pin is assigned to JTAG functionality after power-up. However, it remains in high-Z state until the first valid JTAG command is received.
- Test Data Input (TDI): This pin is assigned to JTAG functionality after power-up. However, it remains in high-Z state until the first valid JTAG command is received. Once enabled, the pin has a built-in pull-up.
- Serial Wire Viewer (SWV): This pin is disabled after reset.

The debug pins have integrated pull devices that are enabled by default after a reset. Leaving them enabled may increase current consumption if the pins are connected to power or ground. The debug pins have enable bits in the GPIO_DBGROUTEPEN register; refer to the GPIO chapter for more details. Upon disabling the debug pins, debug contact with the device is lost once the DAPSWJ power request bits are deasserted. By default after a power cycle, the DAPSWJ is in JTAG mode. If during a debugging session the device is switched to SWD mode, a power cycle is needed to return to JTAG mode.

6.3.2 Embedded Trace Macrocell (ETM)

ETM makes it possible to non-intrusively trace both instruction and data from the processor in real time. Trace can be controlled through a set of triggering and filtering resources. The resources include 4 address comparators, 2 data value comparators, 2 counters, a context ID comparator and a sequencer. Before enabling the ETM, the CMU_TRACECLKCTRL register must be configured to select the desired trace clock source. (See the CMU chapter for details.)

The trace can be exported through a set of trace pins, which include:

- Trace Clock (TRACECLK): Functions as a sample clock for the trace. This pin is disabled after reset.
- Trace Data 0-3 (TRACEDATA0, TRACEDATA1, TRACEDATA2, TRACEDATA3): The trace data pins provide the compressed trace stream. These pins are disabled after reset.

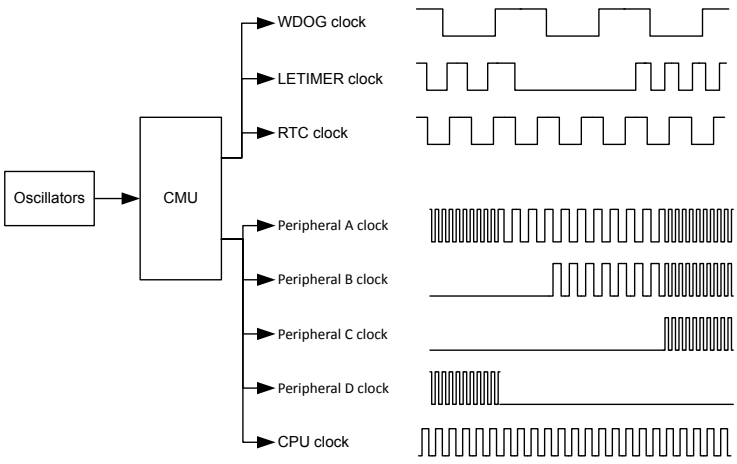
For information on how to configure the ETM, see the ARM Embedded Trace Macrocell Architecture Specification. The Trace Clock and Trace Data pins are enabled through a GPIO register. For more information on how to enable the ETM pins, refer to the GPIO chapter.

6.3.3 Debug and EM2/EM3

Debug connectivity in EM2 and EM3 is unavailable by default, to reduce current consumption. Debugging through EM2 and EM3 can be enabled by setting the EM2DBGEN bit in the EMU_CTRL register. Setting EM2DBGEN ensures that power domain associated with the debug circuitry will remain active, but will result in a small amount of additional current in EM2 and EM3.

Leaving the debugger connected when issuing a WFI or WFE to enter EM2 or EM3 will make the system enter a special EM2 mode. This mode differs from regular EM2 and EM3 in that the high frequency clocks are still enabled, and certain core functionality is still powered in order to maintain debug functionality. Because of this, the current consumption in this mode is closer to EM1, and it is, therefore, important to deassert the power requests in the DAPSWJ and disconnect the debugger before undertaking current consumption measurements.

7. CMU - Clock Management Unit



Quick Facts

What?

The CMU controls clock switching and distribution. EFM32PG28 supports several different oscillators with minimized power consumption and short start-up time. The CMU has HW support for calibration of RC oscillators.

Why?

Oscillators and clocks contribute significantly to the power consumption of the MCU. With the low power oscillators combined with the flexible clock control scheme, it is possible to minimize the energy consumption in any given application.

How?

The CMU switches different clock sources for various peripherals and sets the prescaler for the bus clocks. The short oscillator start-up times makes duty-cycling between active mode and the different low energy modes (EM2 DeepSleep, EM3 Stop, and EM4) very efficient. The calibration feature ensures high accuracy RC oscillators. Interrupts are available to avoid CPU polling of flags.

7.1 Introduction

The Clock Management Unit (CMU) is responsible for switching among various oscillator sources and provides clocks to the peripheral modules. Oscillators are automatically turned on and off based on demand from the peripherals to minimize power consumption.

7.2 Features

- Multiple clock sources available:
  - 38 MHz - 40 MHz High Frequency Crystal Oscillator (HFXO)
  - 1 MHz - 80 MHz High Frequency RC Oscillator (HFRCO0/HFRCODPLL)
  - 1 MHz - 40 MHz Deep Sleep High Frequency RC Oscillator (HFRCOEM23)
  - 20 MHz Fast Startup RC Oscillator (FSRCO)
  - 1 MHz - 38 MHz External Clock from Input Pins (CLKIN0)
  - 32768 Hz Low Frequency Crystal Oscillator (LFXO)
  - 32768 Hz Low Frequency RC Oscillator (LFRCO)
  - 1000 Hz Ultra Low Frequency RC Oscillator (ULFRCO)
- On-demand oscillator request.
- Low power oscillators.
- Fast start-up times.
- Cascaded prescalers for AHB Clocks (HCLK) and APB Clocks (PCLK).
- Clock gating on an individual basis to all peripherals based on module enable.
- Reset on an individual basis for Timer and IADC based on module enable.
- Selectable clocks can be output on external pins and/or PRS.
- Deep Sleep High Frequency RC oscillator (HFRCOEM23) or the Fast-start oscillator (FSRCO), which are asynchronous to the system clock, can be selected for IADC or VDAC0 operation in EM2.
- Hardware support for calibration of RC oscillators.

7.3 Functional Description

The CMU is comprised of several programmable clock trees, which connect oscillator resources to peripherals and buses. This section describes clock sources and peripherals available to the largest devices in the EFM32PG28 family. Please refer to the Configuration Summary in the Device Datasheet to see which core and peripheral modules, and therefore clock connections, are present in a specific device. Bus clock selection, including peripherals clocked directly from bus clocks, is shown in Figure 7.1 Bus Clocks on page 139. Clock selection for peripherals with multiple high-frequency clock sources is shown in Figure 7.2 High Frequency Peripheral Clocks on page 140. Clock selection for peripherals with multiple low-frequency clock sources is shown in Figure 7.3 Low Frequency Peripheral Clocks on page 141. Clock selection for peripherals that can select from a high or low frequency clock source is shown in Figure 7.4 Mixed Frequency Peripheral Clocks on page 142.

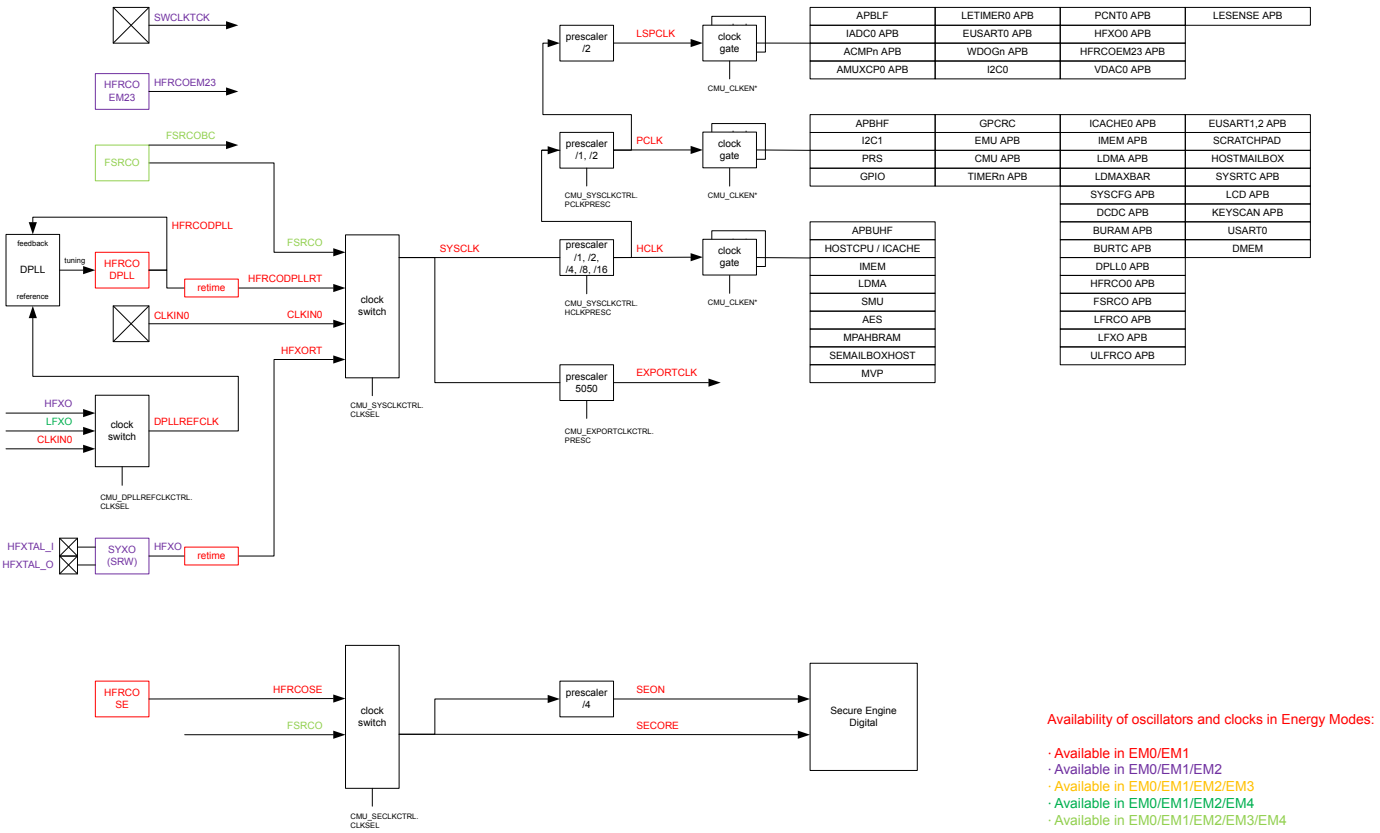


Figure 7.1. Bus Clocks

Availability of oscillators and clocks in Energy Modes:

· Available in EM0/EM1
· Available in EM0/EM1/EM2
· Available in EM0/EM1/EM2/EM3
· Available in EM0/EM1/EM2/EM4
· Available in EM0/EM1/EM2/EM3/EM4

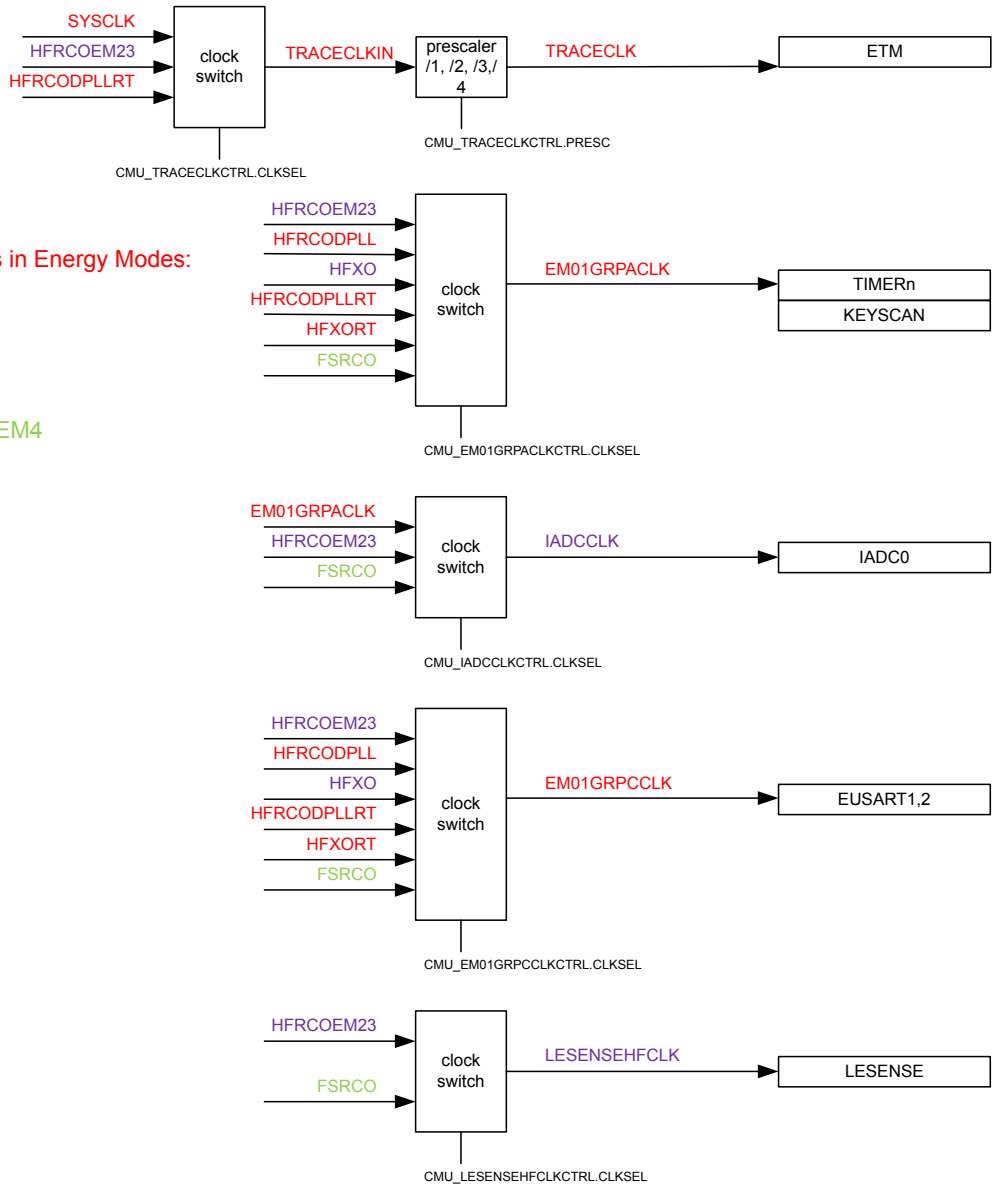


Figure 7.2. High Frequency Peripheral Clocks

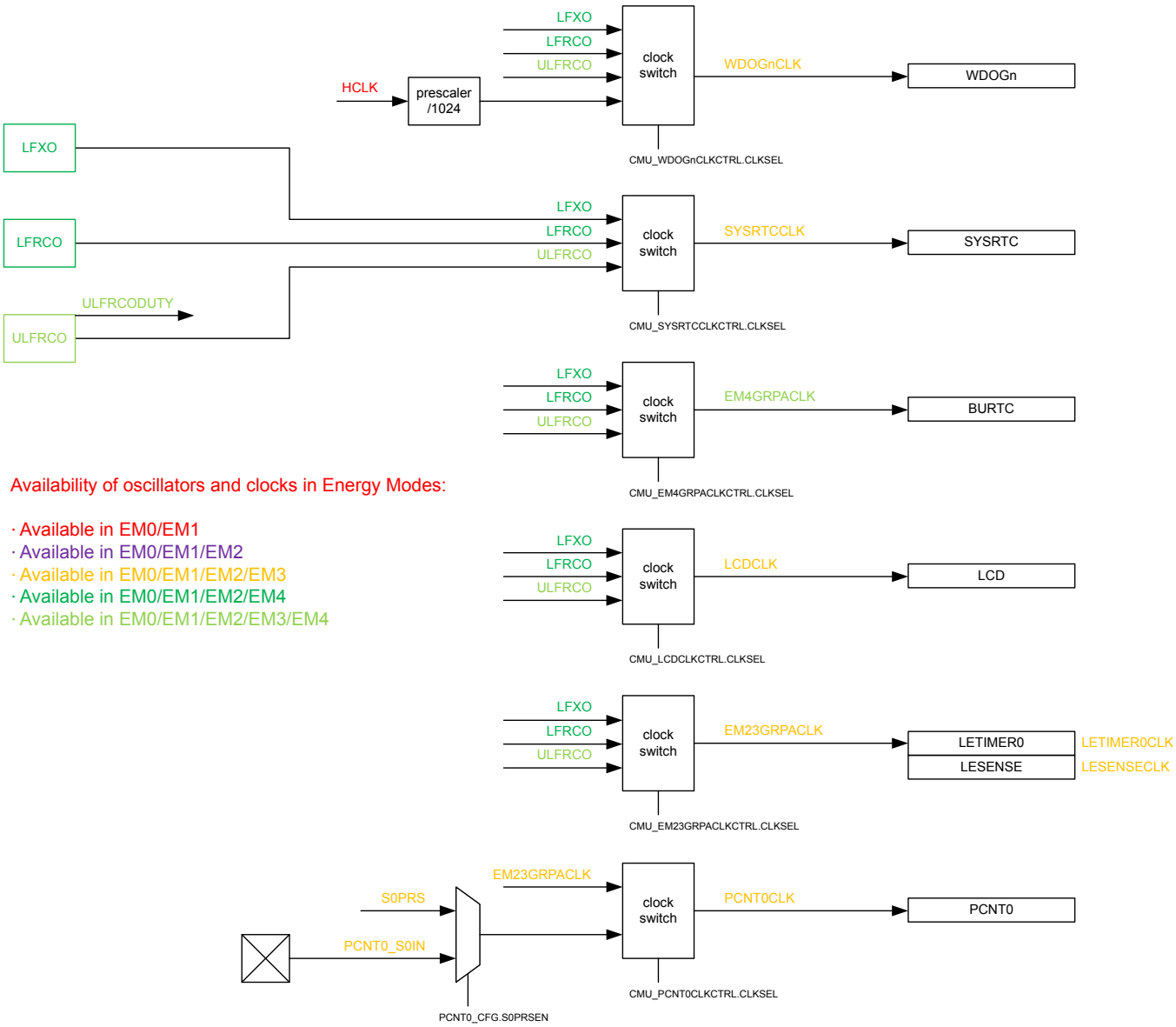


Figure 7.3. Low Frequency Peripheral Clocks

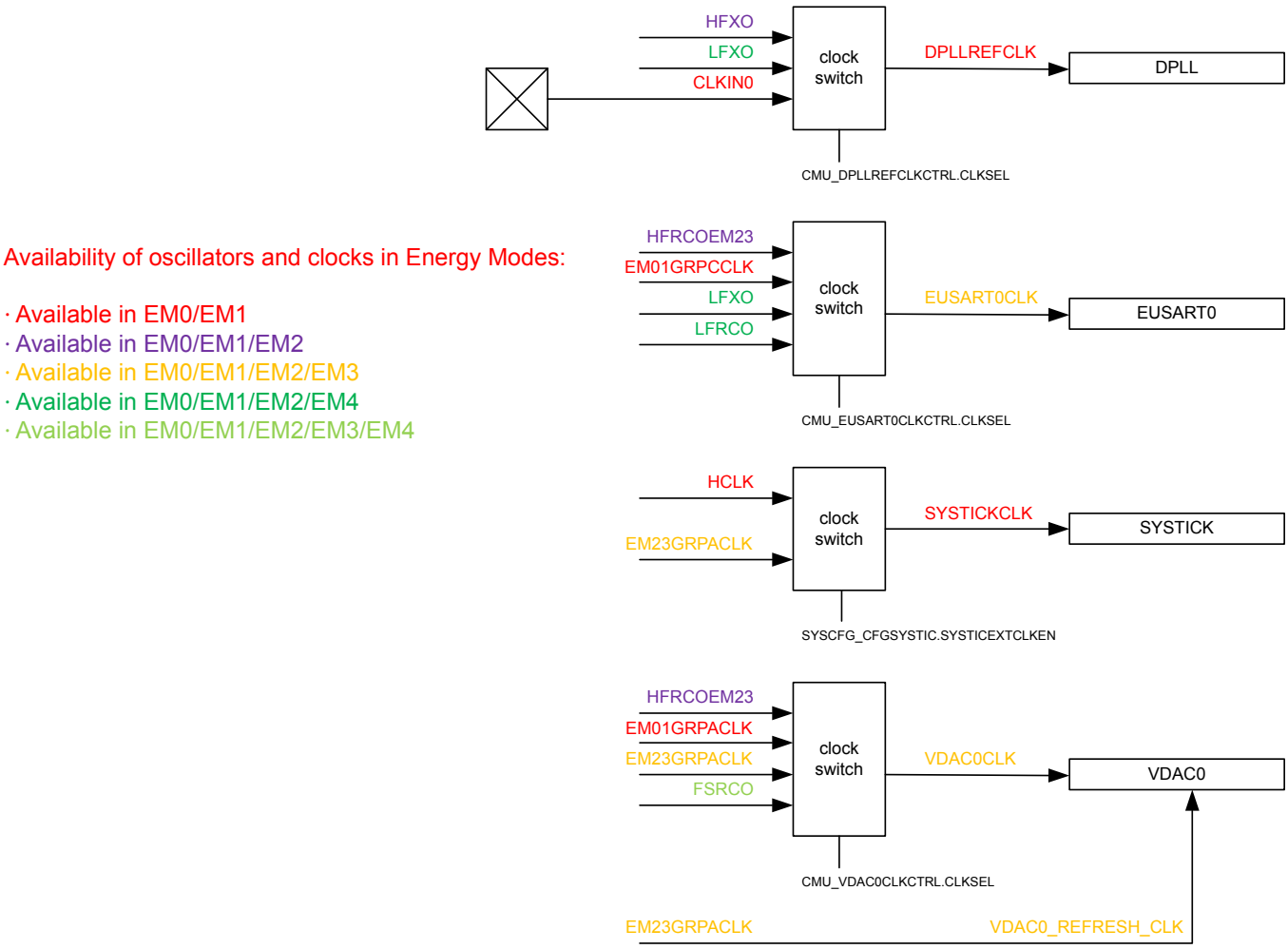


Figure 7.4. Mixed Frequency Peripheral Clocks

7.3.1 System Clocks

7.3.1.1 SYSCLK - Bus Clock

SYSCLK is the selected System Clock. HCLK is an optionally prescaled version of SYSCLK. PCLK is an optionally prescaled version of HCLK. The SYSCLK, and therefore HCLK and PCLK, can be driven by a high-frequency oscillator or be driven from a pin. The system boots using the FSRCO oscillator, and switches to HFRCODPLL before user firmware execution begins. To change the selected clock source, write to the CLKSEL bitfield in CMU_SYSCLKCTRL. If an invalid option is programmed into CLKSEL, FSRCO will be selected. The SYSCLK is running in EM0 Active and EM1 Sleep and is automatically stopped in EM2 DeepSleep.

The prescaler setting can be changed dynamically and the new setting takes effect immediately. When switching to a higher frequency oscillator source, prescaler setting should be adjusted before clock selection to prevent over clocking. For the same reason, when switching to a lower frequency oscillator source, prescaler setting cannot be adjusted until the clock selection is made.

7.3.1.2 HCLK - AHB Clock

HCLK is a prescaled version of SYSCLK. This clock drives the AHB bus interface. HCLK can be prescaled by setting HCLKPRESC in CMU_SYSCLKCTRL to DIV2 or DIV4. This prescales HCLK to all AHB bus clocks and is typically used to save energy in applications where the system is not required to run at the highest frequency. The setting can be changed dynamically and the new setting takes effect immediately. Some of the modules that are driven by this clock can be clock gated completely when not in use. This is done by clearing the module enable (EN) bit in the module's EN register.

7.3.1.3 PCLK - APB Clock

PCLK is a prescaled version of HCLK. This clock drives the APB bus interface. PCLK can be prescaled by setting PCLKPRESC in CMU_SYSCLKCTRL to DIV2. This prescales PCLK to all APB bus clocks and is necessary to prevent PCLK from exceeding the maximum frequency when HCLK is operated at above 40 MHz. The setting can be changed dynamically and the new setting takes effect immediately. Some of the peripherals that are driven by this clock can be clock gated completely when not in use. This is done by clearing the module enable (EN) bit in the module's EN register.

7.3.1.4 LSPCLK - Low Speed APB Clock

LSPCLK is a prescaled version of PCLK. This clock drives the Low Speed APB bus interface. LSPCLK is always prescaled by two. This prescales LSPCLK to all Low Speed APB bus clocks. Some of the peripherals that are driven by this clock can be clock gated completely when not in use. This is done by clearing the module enable (EN) bit in the module's EN register.

7.3.1.5 EM01GRPACLK - Energy Mode 01 Group A Clock

EM01GRPACLK is the selected clock for the Group A Peripherals operating in Energy Modes 0 or 1. These are typically high clock frequency peripheral modules. There are several selectable sources for EM01GRPACLK: HFXO, HFRCODPLL, HFRCOEM23, and FSRCO. In addition, the EM01GRPACLK can be disabled. The selection is configured using the CLKSEL field in CMU_EM01GRPACLKCTRL.

Each High Frequency Peripheral that is clocked by EM01GRPACLK may have its own prescaler setting and enable bit. The prescaler settings, if available, can be found in the peripheral's control registers. The enable bit can be found in the module's EN register.

7.3.1.6 EM01GRPCCLK - Energy Mode 01 Group C Clock

EM01GRPCCLK is the selected clock for the Group C Peripherals operating in Energy Modes 0 or 1. These are typically high clock frequency peripheral modules. There are several selectable sources for EM01GRPCCLK: HFXO, HFRCODPLL, HFRCOEM23, FSRCO, HFRCODPLLRT, and HFXORT. In addition, the EM01GRPCCLK can be disabled. The selection is configured using the CLKSEL field in CMU_EM01GRPCCLKCTRL.

Each High Frequency Peripheral that is clocked by EM01GRPCCLK may have its own prescaler setting and enable bit. The prescaler settings, if available, can be found in the peripheral's control registers. The enable bit can be found in the module's EN register.

7.3.1.7 EM23GRPACLK - Energy Mode 2 and 3 Group A Clock

EM23GRPACLK is the selected clock for the Group A Peripherals operating down to Energy Modes 2 or 3. These are typically low energy consumption peripheral modules. There are three selectable sources for EM23GRPACLK: LFRCO, LFXO and ULFRCO. In addition, the EM23GRPACLK can be disabled. The selection is configured using the CLKSEL field in CMU_EM23GRPACLKCTRL.

Each Low Energy Peripheral that is clocked by EM23GRPACLK may have its own prescaler setting and enable bit. The prescaler settings, if available, can be found in the peripheral's control registers. The enable bit can be found in the module's EN register.

7.3.1.8 EM4GRPACLK - Energy Mode 4 Group A Clock

EM4GRPACLK is the selected clock for the Group A Peripherals operating down to Energy Mode 4. These are typically ultra low energy consumption peripheral modules. There are three selectable sources for EM4GRPACLK: LFRCO, LFXO and ULFRCO. In addition, the EM4GRPACLK can be disabled. The selection is configured using the CLKSEL field in CMU_EM4GRPACLKCTRL.

Note: EM4GRPACLK is in a different power domain than EM23GRPACLK, which makes it available all the way down to EM4.

Each Low Energy Peripheral that is clocked by EM4GRPACLK may have its own prescaler setting and enable bit. The prescaler settings, if available, can be found in the peripheral's control registers. The enable bit can be found in the module's EN register.

7.3.1.9 Peripheral Bus Clock Enable

Peripherals each have an individual bus clock enable bit in the CMU_CLKEN0 or CMU_CLKEN1 registers. Disabling the bus clock to a peripheral can save energy, even when that peripheral is not active.

7.3.1.10 IADCCLK - IADC Clock

IADCCLK is the selected clock for the IADC. The IADCCLK source may be selected from EM01GRPACLK, HFRCOEM23, or FSRCO. In addition, the IADCCLK can be disabled. The selection is configured using the CLKSEL field in CMU_IADCCLKCTRL.

Note: When using a Timer as the synchronous trigger for IADC conversion, EM01GRPACLK must be selected, because Timers run from EM01GRPACLK.

IADC has its own prescaler setting and enable bit. The prescaler settings can be found in the IADC's control registers. The enable bit can be found in the IADC's EN register.

Whichever clock source is selected as the IADC clock via the CLKSEL bitfield in the CMU_IADCCLKCTRL register, this clock will become active automatically when needed. The IADC can automatically start and stop it.

7.3.1.11 VDACnCLK - VDACn Clock

VDACnCLK is the selected clock for VDACn. The VDACnCLK source may be selected from EM01GRPACLK, EM23GRPACLK, HFRCOEM23, or FSRCO. In addition, the VDACnCLK can be disabled. The selection is configured using the CLKSEL field in CMU_VDACnCLKCTRL.

Note: When using a Timer as the synchronous trigger for VDACn conversion, EM01GRPACLK must be selected, because Timers run from EM01GRPACLK.

VDACn has its own prescaler setting and enable bit. The prescaler settings can be found in the VDAC's control registers. The enable bit can be found in the VDAC's EN register.

Whichever clock source is selected as the VDACn clock via the CLKSEL bitfield in the CMU_VDACnCLKCTRL register, this clock will become active automatically when needed. The VDACn can automatically start and stop it.

7.3.1.12 LESENSEHFCLK - LESENSE High Frequency Clock

LESENSEHFCLK is the selected high-frequency clock for the LESENSE peripheral. The selection options are the HFRCOEM23 oscillator or the FSRCO oscillator. The high-frequency clock source is used for faster timing in certain LESENSE configurations. LESENSEHFCLK is selected via the CLKSEL field in CMU_LESENSEHFCLKCTRL. The low-frequency clock for LESENSE is derived from EM23GRPACLK.

7.3.1.13 SYSRTCCLK - SYSRTC Clock

SYSRTCCLK is the selected clock for the SYSRTC peripheral. This clock tree can be clocked from any of the low-frequency oscillators: LFXO, LFRCO, or ULFRCO. SYSRTCCLK is selected via the CLKSEL field in CMU_SYSRTCCLKCTRL.

7.3.1.14 LCDCLK - LCD Clock

LCDCLK is the selected clock for the LCD peripheral. This clock tree can be clocked from any of the low-frequency oscillators: LFXO, LFRCO, or ULFRCO. LCDCLK is selected via the CLKSEL field in CMU_LCDCLKCTRL.

7.3.1.15 PCNT0CLK - PCNT0 Clock

PCNT0CLK is the selected clock for the PCNT peripheral. PCNT can be configured to clock from its S0 input signal, or the EM23GRPACLK, selectable by the CLKSEL field in CMU_PCNT0CLKCTRL. Note that when configured to clock from the S0 input, the clock can further be selected from the direct S0 input pin, or from a PRS channel. Selection of the S0 input is determined by S0PRSEN in PCNT_CFG.

7.3.1.16 EUSART0CLK - EUSART0 Clock

EUSART0CLK is the selected clock for the EUSART0 peripheral, and can choose between EM01GRPACLK, HFRCOEM23, LFXO, and LFRCO. EUSART0CLK is selected via the CLKSEL field in CMU_EUSART0CLKCTRL. When operating EUSART0 as a high-speed UART or SPI main interface (EM0/1 only), EM01GRPACLK or HFRCOEM23 must be selected. To operate as a low-energy UART in EM0, EM1, or EM2, LFXO or LFRCO must be selected. To operate as a SPI secondary interface, EM01GRPACLK or HFRCOEM23 should be selected.

7.3.1.17 TRACECLK - Debug Trace Clock

The CMU scales the clock used for debug trace via the PRESC field in the CMU_TRACECLKCTRL register. The debug trace clock is limited to 40 MHz maximum. Therefore, if the SYSCLK is 40 MHz or less, the default DIV1 setting may be used. When SYSCLK is above 40 MHz, use DIV2 to avoid data pump overflow. The selected debug trace clock will be used to run the Cortex®-M33 trace logic. Note that this register should be configured properly before enabling ETM.

7.3.1.18 WDOGnCLK - Watchdog Timer Clock

The Watchdog Timer (WDOGn) can be configured to use one of four different clock sources: LFRCO, LFXO, ULFRCO, or HCLKDIV1024. Select option HCLKDIV1024 to track Watchdog timeout with CPU clock speed.

7.3.2 Switching Clock Source

The FSRCO oscillator is a fixed frequency (20 MHz), low energy oscillator with extremely short start-up time. Therefore, this oscillator is chosen by hardware as the clock source for SYSCLK when the device starts up (e.g. after reset).

Software can switch between the different clock sources at run-time. For example, when the HFRCODPLL is the clock source, software can switch to HFXO by writing the field CLKSEL in the CMU_SYSCLKCTRL register. See Figure 7.5 CMU Switching From HFRCO to HFXO Before HFXO is Ready on page 146 for a description of the sequence of events for this specific operation.

When switching the SYSCLK to HFXO via the CLKSEL bitfield in CMU_SYSCLKCTRL, HFXO is automatically started. Switching to an oscillator that is not ready yet, the SYSCLK will stop for the duration of the oscillator start-up time. This effectively stalls the Core Modules. It is possible to avoid this by first enabling the target oscillator (e.g. HFXO) and then waiting for that oscillator to become ready before switching the clock source. This way, the system continues to run on the HFRCO until the target oscillator (e.g. HFXO) is ready and provides a reliable clock. This sequence of events is shown in Figure 7.6 CMU Switching From HFRCO to HFXO After HFXO is Ready on page 147.

Generally, all oscillators have a separate flag that is set when the oscillator is ready. This flag can also be configured to generate an interrupt.

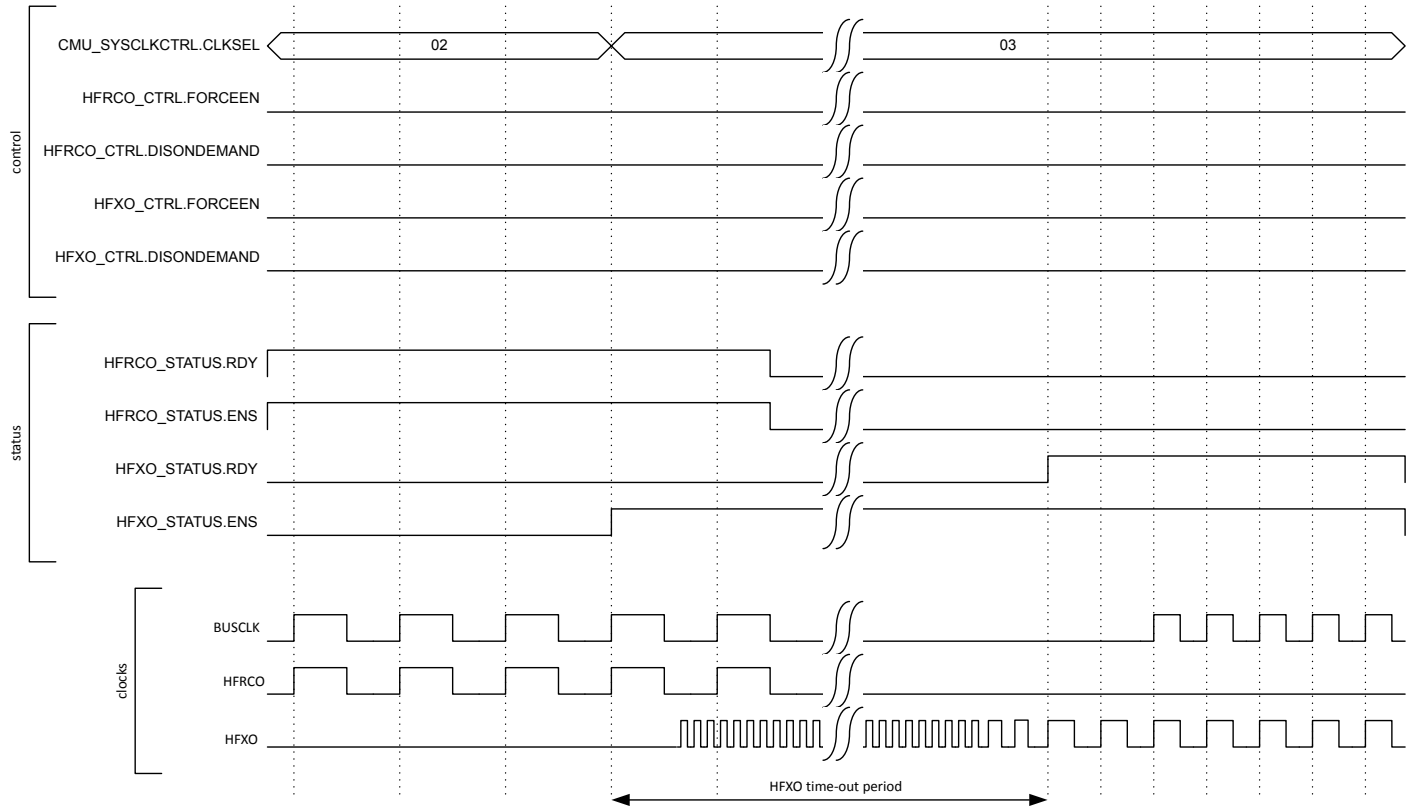


Figure 7.5. CMU Switching From HFRCO to HFXO Before HFXO is Ready

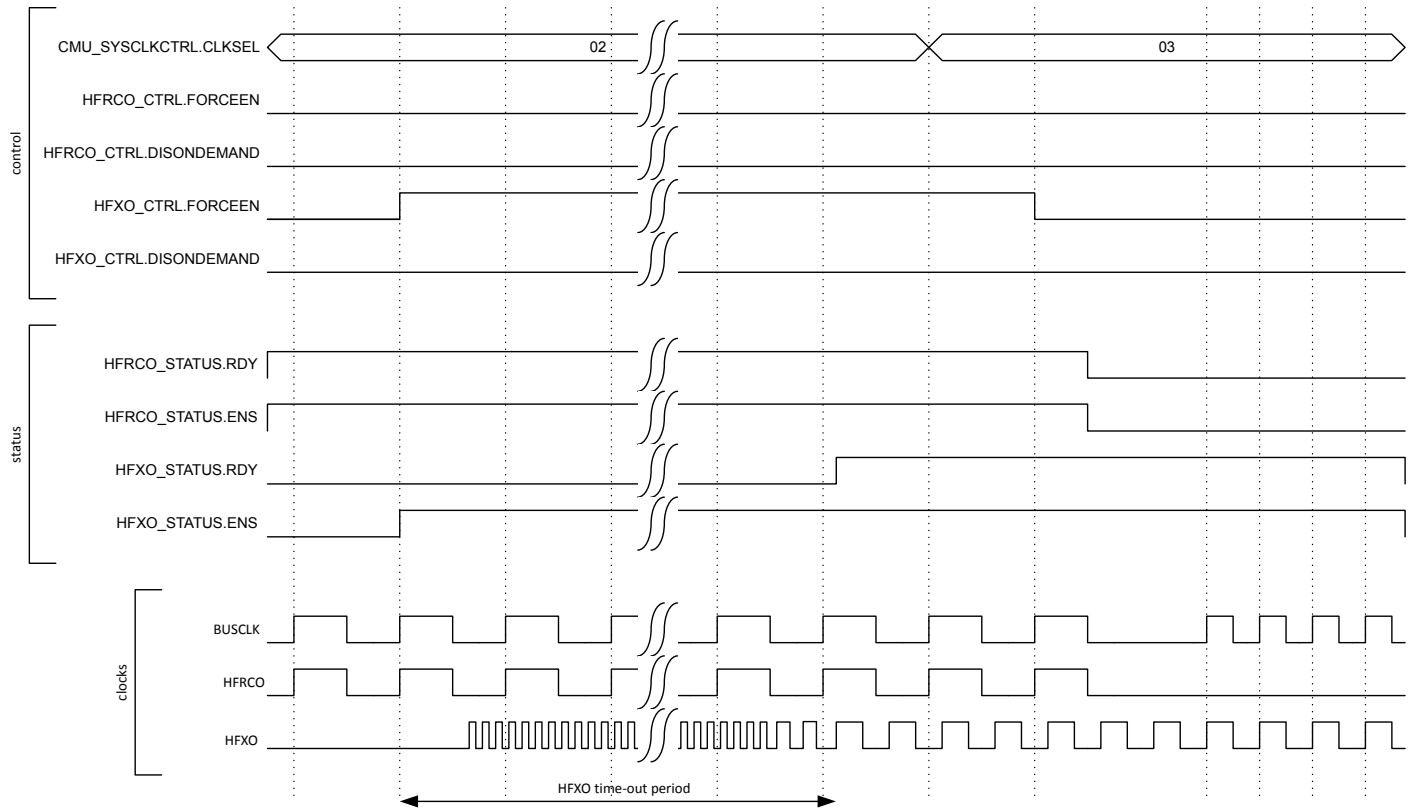


Figure 7.6. CMU Switching From HFRCO to HFXO After HFXO is Ready

Switching clock source for various clock switches is done by setting the CLKSEL bitfields in CMU_*CLKCTRL. To ensure no stalls in the peripherals, the clock source should be ready before switching to it.

Note: To save energy, remember to disable all clock switches and/or module enable bits when not in use.

7.3.3 RC Oscillator Calibration

The CMU has built-in hardware support to efficiently calibrate RC oscillators (LFRCO, HFRCODPLL, HFRCOEM23) at run-time or measure the timing of other periodic signals routed via PRS, see Figure 7.7 Hardware Support for RC Oscillator Calibration on page 148 for an illustration of this circuit.

The concept is to select a reference and compare the RC frequency or PRS timing with the reference frequency. When the calibration circuit is started, one down-counter running on a selectable clock (DOWNSEL in CMU_CALCTRL) and one up-counter running on a selectable clock (UPSEL in CMU_CALCTRL) are started simultaneously. Reference clocks may also be routed through the PRS channels via the CALUP and CALDN consumer inputs. The top value for the down-counter must be written (CALTOP in CMU_CALCTRL) before calibration is started. The down-counter counts for CALTOP + 1 cycles. When the down-counter has reached 0, the up-counter is sampled and the CALRDY interrupt flag in the IF register is set. If CONT in CMU_CALCTRL is cleared, the counters are stopped after finishing the ongoing calibration. If continuous mode is selected by setting CONT in CMU_CALCTRL, the down-counter reloads the top value and continues counting, while the up-counter restarts from 0.

Software can then read out the sampled up-counter value from CMU_CALCNT. The up-counter has counted (the sampled value)+ 1 cycles. The ratio between the reference and the oscillator subject to the calibration can easily be found using (the top value)+1 and (the sampled value)+1. Overflows of the up-counter will not occur. If the up-counter reaches its top value before the down-counter reaches 0, the up-counter stays at its top value. Calibration can be started and stopped by writing CALSTART and CALSTOP bitfields in CMU_CALCMD, respectively. With this hardware support, it is simple to write efficient software calibration algorithms.

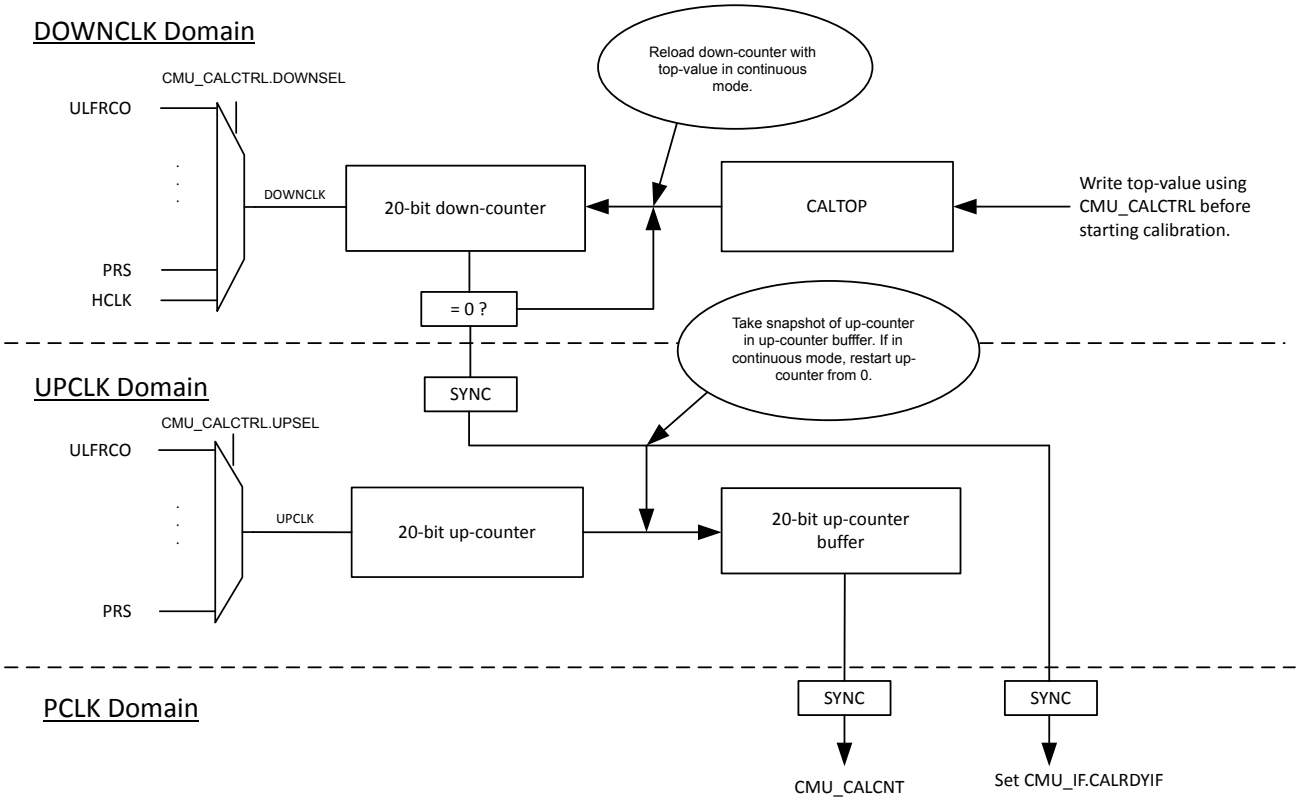


Figure 7.7. Hardware Support for RC Oscillator Calibration

The counter operation for single and continuous mode are shown in Figure 7.8 Single Calibration (CONT=0) on page 149 and Figure 7.9 Continuous Calibration (CONT=1) on page 150 respectively.

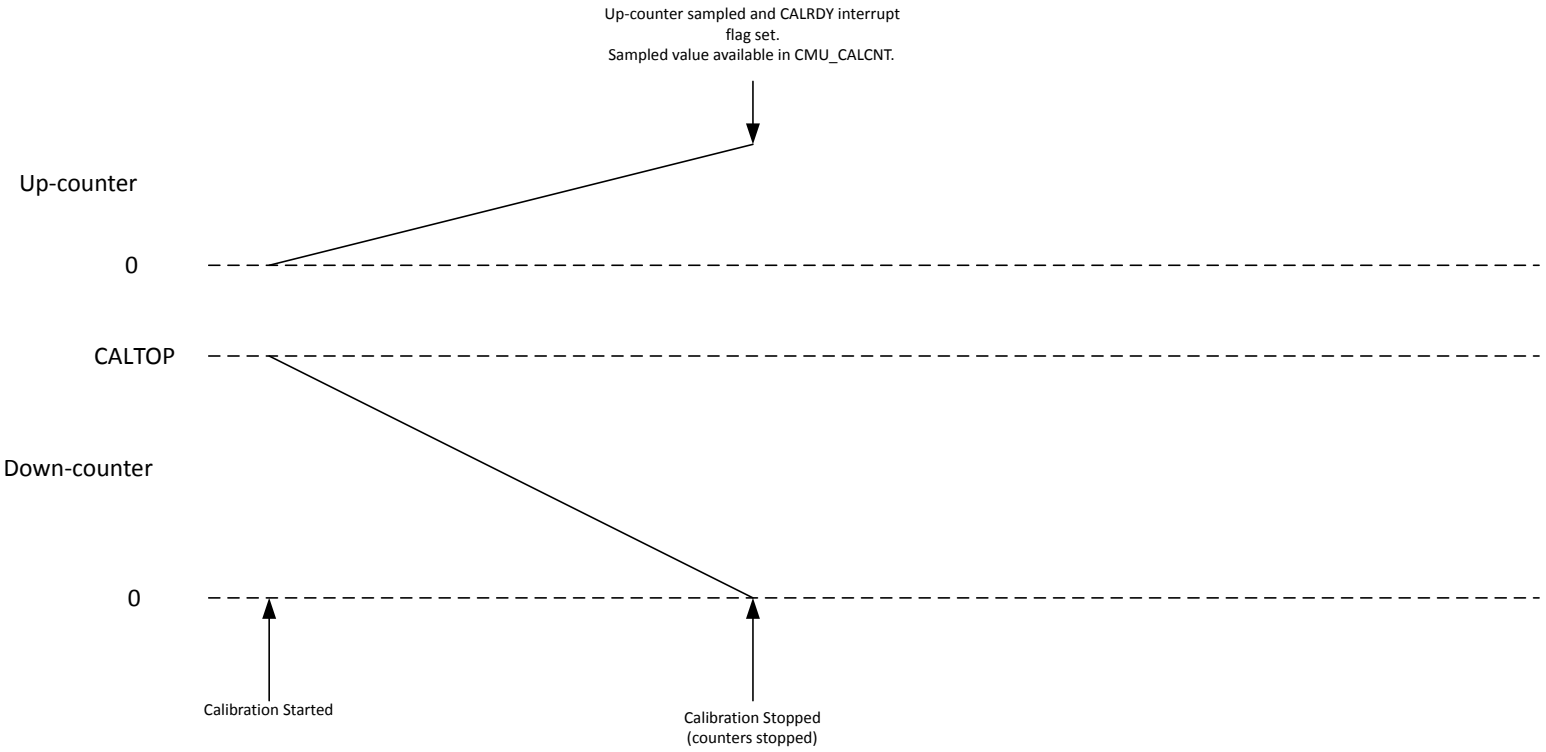


Figure 7.8. Single Calibration (CONT=0)

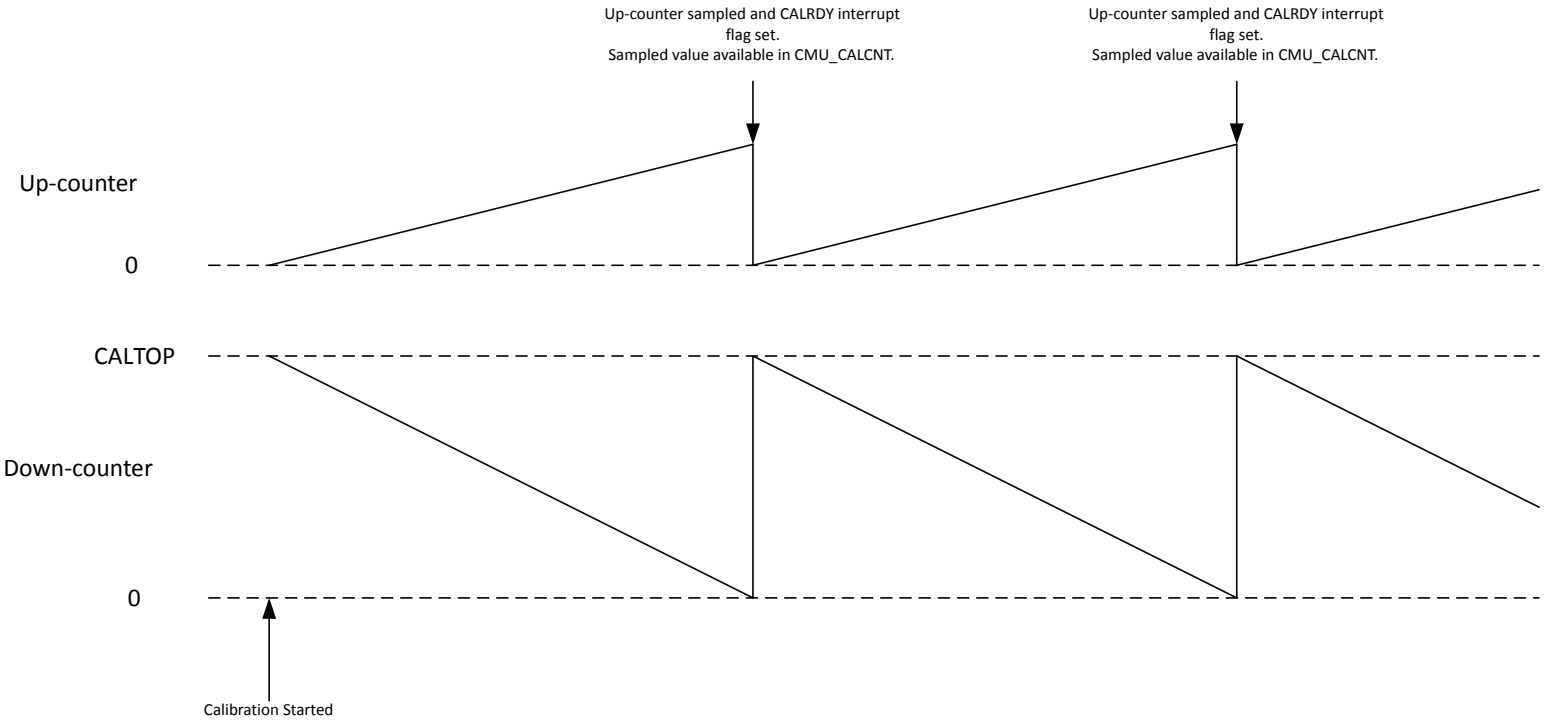


Figure 7.9. Continuous Calibration (CONT=1)

7.3.4 Energy Modes

The availability of oscillators and system clocks depends on the chosen energy mode. By default, the high frequency oscillators and high frequency clocks are available down to EM1 Sleep. From EM2 DeepSleep onwards these oscillators and clocks are normally off, although special cases exist as summarized in Table 7.1 Oscillator and clock availability in Energy Modes on page 151. The CMU figures in 7.3 Functional Description also indicate which oscillators and clocks can be used in what energy modes.

The low frequency oscillators (LFRCO and LFXO) are available in all energy modes except in EM3 Stop when they are off by definition. By default, these oscillators are also off in EM4 Shutoff. The LFXO or LFRCO can be requested in EM4 as needed. The ultra low frequency oscillator (ULFRCO) is on in all energy modes, except for EM4 Shutoff, but it can be requested on in that state as well if needed. The low frequency clocks are in various power domains and therefore their availability not only depends on the chosen clock source, but also on the chosen energy mode as indicated in Table 7.1 Oscillator and clock availability in Energy Modes on page 151.

Table 7.1. Oscillator and clock availability in Energy Modes

	EM0 Active / EM1 Sleep	EM2 DeepSleep	EM3 Stop	EM4 Shutoff
HFRCODPLL	On [1]	Off	Off	Off
HFXO	On [1]	Off	Off	Off
HFRCOEM23	On [1]	On [2]	On [2]	Off
LFRCO, LFXO	On [1]	On [1]	Off	On [3]
ULFRCO	On	On	On	On [3]
SYSCLK, HCLK, PCLK, LSPCLK, EM01GRPACLK, EM01GRPCCLK	On [1]	Off	Off	Off
IADCCLK, LESENSEHFCLK, VDAC0CLK	On [1]	On [2]	On [2]	Off
EM23GRPACLK, WDOGnCLK, SYSRTCCLK, VDAC0_REFRESH_CLK, LCDCLK, PCNT0CLK	On [1]	On [1]	On [4]	Off
EM4GRPACLK	On [1]	On [1]	On [4]	On [3]

1 Under software control.
2 Default off, but kept active if requested by modules.
3 Default off, but kept active if used by BURTC.
4 On only if ULFRCO is used as clock source.

7.3.5 Clock Output

The CMU has up to three CLKOUTn signals that can be routed to the PRS or GPIO. The selections for CLKOUTn are controlled using the CLKOUTSELn bitfields in CMU_EXPORTCLKCTRL (CLKOUTSEL0 controls CLKOUT0, for example).

The following clocks can be selected for CLKOUTn:

- HCLK and EXPORTCLK. The HCLK is the high frequency clock for AHB. The EXPORTCLK is a prescaled version of SYSCLK as controlled by the PRESC bitfield in the CMU_EXPORTCLKCTRL register.
- The qualified clock from any of the on-chip oscillators. A qualified clock will not have any glitches or skewed duty-cycle during start-up. For the LFXO and HFXO, correct configuration of the TIMEOUT bitfield(s) in LFXO_CFG and HFXO_XTALCFG, respectively is required to guarantee a properly qualified clock.

HCLK will only have a 50-50 duty cycle when HCLKPRESC in CMU_SYSCLKCTRL is DIV1. EXPORTCLK will only be 50-50 duty cycle when the selected division factor is even.

The CLKOUTn signals may be routed to GPIO via the DBUS as CMU.CLKOUTn using controls in the GPIO registers. The required output pins must be enabled in the GPIO_CMU_ROUTEEN register and the pin locations can be configured in the GPIO_CMU_CLKOUTnROUTE registers.

The CLKOUTn signals can also be used as PRS producers (see 12.3.3 Producers for more detail on PRS producers). CLKOUTn signals used as PRS producers may be simultaneously routed to GPIO, but this is not required to use CLKOUTn as a PRS producer.

7.3.6 Clock Input from a Pin

It is possible to configure the CMU to input a clock from the CMU_CLKI0. This clock can be selected to drive SYSCLK and DPLL reference using CMU_SYSCLKCTRL.CLKSEL and CMU_DPLLREFCLKCTRL.CLKSEL respectively. The required input pin locations can be configured in the GPIO_CMU_CLKIN0ROUTE register.

7.3.7 Interrupts

The interrupts generated by the CMU module are combined into one interrupt vector. If CMU interrupts are enabled, an interrupt will be made if one or more of the interrupt flags in CMU_IF and their corresponding bits in CMU_IEN are set.

7.3.8 Protection

It is possible to lock the control and command registers to prevent unintended software writes to critical clock settings. This is controlled by the CMU_LOCK register.

The WDOGCLKCTRL registers are separately locked by CMU_WDOGLOCK register. This is to prevent EM3 Stop mode from disabling the watch dog clocks inadvertently.

In addition to software locks, hardware locks are implemented to prevent metastability. CMU_CALCTRL is locked by hardware when calibration is started by CMU_CALCMD.CALSTART. CMU_DPLLREFCLKCTRL is locked by hardware when DPLL is enabled via DPLL_EN.EN. Because these switches are not glitch-less, clock selection must be configured before enabling the operation and cannot be changed during operation.

7.4 CMU Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	CMU_IPVERSION	R	IP Version ID
0x008	CMU_STATUS	RH	Status Register
0x010	CMU_LOCK	W	Configuration Lock Register
0x014	CMU_WDOGLOCK	W	WDOG Configuration Lock Register
0x020	CMU_IF	RWH INTFLAG	Interrupt Flag Register
0x024	CMU_IEN	RW	Interrupt Enable Register
0x050	CMU_CALCMD	W	Calibration Command Register
0x054	CMU_CALCTRL	RW	Calibration Control Register
0x058	CMU_CALCNT	R	Calibration Result Counter Register
0x064	CMU_CLKEN0	RW	Clock Enable Register 0
0x068	CMU_CLKEN1	RW	Clock Enable Register 1
0x070	CMU_SYSCLKCTRL	RW	System Clock Control
0x080	CMU_TRACECLKCTRL	RW	Debug Trace Clock Control
0x090	CMU_EXPORTCLKCTRL	RW	Export Clock Control
0x100	CMU_DPLLREFCLKCTRL	RW	Digital PLL Reference Clock Control
0x120	CMU_EM01GRPACLKCTRL	RW	EM01 Peripheral Group a Clock Control
0x128	CMU_EM01GRPCCLKCTRL	RW	EM01 Peripheral Group C Clock Control
0x140	CMU_EM23GRPACLKCTRL	RW	EM23 Peripheral Group a Clock Control
0x160	CMU_EM4GRPACLKCTRL	RW	EM4 Peripheral Group a Clock Control
0x180	CMU_IADCCLKCTRL	RW	IADC Clock Control
0x200	CMU_WDOG0CLKCTRL	RW	Watchdog0 Clock Control
0x208	CMU_WDOG1CLKCTRL	RW	Watchdog1 Clock Control
0x220	CMU_EUSART0CLKCTRL	RW	EUSART0 Clock Control
0x240	CMU_SYSRTC0CLKCTRL	RW	System RTC0 Clock Control
0x250	CMU_LCDCLKCTRL	RW	LCD Clock Control
0x260	CMU_VDAC0CLKCTRL	RW	VDAC0 Clock Control
0x270	CMU_PCNT0CLKCTRL	RW	Pulse Counter 0 Clock Control
0x290	CMU_LESENSEHFCLKCTRL	RW	LESENSE HF Clock Control
0x1000	CMU_IPVERSION_SET	R	IP Version ID
0x1008	CMU_STATUS_SET	RH	Status Register
0x1010	CMU_LOCK_SET	W	Configuration Lock Register
0x1014	CMU_WDOGLOCK_SET	W	WDOG Configuration Lock Register
0x1020	CMU_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1024	CMU_IEN_SET	RW	Interrupt Enable Register
0x1050	CMU_CALCMD_SET	W	Calibration Command Register

Offset	Name	Type	Description
0x1054	CMU_CALCTRL_SET	RW	Calibration Control Register
0x1058	CMU_CALCNT_SET	R	Calibration Result Counter Register
0x1064	CMU_CLKEN0_SET	RW	Clock Enable Register 0
0x1068	CMU_CLKEN1_SET	RW	Clock Enable Register 1
0x1070	CMU_SYSCLKCTRL_SET	RW	System Clock Control
0x1080	CMU_TRACECLKCTRL_SET	RW	Debug Trace Clock Control
0x1090	CMU_EXPORTCLKCTRL_SET	RW	Export Clock Control
0x1100	CMU_DPLLREFCLKCTRL_SET	RW	Digital PLL Reference Clock Control
0x1120	CMU_EM01GRPACLKCTRL_SET	RW	EM01 Peripheral Group a Clock Control
0x1128	CMU_EM01GRPCCLKCTRL_SET	RW	EM01 Peripheral Group C Clock Control
0x1140	CMU_EM23GRPACLKCTRL_SET	RW	EM23 Peripheral Group a Clock Control
0x1160	CMU_EM4GRPACLKCTRL_SET	RW	EM4 Peripheral Group a Clock Control
0x1180	CMU_IADCCLKCTRL_SET	RW	IADC Clock Control
0x1200	CMU_WDOG0CLKCTRL_SET	RW	Watchdog0 Clock Control
0x1208	CMU_WDOG1CLKCTRL_SET	RW	Watchdog1 Clock Control
0x1220	CMU_EUSART0CLKCTRL_SET	RW	EUSART0 Clock Control
0x1240	CMU_SYSRTC0CLKCTRL_SET	RW	System RTC0 Clock Control
0x1250	CMU_LCDCLKCTRL_SET	RW	LCD Clock Control
0x1260	CMU_VDAC0CLKCTRL_SET	RW	VDAC0 Clock Control
0x1270	CMU_PCNT0CLKCTRL_SET	RW	Pulse Counter 0 Clock Control
0x1290	CMU_LESENSEHFCLKCTRL_SET	RW	LESENSE HF Clock Control
0x2000	CMU_IPVERSION_CLR	R	IP Version ID
0x2008	CMU_STATUS_CLR	RH	Status Register
0x2010	CMU_LOCK_CLR	W	Configuration Lock Register
0x2014	CMU_WDOGLOCK_CLR	W	WDOG Configuration Lock Register
0x2020	CMU_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2024	CMU_IEN_CLR	RW	Interrupt Enable Register
0x2050	CMU_CALCMD_CLR	W	Calibration Command Register
0x2054	CMU_CALCTRL_CLR	RW	Calibration Control Register
0x2058	CMU_CALCNT_CLR	R	Calibration Result Counter Register
0x2064	CMU_CLKEN0_CLR	RW	Clock Enable Register 0
0x2068	CMU_CLKEN1_CLR	RW	Clock Enable Register 1
0x2070	CMU_SYSCLKCTRL_CLR	RW	System Clock Control
0x2080	CMU_TRACECLKCTRL_CLR	RW	Debug Trace Clock Control
0x2090	CMU_EXPORTCLKCTRL_CLR	RW	Export Clock Control

Offset	Name	Type	Description
0x2100	CMU_DPLLREFCLKCTRL_CLR	RW	Digital PLL Reference Clock Control
0x2120	CMU_EM01GRPACLKCTRL_CLR	RW	EM01 Peripheral Group a Clock Control
0x2128	CMU_EM01GRPCCLKCTRL_CLR	RW	EM01 Peripheral Group C Clock Control
0x2140	CMU_EM23GRPACLKCTRL_CLR	RW	EM23 Peripheral Group a Clock Control
0x2160	CMU_EM4GRPACLKCTRL_CLR	RW	EM4 Peripheral Group a Clock Control
0x2180	CMU_IADCCLKCTRL_CLR	RW	IADC Clock Control
0x2200	CMU_WDOG0CLKCTRL_CLR	RW	Watchdog0 Clock Control
0x2208	CMU_WDOG1CLKCTRL_CLR	RW	Watchdog1 Clock Control
0x2220	CMU_EUSART0CLKCTRL_CLR	RW	EUSART0 Clock Control
0x2240	CMU_SYSRTC0CLKCTRL_CLR	RW	System RTC0 Clock Control
0x2250	CMU_LCDCLKCTRL_CLR	RW	LCD Clock Control
0x2260	CMU_VDAC0CLKCTRL_CLR	RW	VDAC0 Clock Control
0x2270	CMU_PCNT0CLKCTRL_CLR	RW	Pulse Counter 0 Clock Control
0x2290	CMU_LESENSEHFCLKCTRL_CLR	RW	LESENSE HF Clock Control
0x3000	CMU_IPVERSION_TGL	R	IP Version ID
0x3008	CMU_STATUS_TGL	RH	Status Register
0x3010	CMU_LOCK_TGL	W	Configuration Lock Register
0x3014	CMU_WDOGLOCK_TGL	W	WDOG Configuration Lock Register
0x3020	CMU_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3024	CMU_IEN_TGL	RW	Interrupt Enable Register
0x3050	CMU_CALCMD_TGL	W	Calibration Command Register
0x3054	CMU_CALCTRL_TGL	RW	Calibration Control Register
0x3058	CMU_CALCNT_TGL	R	Calibration Result Counter Register
0x3064	CMU_CLKEN0_TGL	RW	Clock Enable Register 0
0x3068	CMU_CLKEN1_TGL	RW	Clock Enable Register 1
0x3070	CMU_SYSCLKCTRL_TGL	RW	System Clock Control
0x3080	CMU_TRACECLKCTRL_TGL	RW	Debug Trace Clock Control
0x3090	CMU_EXPORTCLKCTRL_TGL	RW	Export Clock Control
0x3100	CMU_DPLLREFCLKCTRL_TGL	RW	Digital PLL Reference Clock Control
0x3120	CMU_EM01GRPACLKCTRL_TGL	RW	EM01 Peripheral Group a Clock Control
0x3128	CMU_EM01GRPCCLKCTRL_TGL	RW	EM01 Peripheral Group C Clock Control
0x3140	CMU_EM23GRPACLKCTRL_TGL	RW	EM23 Peripheral Group a Clock Control
0x3160	CMU_EM4GRPACLKCTRL_TGL	RW	EM4 Peripheral Group a Clock Control

Offset	Name	Type	Description
0x3180	CMU_IADCCLKCTRL_TGL	RW	IADC Clock Control
0x3200	CMU_WDOG0CLKCTRL_TGL	RW	Watchdog0 Clock Control
0x3208	CMU_WDOG1CLKCTRL_TGL	RW	Watchdog1 Clock Control
0x3220	CMU_EUSART0CLKCTRL_TGL	RW	EUSART0 Clock Control
0x3240	CMU_SYSRTC0CLKCTRL_TGL	RW	System RTC0 Clock Control
0x3250	CMU_LCDCLKCTRL_TGL	RW	LCD Clock Control
0x3260	CMU_VDAC0CLKCTRL_TGL	RW	VDAC0 Clock Control
0x3270	CMU_PCNT0CLKCTRL_TGL	RW	Pulse Counter 0 Clock Control
0x3290	CMU_LESENSEHFCLKCTRL_TGL	RW	LESENSE HF Clock Control

7.5 CMU Register Description

7.5.1 CMU_IPVERSION - IP Version ID

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x6																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x6	R	**IP Version ID**

7.5.2 CMU_STATUS - Status Register

Offset	Bit Position																														
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0																														0x0
Access	R	R																														R
Name	LOCK	WDOGLOCK																														CALRDY

Bit	Name	Reset	Access	Description
31	LOCK	0x0	R	**Configuration Lock Status**
	Indicates the current status of configuration lock			
	Value	Mode		Description
	0	UNLOCKED		Configuration lock is unlocked
	1	LOCKED		Configuration lock is locked
30	WDOGLOCK	0x0	R	**Configuration Lock Status for WDOG**
	Indicates the current status of WDOG configuration lock			
	Value	Mode		Description
	0	UNLOCKED		WDOG configuration lock is unlocked
	1	LOCKED		WDOG configuration lock is locked
29:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	CALRDY	0x0	R	**Calibration Ready**
	Calibration is Ready (0 when calibration is ongoing).			

7.5.3 CMU_LOCK - Configuration Lock Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x93F7															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x93F7	W	**Configuration Lock Key**

Write any other value than the unlock code to lock registers from editing. Write the unlock code to unlock.

Value	Mode	Description
37879	UNLOCK	Write this value to unlock

7.5.4 CMU_WDOGLOCK - WDOG Configuration Lock Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x5257															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x5257	W	**Configuration Lock Key**

Write any other value than the unlock code to lock registers from editing. Write the unlock code to unlock.

Value	Mode	Description
37879	UNLOCK	Write this value to unlock

7.5.5 CMU_IF - Interrupt Flag Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															CALOF	CALRDY

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	CALOF	0x0	RW	**Calibration Overflow Interrupt Flag**
	Set when calibration overflow has occurred (i.e. if a new calibration completes before CMU_CALSTATUS has been read)			
0	CALRDY	0x0	RW	**Calibration Ready Interrupt Flag**
	Set when calibration is completed			

7.5.6 CMU_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															CALOF	CALRDY

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	CALOF	0x0	RW	**Calibration Overflow Interrupt Enable**
	Enable/disable CALOF interrupt			
0	CALRDY	0x0	RW	**Calibration Ready Interrupt Enable**
	Enable/disable CALRDY interrupt			

7.5.7 CMU_CALCMD - Calibration Command Register

Offset	Bit Position																													
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															W(nB)	W(nB)
Name																															CALSTOP	CALSTART

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	CALSTOP	0x0	W(nB)	**Calibration Stop**
	Stops the calibration counters.			
0	CALSTART	0x0	W(nB)	**Calibration Start**
	Starts the calibration, effectively loading the CMU_CALCTRL.CALCNT into the down-counter and start decrementing.			

7.5.8 CMU_CALCTRL - Calibration Control Register

Offset	Bit Position																													
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0				0x0				0x0				0x0																			
Access	RW				RW				RW				RW																			
Name	DOWNSEL				UPSEL				CONT				CALTOP																			

Bit	Name	Reset	Access	Description
31:28	DOWNSEL	0x0	RW	**Calibration Down-counter Select**

Selects clock source for the calibration down-counter. Changing this while calibration is running results in bus fault..

Value	Mode	Description
0	DISABLED	Down-counter is not clocked
1	HCLK	HCLK is clocking down-counter
2	PRS	PRS CMU_CALDN consumer is clocking down-counter
3	HFXO	HFXO is clocking down-counter
4	LFXO	LFXO is clocking down-counter
5	HFRCODPLL	HFRCODPLL is clocking down-counter
6	HFRCOEM23	HFRCOEM23 is clocking down-counter
9	FSRCO	FSRCO is clocking down-counter
10	LFRCO	LFRCO is clocking down-counter
11	ULFRCO	ULFRCO is clocking down-counter

Bit	Name	Reset	Access	Description
27:24	UPSEL	0x0	RW	**Calibration Up-counter Select**

Selects clock source for the calibration up-counter. Changing this while calibration is running results in bus fault.

Value	Mode	Description
0	DISABLED	Up-counter is not clocked
1	PRS	PRS CMU_CALUP consumer is clocking up-counter
2	HFXO	HFXO is clocking up-counter
3	LFXO	LFXO is clocking up-counter
4	HFRCODPLL	HFRCODPLL is clocking up-counter
5	HFRCOEM23	HFRCOEM23 is clocking up-counter
8	FSRCO	FSRCO is clocking up-counter
9	LFRCO	LFRCO is clocking up-counter
10	ULFRCO	ULFRCO is clocking up-counter

Bit	Name	Reset	Access	Description
23	CONT	0x0	RW	**Continuous Calibration**

Bit	Name	Reset	Access	Description
	Set this bit to enable continuous calibration. Changing this while calibration is running results in bus fault.			
22:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:0	CALTOP	0x0	RW	**Calibration Counter Top Value**
	Write top value before calibration. Changing this while calibration is running results in bus fault.			

7.5.9 CMU_CALCNT - Calibration Result Counter Register

Offset	Bit Position																															
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																			
Access													R																			
Name													CALCNT																			

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:0	CALCNT	0x0	R	**Calibration Result Counter Value**
	Read calibration result when Calibration Ready flag has been set.			

7.5.10 CMU_CLKEN0 - Clock Enable Register 0

Offset	Bit Position																															
0x064	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0		0x0	0x0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW	RW
Name	DCDC	SYSRTC0	BURTC	BURAM	PRS	GPIO	LESENSE	ULFRCO	LFXO	LFRCO	FSRCO	HFXO0	HFRCOEM23	HFRCO0	DPLL0	SYSCFG	I2C1	I2C0	WDOG0	LETIMER0	AMUXCP0	IADC0	USART0	TIMER4	TIMER3	TIMER2	TIMER1	TIMER0	GPCRC		LDMAXBAR	LDMA

Bit	Name	Reset	Access	Description
31	DCDC	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
30	SYSRTC0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
29	BURTC	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
28	BURAM	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
27	PRS	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
26	GPIO	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
25	LESENSE	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
24	ULFRCO	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
23	LFXO	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
22	LFRCO	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
21	FSRCO	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
20	HFXO0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
19	HFRCOEM23	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
18	HFRCO0	0x0	RW	**Enable Bus Clock**

Bit	Name	Reset	Access	Description
	Enables module PCLK/HCLK			
17	DPLL0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
16	SYSCFG	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
15	I2C1	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
14	I2C0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
13	WDOG0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
12	LETIMER0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
11	AMUXCP0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
10	IADC0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
9	USART0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
8	TIMER4	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
7	TIMER3	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
6	TIMER2	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
5	TIMER1	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
4	TIMER0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
3	GPCRC	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	LDMAXBAR	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
0	LDMA	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			

7.5.11 CMU_CLKEN1 - Clock Enable Register 1

Offset	Bit Position																															
0x068	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset			0x0		0x0			0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0		0x0		0x0								
Access			RW		RW			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW		RW								
Name			MVP		DMEM			EUSART2	EUSART1	EUSART0	PCNT0	VDAC0	ACMP1	ACMP0	WDOG1	MSC	ICACHE0	SMU	KEYSCAN	LCD		SEMAILBOXHOST		HOSTMAILBOX								

Bit	Name	Reset	Access	Description
31:30	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
29	MVP	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27	DMEM	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
26:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24	EUSART2	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
23	EUSART1	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
22	EUSART0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
21	PCNT0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
20	VDAC0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
19	ACMP1	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
18	ACMP0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
17	WDOG1	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
16	MSC	0x0	RW	**Enable Bus Clock**

Bit	Name	Reset	Access	Description
	Enables module PCLK/HCLK			
15	ICACHE0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
14	SMU	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
13	KEYSCAN	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
12	LCD	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	SEMAILBOXHOST	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	HOSTMAILBOX	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
7:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

7.5.12 CMU_SYSCLKCTRL - System Clock Control

Offset	Bit Position																															
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0					0x0									0x1	
Access																	RW					RW									RW	
Name																	HCLKPRESC					PCLKPRESC									CLKSEL	

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:12	HCLKPRESC	0x0	RW	**HCLK Prescaler**

Specifies the clock divider for HCLK

Value	Mode	Description
0	DIV1	HCLK is SYSCLK divided by 1
1	DIV2	HCLK is SYSCLK divided by 2
3	DIV4	HCLK is SYSCLK divided by 4
7	DIV8	HCLK is SYSCLK divided by 8
15	DIV16	HCLK is SYSCLK divided by 16

11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	PCLKPRESC	0x0	RW	**PCLK Prescaler**

Specifies the clock divider for PCLK

Value	Mode	Description
0	DIV1	PCLK is HCLK divided by 1
1	DIV2	PCLK is HCLK divided by 2

9:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for SYSCLK.

Value	Mode	Description
1	FSRCO	FSRCO is clocking SYSCLK
2	HFRCODPLL	HFRCODPLL is clocking SYSCLK
3	HFXO	HFXO is clocking SYSCLK
4	CLKIN0	CLKIN0 is clocking SYSCLK

7.5.13 CMU_TRACECLKCTRL - Debug Trace Clock Control

Offset	Bit Position																													
0x080	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0				0x1	
Access																											RW				RW	
Name																											PRESC				CLKSEL	

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	PRESC	0x0	RW	**TRACECLK Prescaler**

Clock prescaler for the TRACECLKIN of TPIU. Changing this while the TRCENA bit is set in the ARM M33 Debug Exception and Monitor Control Register (DEMCR) will result in a bus fault.

Value	Mode	Description
0	DIV1	TRACECLK is selected clock source divided by 1
1	DIV2	TRACECLK is selected clock source divided by 2
2	DIV3	TRACECLK is selected clock source divided by 3
3	DIV4	TRACECLK is selected clock source divided by 4

3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x1	RW	**Clock Select**

Selects clock source for the TRACECLKIN of TPIU. Changing this while the TRCENA bit is set in the ARM M33 Debug Exception and Monitor Control Register (DEMCR) will result in a bus fault.

Value	Mode	Description
0	DISABLE	TRACE clock is disable
1	SYSCLK	SYSCLK is driving TRACE
2	HFRCOEM23	HFRCOEM23 is driving TRACE
3	HFRCODPLLRT	HFRCODPLLRT is driving TRACE

7.5.14 CMU_EXPORTCLKCTRL - Export Clock Control

Offset	Bit Position																													
0x090	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x0									0x0								0x0								0x0			
Access				RW									RW								RW								RW			
Name				PRESC									CLKOUTSEL2								CLKOUTSEL1								CLKOUTSEL0			

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28:24	PRESC	0x0	RW	**EXPORTCLK Prescaler**
	Specifies the clock divider for EXPORTCLK (relative to SYSCLK).			
23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	CLKOUTSEL2	0x0	RW	**Clock Output Select 2**
	Controls the clock output 2 multiplexer.			

Value	Mode	Description
0	DISABLED	CLKOUT2 is not clocked
1	HCLK	HCLK is clocking CLKOUT2
2	HFEXPCLK	EXPORTCLK is clocking CLKOUT2
3	ULFRCO	ULFRCO is clocking CLKOUT2
4	LFRCO	LFRCO is clocking CLKOUT2
5	LFXO	LFXO is clocking CLKOUT2
6	HFRCODPLL	HFRCODPLL is clocking CLKOUT2
7	HFXO	HFXO is clocking CLKOUT2
8	FSRCO	FSRCO is clocking CLKOUT2
9	HFRCOEM23	HFRCOEM23 is clocking CLKOUT2

15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:8	CLKOUTSEL1	0x0	RW	**Clock Output Select 1**
	Controls the clock output 1 multiplexer.			

Value	Mode	Description
0	DISABLED	CLKOUT1 is not clocked
1	HCLK	HCLK is clocking CLKOUT1
2	HFEXPCLK	EXPORTCLK is clocking CLKOUT1

Bit	Name	Reset	Access	Description
	3	ULFRCO		ULFRCO is clocking CLKOUT1
	4	LFRCO		LFRCO is clocking CLKOUT1
	5	LFXO		LFXO is clocking CLKOUT1
	6	HFRCODPLL		HFRCODPLL is clocking CLKOUT1
	7	HFXO		HFXO is clocking CLKOUT1
	8	FSRCO		FSRCO is clocking CLKOUT1
	9	HFRCOEM23		HFRCOEM23 is clocking CLKOUT1
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	CLKOUTSEL0	0x0	RW	**Clock Output Select 0**
	Controls the clock output 0 multiplexer.			

Value	Mode	Description
0	DISABLED	CLKOUT0 is not clocked
1	HCLK	HCLK is clocking CLKOUT0
2	HFEXPCLK	EXPORTCLK is clocking CLKOUT0
3	ULFRCO	ULFRCO is clocking CLKOUT0
4	LFRCO	LFRCO is clocking CLKOUT0
5	LFXO	LFXO is clocking CLKOUT0
6	HFRCODPLL	HFRCODPLL is clocking CLKOUT0
7	HFXO	HFXO is clocking CLKOUT0
8	FSRCO	FSRCO is clocking CLKOUT0
9	HFRCOEM23	HFRCOEM23 is clocking CLKOUT0

7.5.15 CMU_DPLLREFCLKCTRL - Digital PLL Reference Clock Control

Offset	Bit Position																													
0x100	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	
Access																															RW	
Name																															CLKSEL	

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x0	RW	**Clock Select**

Selects the clock source for DPLL reference. Changing this while DPLL is enabled results in bus fault.

Value	Mode	Description
0	DISABLED	DPLLREFCLK is not clocked
1	HFXO	HFXO is clocking DPLLREFCLK
2	LFXO	LFXO is clocking DPLLREFCLK
3	CLKIN0	CLKIN0 is clocking DPLLREFCLK

7.5.16 CMU_EM01GRPACLKCTRL - EM01 Peripheral Group a Clock Control

Offset	Bit Position																													
0x120	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x1		
Access																														RW		
Name																														CLKSEL		

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for EM01 Group A Clock.

Value	Mode	Description
1	HFRCODPLL	HFRCODPLL is clocking EM01GRPACLK
2	HFXO	HFXO is clocking EM01GRPACLK
3	FSRCO	FSRCO is clocking EM01GRPACLK
4	HFRCOEM23	HFRCOEM23 is clocking EM01GRPACLK
5	HFRCODPLLRT	HFRCODPLL (retimed) is clocking EM01GRPACLK. Check with datasheet for frequency limitation when using retiming with voltage scaling.
6	HFXORT	HFXO (retimed) is clocking EM01GRPACLK. Check with datasheet for frequency limitation when using retiming with voltage scaling.

7.5.17 CMU_EM01GRPCCLKCTRL - EM01 Peripheral Group C Clock Control

Offset	Bit Position																														
0x128	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x1		
Access																														RW		
Name																														CLKSEL		

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for EM01 Group C Clock.

Value	Mode	Description
1	HFRCODPLL	HFRCODPLL is clocking EM01GRPCCLK
2	HFXO	HFXO is clocking EM01GRPCCLK
3	FSRCO	FSRCO is clocking EM01GRPCCLK
4	HFRCOEM23	HFRCOEM23 is clocking EM01GRPCCLK
5	HFRCODPLLRT	HFRCODPLL (retimed) is clocking EM01GRPCCLK. Check with datasheet for frequency limitation when using retiming with voltage scaling.
6	HFXORT	HFXO (retimed) is clocking EM01GRPCCLK. Check with datasheet for frequency limitation when using retiming with voltage scaling.

7.5.18 CMU_EM23GRPACLKCTRL - EM23 Peripheral Group a Clock Control

Offset	Bit Position																													
0x140	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	
Access																															RW	
Name																															CLKSEL	

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for EM23 Group A Clock.

Value	Mode	Description
1	LFRCO	LFRCO is clocking EM23GRPACLK
2	LFXO	LFXO is clocking EM23GRPACLK
3	ULFRCO	ULFRCO is clocking EM23GRPACLK

7.5.19 CMU_EM4GRPACLKCTRL - EM4 Peripheral Group a Clock Control

Offset	Bit Position																													
0x160	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	
Access																															RW	
Name																															CLKSEL	

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for EM4 Group A Clock.

Value	Mode	Description
1	LFRCO	LFRCO is clocking EM4GRPACLK
2	LFXO	LFXO is clocking EM4GRPACLK
3	ULFRCO	ULFRCO is clocking EM4GRPACLK

7.5.20 CMU_IADCCLKCTRL - IADC Clock Control

Offset	Bit Position																													
0x180	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	
Access																															RW	
Name																															CLKSEL	

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for for IADC. EM01GRPACLK should never be selected as clock source for IADC when disabling the EM01GRACLK (e.g. because of EM23 entry).

Value	Mode	Description
1	EM01GRPACLK	EM01GRPACLK is clocking IADCCLK
2	FSRCO	FSRCO is clocking IADCCLK
3	HFRCOEM23	HFRCOEM23 is clocking IADCCLK

7.5.21 CMU_WDOG0CLKCTRL - Watchdog0 Clock Control

Offset	Bit Position																													
0x200	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	
Access																															RW	
Name																															CLKSEL	

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for WDOG0.

Value	Mode	Description
1	LFRCO	LFRCO is clocking WDOG0CLK
2	LFXO	LFXO is clocking WDOG0CLK
3	ULFRCO	ULFRCO is clocking WDOG0CLK
4	HCLKDIV1024	HCLKDIV1024 is clocking WDOG0CLK

7.5.22 CMU_WDOG1CLKCTRL - Watchdog1 Clock Control

Offset	Bit Position																														
0x208	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x1		
Access																														RW		
Name																														CLKSEL		

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for WDOG1.

Value	Mode	Description
1	LFRCO	LFRCO is clocking WDOG0CLK
2	LFXO	LFXO is clocking WDOG0CLK
3	ULFRCO	ULFRCO is clocking WDOG0CLK
4	HCLKDIV1024	HCLKDIV1024 is clocking WDOG0CLK

7.5.23 CMU_EUSART0CLKCTRL - EUSART0 Clock Control

Offset	Bit Position																														
0x220	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x1		
Access																														RW		
Name																														CLKSEL		

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	CLKSEL	0x1	RW	**Clock Select**

This bit controls which clock is used for EUSART0. EM01GRPCCLK should never be selected as clock source when disabling the EM01GRCCLK (e.g. because of EM23 entry).

Value	Mode	Description
0	DISABLED	EUSART0 is not clocked
1	EM01GRPCCLK	EM01GRPCCLK is clocking EUSART0
2	HFRCOEM23	HFRCOEM23 is clocking EUSART0
3	LFRCO	LFRCO is clocking EUSART0
4	LFXO	LFXO is clocking EUSART0

7.5.24 CMU_SYSRTC0CLKCTRL - System RTC0 Clock Control

Offset	Bit Position																													
0x240	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	
Access																															RW	
Name																															CLKSEL	

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for SYSRTC0.

Value	Mode	Description
1	LFRCO	LFRCO is clocking SYSRTC0CLK
2	LFXO	LFXO is clocking SYSRTC0CLK
3	ULFRCO	ULFRCO is clocking SYSRTC0CLK

7.5.25 CMU_LCDCLKCTRL - LCD Clock Control

Offset	Bit Position																													
0x250	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	
Access																															RW	
Name																															CLKSEL	

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for LCD

Value	Mode	Description
1	LFRCO	LFRCO is clocking LCDCLK
2	LFXO	LFXO is clocking LCDCLK
3	ULFRCO	ULFRCO is clocking LCDCLK

7.5.26 CMU_VDAC0CLKCTRL - VDAC0 Clock Control

Offset	Bit Position																													
0x260	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x1		
Access																														RW		
Name																														CLKSEL		

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	CLKSEL	0x1	RW	**Clock Select**

This bit controls which clock is used for VDAC. EM01GRPACLK should never be selected as clock source when disabling the EM01GRACLK (e.g. because of EM23 entry).

Value	Mode	Description
0	DISABLED	VDAC is not clocked
1	EM01GRPACLK	EM01GRPACLK is clocking VDAC
2	EM23GRPACLK	EM23GRPACLK is clocking VDAC
3	FSRCO	FSRCO is clocking VDAC
4	HFRCOEM23	HFRCOEM23 is clocking VDAC

7.5.27 CMU_PCNT0CLKCTRL - Pulse Counter 0 Clock Control

Offset	Bit Position																													
0x270	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	
Access																															RW	
Name																															CLKSEL	

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x1	RW	**Clock Select**

This bit controls which clock is used for PCNT0.

Value	Mode	Description
0	DISABLED	PCNT0 is not clocked
1	EM23GRPACLK	EM23GRPACLK is clocking PCNT0
2	PCNTS0	External pin PCNT_S0 is clocking PCNT0

7.5.28 CMU_LESENSEHFCLKCTRL - LESENSE HF Clock Control

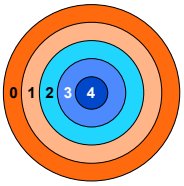
Offset	Bit Position																													
0x290	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	
Access																															RW	
Name																															CLKSEL	

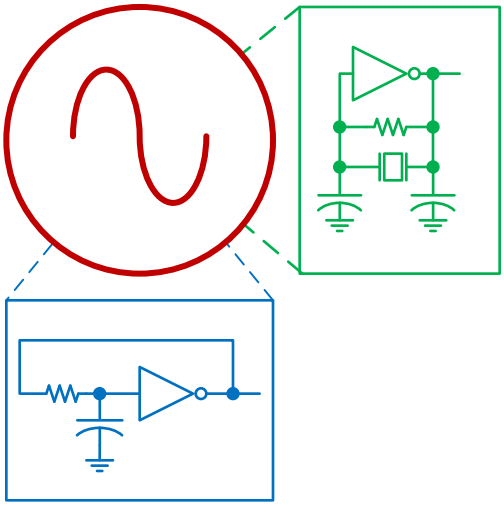
Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for LESENSE high frequency logic.

Value	Mode	Description
1	FSRCO	FSRCO is clocking LESENSEHFCLK
2	HFRCOEM23	HFRCOEM23 is clocking LESENSEHFCLK

8. Oscillators





Quick Facts

What?

The EFM32PG28 has a wide range of high frequency and low frequency oscillators.

Why?

The High Frequency oscillators support EM0/1 operation. The Low-frequency oscillators provide a low frequency clock for the low energy peripherals in EM/2/3/4.

How?

The HFXO supports high freqeuency crystal oscillators. The LFXO supports 32.768 kHz crystal oscillators. The RC oscillators are internal oscillators that require no external components.

8.1 Introduction

The EFM32PG28 has several oscillators. This chapter contains a detailed function description and register descriptions for each oscillator. The CMU chapter includes information on how to select clock sources. Each oscillator may require some initial configuration or calibration before being enabled. The CMU supports clock on demand and can enable and disable oscillators. Therefore, it is important to properly configure each oscillator before selecting it as a clock source in the CMU.

8.2 HFXO - High Frequency Crystal Oscillator

8.2.1 Introduction

The High Frequency Crystal Oscillator (HFXO) uses an external high frequency crystal and provides a sequencer for starting up the crystal safely and reliably, while minimize energy consumption. An external sine wave clock source can also be used in the absence of a crystal.

8.2.2 Features

- Optimized for 39 MHz crystals
- Multiple programming options of start-up parameters to enable optimization of different crystals, supporting a wide range of ESR and ESL
- Support for external sine wave input
- Programmable two-phase start-up to minimize energy consumption
- Built-in current optimization (Automatic oscillation amplitude control)
- Independent on-chip frequency tuning capacitors
- Hardware request for on-demand enable/disable
- Register lock

8.2.3 Functional Description

8.2.3.1 Enabling and Disabling

While the HFXO supports on-demand clocking, it is generally recommended to manually manage the HFXO, at least initially, because it requires software configuration and has a long start-up time. Software can set the FORCEEN to start HFXO and keep it enabled even if it is not selected as a clock source.

However, once started and before EM2 entry, switching the HFXO to on-demand mode may be desirable. This allows the MCU to enter EM2 and then restart the HFXO automatically upon EM2 exit. (During EM1P the HFXO can be conditionally started, depending on the wake-up trigger source.)

The HFXO can be enabled and disabled via both hardware and software mechanisms. Enabling via software is done by setting the FORCEEN bit in the HFXO_CTRL register. Disabling via software is done by setting the DISONDEMAND bit and clearing FORCEEN bit in the HFXO_CTRL register. The hardware controlled on-demand mode is enabled by clearing the FORCEEN and DISONDEMAND bits in the HFXO_CTRL register. Once configured the on-demand mode hardware can autonomously start and stop the HFXO based on various peripheral clock requests in combination with clock switch selections in the CMU. The HFXO is automatically stopped when entering EM2, EM3, or EM4. Hardware can also stop the HFXO via hardware in response to change in peripheral requests and clock switch selections in the CMU.

8.2.3.2 Start-up Time

The start-up time differs for different crystals and the HFXO has a configurable time-out to accommodate each crystal type. Software configures the timeout by setting the various TIMEOUT bit fields of the HFXO_XTALCFG register. The time-out delays the assertion of the RDY signal for HFXO. The programmed timeout should allow enough time for the oscillator to stabilize. The time-out can be optimized for the chosen crystal used in the application.

The start-up behavior of the HFXO also depends on how and how long the HFXO is disabled.

8.2.3.3 Configuration

The High Frequency Crystal Oscillator needs to be configured to ensure safe start-up for the given crystal. Refer to the Device Data sheet and application notes for guidelines in selecting correct components and crystals as well as for configuration trade-offs.

The HFXO crystal is connected to the HFXTAL_I/HFXTAL_O pins as shown in Figure 8.1 HFXO Pin Connection on page 183.

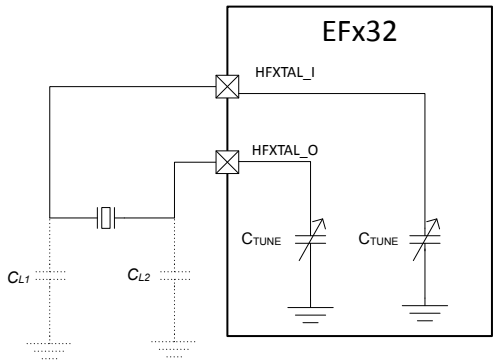


Figure 8.1. HFXO Pin Connection

Upon enabling the HFXO, a hardware state machine sequentially applies the configurable start-up state, intermediate start-up state, and steady state control settings from the HFXO_XTALCFG and HFXO_XTALCTRL registers. After reaching the steady operation state of the HFXO, it is recommended to further optimize current consumption using the Core Bias Optimization Algorithm to trade off noise and current consumption.

Refer to AN0016.2 for more information on settings for different crystals. Write the configuration values, which depends on the crystal's CL, RESR and oscillation frequency, into HFXO_XTALCFG and HFXO_XTALCTRL registers.

- COREBIASSTARTUP (HFXO_XTALCFG) - current setting applied at start-up time
- COREBIASSTARTUPI (HFXO_XTALCFG) - current setting applied at intermediate start-up time
- COREBIASANA (HFXO_XTALCTRL) - current setting applied at steady state
- CTUNEXISTARTUP (HFXO_XTALCFG) - tuning cap setting for XI applied at start-up time
- CTUNEXIANA (HFXO_XTALCTRL) - tuning cap setting for XI applied at steady state
- CTUNEXOSTARTUP (HFXO_XTALCFG) - tuning cap setting for XO applied at start-up time
- CTUNEXOANA (HFXO_XTALCTRL) - tuning cap setting for XO applied at steady state
- CTUNEFIXANA (HFXO_XTALCTRL) - fixed tuning cap setting applied throughout
- TIMEOUTSTEADY (HFXO_XTALCFG) - duration for the steady state settling time
- TIMEOUTCBLSB (HFXO_XTALCFG) - duration for the optimization settling after each step

All HFXO configuration needs to be performed prior to enabling the HFXO, whether via software by setting FORCEEN bit field, or allowing hardware request by clearing DISONDEMAND bit field in the HFXO_CTRL register.

By default, the HFXO is started in crystal mode, but it is possible to connect an active external sine or clipped sine wave clock source to the HFXTAL_I pin of the HFXO. By configuring the MODE field in HFXO_CFG to EXTCLK, the HFXO can be bypassed and the source clock can be provided through the HFXTAL_I pin.

8.2.3.4 Status Flags

The ENS flag in the HFXO_STATUS indicates if the HFXO has been successfully enabled. Once the HFXO oscillation amplitude has exceeded the start-up threshold and intermediate start-up threshold, the steady state settling timeout begins. When the steady state timeout has expired, the HFXO is ready for use as indicated by the RDY flag in the HFXO_STATUS. Once Core Bias Optimization is enabled, the COREBIASOPTRDY flag in the CMU_STATUS register indicates when the optimization is ready. It is advised to wait for this flag before using the HFXO, because optimization can cause minor disturbance to the oscillator frequency.

8.2.3.5 On-Demand Clocking

Software can request to enable the HFXO by setting the HFXO_CTRL.FORCEEN bit field. The HFXO can also optionally be configured via the HFXO_CTRL.DISONDEMAND to shut down when no hardware request is present. This is known as on-demand clocking and allows the oscillator to be controlled without any software intervention. Any hardware request for HFXO is indicated in the HWREQ bit field of the HFXO_STATUS register. This request enables the HFXO, provided that DISONDEMAND bit field is cleared in HFXO_CTRL register. The HFXO is only disabled by hardware upon EM2, EM3 or EM4 entry.

The HFXO analog circuitry can optionally continue operating with the clock output shut off when the HFXO is disabled. This is configured by setting the KEEPWARM bit in HFXO_STATUS.

8.2.3.6 Interrupts

RDYIF and COREBIASOPTRDYIF are interrupt flags as well as status flags. This allows software flexibility to implement interrupt service routine or polling loop for these events. When steady state timeout has exceeded, sticky RDYIF is set until it is cleared by software. If optimization is enabled, sticky COREBIASOPTRDYIF is set when optimization is completed successfully. However, if optimization fails to complete, sticky COREBIASOPTERRIF is set, and the HFXO control state machine stays in the error state until the oscillator is disabled. Similarly, if HFXO fails to start-up, meaning it has not reached the steady state, sticky DNSERRIF is set. The HFXO control state machine stays in the error state until the oscillator is disabled.

8.2.3.7 Protection

It is possible to lock the control registers, configuration registers, and command register to prevent unintended software writes to critical clock settings. This is controlled by the HFXO_LOCK register. A LOCK bit is available in HFXO_STATUS register. Furthermore, these registers are locked automatically by hardware to prevent clock domain crossing malfunction. To gain access to these registers while oscillator is in steady operation state, set FORCEEN to 1, then set DISONDEMAND to 1 in the HFXO_CTRL register. A FSMLOCK bit in HFXO_STATUS register indicates when it is safe for software to update control registers and configuration registers. When software is finished with updates, put the oscillator back to on-demand mode by clearing DISONDEMAND to 0, followed by clearing FORCEEN to 0 in the HFXO_CTRL register. While DISONDEMAND is 0, FSMLOCK is always set, even if hardware is not requesting. This is to prevent a race condition between software access and hardware lock.

8.2.3.8 Tuning

While the oscillator is running in steady operation state, it may be desirable to change control settings. One example is frequency tuning by modifying the tuning capacitance via CTUNEXIANA and CTUNEXOANA fields in the HFXO_XTALCTRL register. When tuning, care should be taken to make small changes to the CTUNE registers. Ideally, change the CTUNE registers by one LSB at a time and alternate between the XI and XO registers. Sufficient wait time for settling, on the order of TIMEOUTSTEADY, should pass before new frequency measurement is taken.

8.2.4 HFXO Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	HFXO_IPVERSION	R	IP Version ID
0x010	HFXO_XTALCFG	RW SYNC	Crystal Configuration Register
0x018	HFXO_XTALCTRL	RWH SYNC	Crystal Control Register
0x01C	HFXO_XTALCTRL1	RW SYNC	BUFOUT Crystal Control Register
0x020	HFXO_CFG	RW SYNC	Configuration Register
0x028	HFXO_CTRL	RWH SYNC	Control Register
0x040	HFXO_BUFOUTTRIM	RW SYNC	BUFOUT Trim Configuration Register
0x044	HFXO_BUFOUTCTRL	RW SYNC	BUFOUT Control Register
0x050	HFXO_CMD	W SYNC	Command Register
0x058	HFXO_STATUS	RH	Status Register
0x070	HFXO_IF	RWH INTFLAG	Interrupt Flag Register
0x074	HFXO_IEN	RW	Interrupt Enable Register
0x080	HFXO_LOCK	W	Configuration Lock Register
0x1000	HFXO_IPVERSION_SET	R	IP Version ID
0x1010	HFXO_XTALCFG_SET	RW SYNC	Crystal Configuration Register
0x1018	HFXO_XTALCTRL_SET	RWH SYNC	Crystal Control Register
0x101C	HFXO_XTALCTRL1_SET	RW SYNC	BUFOUT Crystal Control Register
0x1020	HFXO_CFG_SET	RW SYNC	Configuration Register
0x1028	HFXO_CTRL_SET	RWH SYNC	Control Register
0x1040	HFXO_BUFOUTTRIM_SET	RW SYNC	BUFOUT Trim Configuration Register
0x1044	HFXO_BUFOUTCTRL_SET	RW SYNC	BUFOUT Control Register
0x1050	HFXO_CMD_SET	W SYNC	Command Register
0x1058	HFXO_STATUS_SET	RH	Status Register
0x1070	HFXO_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1074	HFXO_IEN_SET	RW	Interrupt Enable Register
0x1080	HFXO_LOCK_SET	W	Configuration Lock Register
0x2000	HFXO_IPVERSION_CLR	R	IP Version ID
0x2010	HFXO_XTALCFG_CLR	RW SYNC	Crystal Configuration Register
0x2018	HFXO_XTALCTRL_CLR	RWH SYNC	Crystal Control Register
0x201C	HFXO_XTALCTRL1_CLR	RW SYNC	BUFOUT Crystal Control Register
0x2020	HFXO_CFG_CLR	RW SYNC	Configuration Register
0x2028	HFXO_CTRL_CLR	RWH SYNC	Control Register
0x2040	HFXO_BUFOUTTRIM_CLR	RW SYNC	BUFOUT Trim Configuration Register
0x2044	HFXO_BUFOUTCTRL_CLR	RW SYNC	BUFOUT Control Register
0x2050	HFXO_CMD_CLR	W SYNC	Command Register

Offset	Name	Type	Description
0x2058	HFXO_STATUS_CLR	RH	Status Register
0x2070	HFXO_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2074	HFXO_IEN_CLR	RW	Interrupt Enable Register
0x2080	HFXO_LOCK_CLR	W	Configuration Lock Register
0x3000	HFXO_IPVERSION_TGL	R	IP Version ID
0x3010	HFXO_XTALCFG_TGL	RW SYNC	Crystal Configuration Register
0x3018	HFXO_XTALCTRL_TGL	RWH SYNC	Crystal Control Register
0x301C	HFXO_XTALCTRL1_TGL	RW SYNC	BUFOUT Crystal Control Register
0x3020	HFXO_CFG_TGL	RW SYNC	Configuration Register
0x3028	HFXO_CTRL_TGL	RWH SYNC	Control Register
0x3040	HFXO_BUFOUTTRIM_TGL	RW SYNC	BUFOUT Trim Configuration Register
0x3044	HFXO_BUFOUTCTRL_TGL	RW SYNC	BUFOUT Control Register
0x3050	HFXO_CMD_TGL	W SYNC	Command Register
0x3058	HFXO_STATUS_TGL	RH	Status Register
0x3070	HFXO_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3074	HFXO_IEN_TGL	RW	Interrupt Enable Register
0x3080	HFXO_LOCK_TGL	W	Configuration Lock Register

8.2.5 HFXO Register Description

8.2.5.1 HFXO_IPVERSION - IP Version ID

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x3																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x3	R	**IP Version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

8.2.5.2 HFXO_XTALCFG - Crystal Configuration Register

Offset	Bit Position																											
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0xB				0xB				0x0				0x0				0x20						0x20					
Access					RW				RW				RW				RW				RW						RW					
Name					TIMEOUTCBLSB				TIMEOUTSTEADY				CTUNEXOSTARTUP				CTUNEXISTARTUP				COREBIASSTARTUP						COREBIASSTARTUPI					

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:24	TIMEOUTCBLSB	0xB	RW	**Core Bias LSB Change Timeout**

wait duration for the COREBIAS change to settle out, used at each step of COREBIAS optimization algorithm

Value	Mode	Description
0	T8US	The core bias LSB change timeout is set to 8 us minimum. The maximum can be +40%.
1	T20US	The core bias LSB change timeout is set to 20 us minimum. The maximum can be +40%.
2	T41US	The core bias LSB change timeout is set to 41 us minimum. The maximum can be +40%.
3	T62US	The core bias LSB change timeout is set to 62 us minimum. The maximum can be +40%.
4	T83US	The core bias LSB change timeout is set to 83 us minimum. The maximum can be +40%.
5	T104US	The core bias LSB change timeout is set to 104 us minimum. The maximum can be +40%.
6	T125US	The core bias LSB change timeout is set to 125 us minimum. The maximum can be +40%.
7	T166US	The core bias LSB change timeout is set to 166 us minimum. The maximum can be +40%.
8	T208US	The core bias LSB change timeout is set to 208 us minimum. The maximum can be +40%.
9	T250US	The core bias LSB change timeout is set to 250 us minimum. The maximum can be +40%.
10	T333US	The core bias LSB change timeout is set to 333 us minimum. The maximum can be +40%.
11	T416US	The core bias LSB change timeout is set to 416 us minimum. The maximum can be +40%.
12	T833US	The core bias LSB change timeout is set to 833 us minimum. The maximum can be +40%.

Bit	Name	Reset	Access	Description
	13	T1250US		The core bias LSB change timeout is set to 1250 us minimum. The maximum can be +40%.
	14	T2083US		The core bias LSB change timeout is set to 2083 us minimum. The maximum can be +40%.
	15	T3750US		The core bias LSB change timeout is set to 3750 us minimum. The maximum can be +40%.
23:20	TIMEOUTSTEADY	0xB	RW	**Steady State Timeout**
	wait duration for the steady state settings to settle out			
	Value	Mode		Description
	0	T4US		The steady state timeout is set to 16 us minimum. The maximum can be +40%.
	1	T16US		The steady state timeout is set to 41 us minimum. The maximum can be +40%.
	2	T41US		The steady state timeout is set to 83 us minimum. The maximum can be +40%.
	3	T83US		The steady state timeout is set to 125 us minimum. The maximum can be +40%.
	4	T125US		The steady state timeout is set to 166 us minimum. The maximum can be +40%.
	5	T166US		The steady state timeout is set to 208 us minimum. The maximum can be +40%.
	6	T208US		The steady state timeout is set to 250 us minimum. The maximum can be +40%.
	7	T250US		The steady state timeout is set to 333 us minimum. The maximum can be +40%.
	8	T333US		The steady state timeout is set to 416 us minimum. The maximum can be +40%.
	9	T416US		The steady state timeout is set to 500 us minimum. The maximum can be +40%.
	10	T500US		The steady state timeout is set to 666 us minimum. The maximum can be +40%.
	11	T666US		The steady state timeout is set to 833 us minimum. The maximum can be +40%.
	12	T833US		The steady state timeout is set to 1666 us minimum. The maximum can be +40%.
	13	T1666US		The steady state timeout is set to 2500 us minimum. The maximum can be +40%.
	14	T2500US		The steady state timeout is set to 4166 us minimum. The maximum can be +40%.
	15	T4166US		The steady state timeout is set to 7500 us minimum. The maximum can be +40%.
19:16	CTUNEXOSTARTUP	0x0	RW	**Startup Tuning Capacitance on XO**
	4 most significant bits of CTUNEXOANA applied during startup phase			
15:12	CTUNEXISTARTUP	0x0	RW	**Startup Tuning Capacitance on XI**

Bit	Name	Reset	Access	Description
	4 most significant bits of CTUNEXIANA applied during startup phase			
11:6	COREBIASSTARTUP	0x20	RW	**Startup Core Bias Current**
	6 most significant bits of COREBIASANA applied during startup phase			
5:0	COREBIASSTARTUPI	0x20	RW	**Intermediate Startup Core Bias Current**
	6 most significant bits of COREBIASANA applied during intermediate startup phase			

8.2.5.3 HFXO_XTALCTRL - Crystal Control Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0				0x0		0x3		0x3C								0x3C								0x3C							
Access	RW				RW		RW		RW								RW								RW							
Name	SKIPCOREBIASOPT				COREDGENANA		CTUNEFIXANA		CTUNEXOANA								CTUNEXIANA								COREBIASANA							

Bit	Name	Reset	Access	Description
31	SKIPCOREBIASOPT	0x0	RW	**Skip Core Bias Optimization**
	Set to skip the core bias current optimization algorithm at next startup. Reuse the value stored in COREBIASANA. At the successful completion of core bias current optimization algorithm, hardware sets this bit to skip optimization during subsequent startup.			
30:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:26	COREDGENANA	0x0	RW	**Core Degeneration**
	Core degeneration control			

Value	Mode	Description
0	NONE	Do not apply core degeneration resistence
1	DGEN33	Apply 33 ohm core degeneration resistence
2	DGEN50	Apply 50 ohm core degeneration resistence
3	DGEN100	Apply 100 ohm core degeneration resistence

Bit	Name	Reset	Access	Description
25:24	CTUNEFIXANA	0x3	RW	**Fixed Tuning Capacitance**
	Adds or removes fixed capacitance on XI or XO			

Value	Mode	Description
0	NONE	Remove fixed capacitance on XI and XO nodes
1	XI	Adds fixed capacitance on XI node
2	XO	Adds fixed capacitance on XO node
3	BOTH	Adds fixed capacitance on both XI and XO nodes

Bit	Name	Reset	Access	Description
23:16	CTUNEXOANA	0x3C	RW	**Tuning Capacitance on XO**
	Approximately 80fF per step. 0 is min. 255 is max.			
15:8	CTUNEXIANA	0x3C	RW	**Tuning Capacitance on XI**
	Approximately 80fF per step. 0 is min. 255 is max.			
7:0	COREBIASANA	0x3C	RW	**Core Bias Current**

Bit	Name	Reset	Access	Description
	Approximately 10uA per step			

8.2.5.4 HFXO_XTALCTRL1 - BUFOUT Crystal Control Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x3C				
Access																												RW				
Name																												CTUNEXIBUFOUTANA				

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	CTUNEXIBUFOUTANA	0x3C	RW	**BUFOUT Tuning Capacitance on XI**
	Tuning Capacitance on XI when BUFOUT is ON. Approximately 80fF per step. 0 is min. 255 is max.			

8.2.5.5 HFXO_CFG - Configuration Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x1																									0x0	0x0	0x0	
Access				RW																									RW	RW	RW	
Name				FORCELFTIMEOUT																									SQBUFSCHTRGANA	ENXIDCBIASANA	MODE	

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28	FORCELFTIMEOUT	0x1	RW	**Force Low Frequency Timeout**
	For deterministic timeout, clear this bit and configure PRS to trigger based on 32kHz timer (e.g., RTC).			
27:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	SQBUFSCHTRGANA	0x0	RW	**Squaring Buffer Schmitt Trigger**
	Used in EXTCLK mode to prevent self oscillation			

Value	Mode	Description
0	DISABLE	Squaring buffer schmitt trigger is disabled
1	ENABLE	Squaring buffer schmitt trigger is enabled

2	ENXIDCBIASANA	0x0	RW	**Enable XI Internal DC Bias**
	Set to enable internal DC bias. Bit is ignored in XTAL mode.			
1:0	MODE	0x0	RW	**Crystal Oscillator Mode**
	Set this to configure the external source for the HFXO.			

Value	Mode	Description
0	XTAL	crystal oscillator
1	EXTCLK	external sinusoidal clock can be supplied on XI pin.
2	EXTCLKPKDET	external sinusoidal clock can be supplied on XI pin (peak detector used).

8.2.5.6 HFXO_CTRL - Control Register

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset						0x1	0x1	0x1						0x0	0x0	0x0	0x0				0x0					0x1	0x0	0x0	0x0	0x0		0x0
Access						RW	RW	RW						RW	RW	RW	RW				RW					RW	RW	RW	RW	RW		RW
Name						DISONDEMANDBUFOUT	DISONDEMANDPRS	DISONDEMAND						FORCEENBUFOUT	FORCEENPRS	FORCEEN	PRSSTATUSSEL1				PRSSTATUSSEL0					FORCECTUNEMAX	FORCEXO2GNDANA	FORCEXI2GNDANA	EM23ONDEMAND	KEEPWARM		BUFOUTFREEZE

Bit	Name	Reset	Access	Description
31:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26	DISONDEMANDBUFOUT	0x1	RW	**Disable On-demand For BUFOUT**
	Disable On Demand for the BUFOUT request interface.			
25	DISONDEMANDPRS	0x1	RW	**Disable On-demand For PRS**
	Disable On Demand for the PRS request interface.			
24	DISONDEMAND	0x1	RW	**Disable On-demand For Digital Clock**
	Disable On Demand for the digital clock request interface.			
23:19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18	FORCEENBUFOUT	0x0	RW	**Force BUFOUT Request**
	Force BUFOUT request.			
17	FORCEENPRS	0x0	RW	**Force PRS Oscillator Request**
	Force PRS oscillator request.			
16	FORCEEN	0x0	RW	**Force Digital Clock Request**
	Force digital clock request.			
15:12	PRSSTATUSSEL1	0x0	RW	**PRS Status 1 Output Select**
	Mux select for various status signals to be output through PRS.			

Value	Mode	Description
0	DISABLED	PRS mux outputs 0
1	ENS	PRS mux outputs enabled status
2	COREBIASOPTRDY	PRS mux outputs core bias optimization ready status
3	RDY	PRS mux outputs ready status
4	PRSRDY	PRS mux outputs PRS ready status
5	BUFOUTRDY	PRS mux outputs BUFOUT ready status

Bit	Name	Reset	Access	Description
	8	HWREQ		PRS mux outputs oscillator requested by digital clock status
	9	PRSHWREQ		PRS mux outputs oscillator requested by PRS request status
	10	BUFOUTHWREQ		PRS mux outputs oscillator requested by BUFOUT request status
11:8	PRSSTATUSSEL0	0x0	RW	**PRS Status 0 Output Select**
	Mux select for various status signals to be output through PRS.			
	Value	Mode		Description
	0	DISABLED		PRS mux outputs 0
	1	ENS		PRS mux outputs enabled status
	2	COREBIASOPTRDY		PRS mux outputs core bias optimization ready status
	3	RDY		PRS mux outputs ready status
	4	PRSRDY		PRS mux outputs PRS ready status
	5	BUFOUTRDY		PRS mux outputs BUFOUT ready status
	8	HWREQ		PRS mux outputs oscillator requested by digital clock status
	9	PRSHWREQ		PRS mux outputs oscillator requested by PRS request status
	10	BUFOUTHWREQ		PRS mux outputs oscillator requested by BUFOUT request status
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	FORCECTUNEMAX	0x1	RW	**Force Tuning Cap to Max Value**
	When oscillator is disabled, force tuning capacitor to maximum value. Set this bit to 1 in XTAL mode to prevent overshoot upon disable.			
5	FORCEXO2GNDANA	0x0	RW	**Force XO Pin to Ground**
	Set to enable grounding of XO pin.			
	Value	Mode		Description
	0	DISABLE		Disabled (not pulled)
	1	ENABLE		Enabled (pulled)
4	FORCEXI2GNDANA	0x0	RW	**Force XI Pin to Ground**
	Set to enable grounding of XI pin. Do not enable if MODE=EXTCLK and an external source is supplied.			
	Value	Mode		Description
	0	DISABLE		Disabled (not pulled)
	1	ENABLE		Enabled (pulled)
3	EM23ONDEMAND	0x0	RW	**On-demand During EM23**
	Use this bit to prevent EM23 shutdown of the module's power domain upon EM23 entry. Set this bit to 1 if on-demand requests are supposed to be honored while in EM23.			
2	KEEPWARM	0x0	RW	**Keep Warm**

Bit	Name	Reset	Access	Description
	Upon disable, if this bit is set, analog oscillator will keep running, while clock output is shutoff. Clearing this bit has no effect until the next disable event.			
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	BUFOUTFREEZE	0x0	RW	**Freeze BUFOUT Controls**
	Freeze BUFOUT Controls in current state (ON or OFF).			

8.2.5.7 HFXO_BUFOUTTRIM - BUFOUT Trim Configuration Register

Offset	Bit Position																															
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x8		
Access																													RW		
Name																													VTRTRIMANA		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	VTRTRIMANA	0x8	RW	**BUFOUT Reference Trim**
	Change this field to set bias levels between 200uA and 400uA. bias_current = VTRTCANA * VTRTRIMANA * scale_factor. The default setting corresponds to 200uA.			

8.2.5.8 HFXO_BUFOUTCTRL - BUFOUT Control Register

Offset	Bit Position																													
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0								0x6				0x4				0x3				0xC				0x1				0x5			
Access	RW								RW				RW				RW				RW				RW				RW			
Name	MINIMUMSTARTUPDELAY								TIMEOUTSTARTUP				TIMEOUTCTUNE				PEAKDETTHRESANA				XOUTGMANA				XOUTCFANA				XOUTBIASANA			

Bit	Name	Reset	Access	Description
31	MINIMUMSTARTUPDE-LAY	0x0	RW	**Minimum Startup Delay**
	If set, BUFOUT does not start until timeout expires. This prevents waste of power if BUFOUT is ready too early.			
30:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:20	TIMEOUTSTARTUP	0x6	RW	**Oscillator Startup Timeout**
	Wait duration for the typical oscillator startup sequence to prevent BUFOUT starting too early, used when MINIMUMSTARTUPDELAY is set.			

Value	Mode	Description
0	T42US	The oscillator startup timeout is set to 42 us minimum. The maximum can be +40%.
1	T83US	The oscillator startup timeout is set to 83 us minimum. The maximum can be +40%.
2	T108US	The oscillator startup timeout is set to 108 us minimum. The maximum can be +40%.
3	T133US	The oscillator startup timeout is set to 133 us minimum. The maximum can be +40%.
4	T158US	The oscillator startup timeout is set to 158 us minimum. The maximum can be +40%.
5	T183US	The oscillator startup timeout is set to 183 us minimum. The maximum can be +40%.
6	T208US	The oscillator startup timeout is set to 208 us minimum. The maximum can be +40%.
7	T233US	The oscillator startup timeout is set to 233 us minimum. The maximum can be +40%.
8	T258US	The oscillator startup timeout is set to 258 us minimum. The maximum can be +40%.
9	T283US	The oscillator startup timeout is set to 283 us minimum. The maximum can be +40%.

Bit	Name	Reset	Access	Description
	10	T333US		The oscillator startup timeout is set to 333 us minimum. The maximum can be +40%.
	11	T375US		The oscillator startup timeout is set to 375 us minimum. The maximum can be +40%.
	12	T417US		The oscillator startup timeout is set to 417 us minimum. The maximum can be +40%.
	13	T458US		The oscillator startup timeout is set to 458 us minimum. The maximum can be +40%.
	14	T500US		The oscillator startup timeout is set to 500 us minimum. The maximum can be +40%.
	15	T667US		The oscillator startup timeout is set to 667 us minimum. The maximum can be +40%.
19:16	TIMEOUTCTUNE	0x4	RW	**Tuning Cap Change Timeout**

Wait duration for the CTUNE change to settle out, used when CTUNE changes as result of enabling BUFOUT.

Value	Mode	Description
0	T2US	The tuning cap change timeout is set to 2 us minimum. The maximum can be +40%.
1	T5US	The tuning cap change timeout is set to 5 us minimum. The maximum can be +40%.
2	T10US	The tuning cap change timeout is set to 10 us minimum. The maximum can be +40%.
3	T16US	The tuning cap change timeout is set to 16 us minimum. The maximum can be +40%.
4	T21US	The tuning cap change timeout is set to 21 us minimum. The maximum can be +40%.
5	T26US	The tuning cap change timeout is set to 26 us minimum. The maximum can be +40%.
6	T31US	The tuning cap change timeout is set to 31 us minimum. The maximum can be +40%.
7	T42US	The tuning cap change timeout is set to 42 us minimum. The maximum can be +40%.
8	T52US	The tuning cap change timeout is set to 52 us minimum. The maximum can be +40%.
9	T63US	The tuning cap change timeout is set to 63 us minimum. The maximum can be +40%.
10	T83US	The tuning cap change timeout is set to 83 us minimum. The maximum can be +40%.
11	T104US	The tuning cap change timeout is set to 104 us minimum. The maximum can be +40%.
12	T208US	The tuning cap change timeout is set to 208 us minimum. The maximum can be +40%.
13	T313US	The tuning cap change timeout is set to 313 us minimum. The maximum can be +40%.
14	T521US	The tuning cap change timeout is set to 521 us minimum. The maximum can be +40%.

Bit	Name	Reset	Access	Description
	15	T938US		The tuning cap change timeout is set to 938 us minimum. The maximum can be +40%.
15:12	PEAKDETTHRESANA	0x3	RW	**Peak Detector Threshold for XOUT**
	Sets the peak detector threshold for BUFOUT.			
	Value	Mode		Description
	0	V105MV		
	1	V132MV		
	2	V157MV		
	3	V184MV		
	4	V210MV		
	5	V236MV		
	6	V262MV		
	7	V289MV		
	8	V315MV		
	9	V341MV		
	10	V367MV		
	11	V394MV		
	12	V420MV		
	13	V446MV		
	14	V472MV		
	15	V499MV		
11:8	XOUTGMANA	0xC	RW	
7:4	XOUTCFANA	0x1	RW	**Buffer Gain**
	Buffer gain.			
3:0	XOUTBIASANA	0x5	RW	**Driver Bias Current**
	Driver bias current.			

8.2.5.9 HFXO_CMD - Command Register

Offset	Bit Position																															
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																W(nB)
Name																																COREBIASOPT

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	COREBIASOPT	0x0	W(nB)	**Core Bias Optimizaton** On devices with a radio, this bit is used to start the core bias current optimization algorithm and run it one time. Optimization should be executed if the temperature changes by more than 40 degC. Do not run this command while the radio is in RX or TX modes. Do not issue this command more than once until COREBIASOPTRDY is asserted, or the previous command may be cancelled.

8.2.5.10 HFXO_STATUS - Status Register

Offset	Bit Position																													
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0									0x0	0x0	0x0		0x0	0x0	0x0												0x0	0x0	0x0	0x0
Access	R	R									R	R	R		R	R	R												R	R	R	R
Name	LOCK	SYNCBUSY									BUFOUTHWREQ	PRSHWREQ	ISWARM		HWREQ	ENS	BUFOUTFROZEN												BUFOUTRDY	PRSRDY	COREBIASOPTRDY	RDY

Bit	Name	Reset	Access	Description
31	LOCK	0x0	R	**Configuration Lock Status**
	Indicates the current status of configuration lock.			

Value	Mode	Description
0	UNLOCKED	Configuration lock is unlocked
1	LOCKED	Configuration lock is locked

30	SYNCBUSY	0x0	R	**Sync Busy**
	Indicates synchronization is ongoing.			
29:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	BUFOUTHWREQ	0x0	R	**Oscillator Requested by BUFOUT Request**
	Oscillator is requested by the BUFOUT oscillator request interface.			
20	PRSHWREQ	0x0	R	**Oscillator Requested by PRS Request**
	Oscillator is requested by the PRS oscillator request interface.			
19	ISWARM	0x0	R	**Oscillator Is Kept Warm**
	Oscillator is currently kept in warm state. Re-enable from warm state skips startup sequence.			
18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	HWREQ	0x0	R	**Oscillator Requested by Digital Clock**
	Oscillator is requested by digital clock request interface.			
16	ENS	0x0	R	**Enabled Status**
	Oscillator is enabled.			
15	BUFOUTFROZEN	0x0	R	**BUFOUT Frozen**
	FSM is frozen with respect to starting BUFOUT enable or disable sequences.			
14:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	BUFOUTRDY	0x0	R	**BUFOUT Ready Status**
	The BUFOUT clock is ready.			

Bit	Name	Reset	Access	Description
2	PRSRDY	0x0	R	**PRS Ready Status**
	The PRS oscillator startup is ready.			
1	COREBIASOPTRDY	0x0	R	**Core Bias Optimization Ready**
	Core bias current optimization algorithm is complete.			
0	RDY	0x0	R	**Ready Status**
	The digital clock branch (osc.clk_qual) is ready.			

8.2.5.11 HFXO_IF - Interrupt Flag Register

Offset	Bit Position																													
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0	0x0	0x0						0x0	0x0					0x0												0x0	0x0	0x0	0x0
Access	RW	RW	RW	RW	RW						RW	RW					RW												RW	RW	RW	RW
Name	COREBIASOPTERR	LFTIMEOUTERR	DNSERR	BUFOUTDNSERR	BUFOUTFREEZEERR						BUFOUTERR	PRSERR					BUFOUTFROZEN												BUFOUTRDY	PRSRDY	COREBIASOPTRDY	RDY

Bit	Name	Reset	Access	Description
31	COREBIASOPTERR	0x0	RW	**Core Bias Optimization Error Interrupt**
	Core bias current optimization algorithm fails to complete.			
30	LFTIMEOUTERR	0x0	RW	**Low Frequency Timeout Error Interrupt**
	Low frequency timeout triggers before the steady state timeout triggers.			
29	DNSERR	0x0	RW	**Did Not Start Error Interrupt**
	Crystal oscillator fails to startup.			
28	BUFOUTDNSERR	0x0	RW	**BUFOUT Did Not Start Error Interrupt**
	BUFOUT fails to startup.			
27	BUFOUTFREEZEERR	0x0	RW	**BUFOUT Freeze Error Interrupt**
	BUFOUTFREEZE should not be set when HWREQ is low as this can prevent service to companion chip indefinitely.			
26:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	BUFOUTERR	0x0	RW	**BUFOUT Request Error Interrupt**
	BUFOUT request is asserted while oscillator is forced to shutdown.			
20	PRSERR	0x0	RW	**PRS Requset Error Interrupt**
	PRS request is asserted while oscillator is forced to shutdown.			
19:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15	BUFOUTFROZEN	0x0	RW	**BUFOUT FROZEN Interrupt**
	FSM is frozen with respect to starting BUFOUT enable or disable sequences.			
14:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	BUFOUTRDY	0x0	RW	**BUFOUT Ready Interrupt**
	The BUFOUT clock is ready.			
2	PRSRDY	0x0	RW	**PRS Ready Interrupt**
	The PRS oscillator startup is ready.			
1	COREBIASOPTRDY	0x0	RW	**Core Bias Optimization Ready Interrupt**

Bit	Name	Reset	Access	Description
	Core bias current optimization algorithm is complete.			
0	RDY	0x0	RW	**Digital Clock Ready Interrupt**
	The digital clock branch (osc.clk_qual) is ready.			

8.2.5.12 HFXO_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x074	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0	0x0	0x0						0x0	0x0					0x0												0x0	0x0	0x0	0x0
Access	RW	RW	RW	RW	RW						RW	RW					RW												RW	RW	RW	RW
Name	COREBIASOPTERR	LFTIMEOUTERR	DNSERR	BUFOUTDNSERR	BUFOUTFREEZEERR						BUFOUTERR	PRSERR					BUFOUTFROZEN												BUFOUTRDY	PRSRDY	COREBIASOPTRDY	RDY

Bit	Name	Reset	Access	Description
31	COREBIASOPTERR	0x0	RW	**Core Bias Optimization Error Interrupt**
	Core bias current optimization algorithm fails to complete.			
30	LFTIMEOUTERR	0x0	RW	**Low Frequency Timeout Error Interrupt**
	Low frequency timeout triggers before the steady state timeout triggers.			
29	DNSERR	0x0	RW	**Did Not Start Error Interrupt**
	Crystal oscillator fails to startup.			
28	BUFOUTDNSERR	0x0	RW	**BUFOUT Did Not Start Error Interrupt**
	BUFOUT fails to startup.			
27	BUFOUTFREEZEERR	0x0	RW	**BUFOUT Freeze Error Interrupt**
	BUFOUTFREEZE should not be set when HWREQ is low as this can prevent service to companion chip indefinitely.			
26:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	BUFOUTERR	0x0	RW	**BUFOUT Request Error Interrupt**
	BUFOUT request is asserted while oscillator is forced to shutdown.			
20	PRSERR	0x0	RW	**PRS Requset Error Interrupt**
	PRS request is asserted while oscillator is forced to shutdown.			
19:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15	BUFOUTFROZEN	0x0	RW	**BUFOUT FROZEN Interrupt**
	FSM is frozen with respect to starting BUFOUT enable or disable sequences.			
14:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	BUFOUTRDY	0x0	RW	**BUFOUT Ready Interrupt**
	The BUFOUT clock is ready.			
2	PRSRDY	0x0	RW	**PRS Ready Interrupt**
	The PRS oscillator startup is ready.			
1	COREBIASOPTRDY	0x0	RW	**Core Bias Optimization Ready Interrupt**

Bit	Name	Reset	Access	Description
	Core bias current optimization algorithm is complete.			
0	RDY	0x0	RW	**Digital Clock Ready Interrupt**
	The digital clock branch (osc.clk_qual) is ready.			

8.2.5.13 HFXO_LOCK - Configuration Lock Register

Offset	Bit Position																													
0x080	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x580E															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x580E	W	**Configuration Lock Key**
	Write any other value than the unlock code to lock registers from editing. Write the unlock code to unlock.			

Value	Mode	Description
22542	UNLOCK	Write this value to unlock

8.3 HFRCO - High-Frequency RC Oscillator

8.3.1 Introduction

The HFRCO is a calibrated internal High Frequency RC oscillator.

8.3.2 Features

- 1 MHz - 80 MHz High Frequency RC Oscillator with DPLL working in EM01 (HFRCO0/HFRCODPLL)
- 1 MHz - 40 MHz High Frequency RC Oscillator working in EM23 (HFRCOEM23)
- Low start-up time
- Run-time band change or tuning

8.3.3 Functional Description

8.3.3.1 Start-up

The HFRCO starts up quickly in a few micro-seconds (refer to device data sheet for start-up time specifications.) After the start-up time, the RDY status bit will go high and the RDY interrupt will be triggered. It can take another two clock cycles for the clock to propagate through the CMU before the clock is seen by peripherals.

8.3.3.2 On-Demand Clocking

Software can request to enable the HFRCO by setting the HFRCO_CTRL.FORCEEN bit field. The HFRCO can also optionally be configured via the HFRCO_CTRL.DISONDEMAND to shut down when no hardware request is present. This is known as on-demand clocking and allows the oscillator to be controlled without any software intervention. This means that HFRCO receives a request for clock from the CMU whenever the oscillator clock is needed. These requests can come at any time from any power domain (depending on the which peripheral is requesting the clock.)

8.3.3.2.1 EM2/EM3 On-Demand Operation (HFRCOEM23)

The HFRCOEM23 can be used by certain peripherals as an on-demand, high-speed clock source in energy modes down to EM3. To enable operation as an on-demand clock in EM2 and EM3, the EM23ONDEMAND bit in the CTRL register should be set to 1. Setting this bit ensures that the associated PD0 power domain will remain active and allow the oscillator to honor the requesst.

Note: This feature is not available on the HFRCODPLL oscillator, which only operates in EM0 and EM1.

8.3.3.3 Calibration

Several different frequencies are calibrated during production test on every device. In order to use a factory-calibrated value, software must read the value from the appropriate location in the DEVINFO page and write it to the CAL register.

The TUNING and FINETUNING bit fields in the CAL register can be used to trim HFRCO manually.

Software may write the CAL register at any time. If there is already a frequency updating occuring, the current change would apply when the previous update is done. FREQBSY in STATUS register indicates if the updating is finished.

The minimum and maximum frequencies attainable for each setting of the FREQRANGE field are listed in the device data sheet.

Table 8.1. HFRCODPLL Calibration Frequencies

DEVINFO Location	Target Frequency
HFRCODPLLCAL0	4 MHz
HFRCODPLLCAL3	7 MHz
HFRCODPLLCAL6	13 MHz
HFRCODPLLCAL7	16 MHz
HFRCODPLLCAL8	19 MHz (default)
HFRCODPLLCAL10	26 MHz
HFRCODPLLCAL11	32 MHz
HFRCODPLLCAL12	38 MHz
HFRCODPLLCAL13	48 MHz
HFRCODPLLCAL14	56 MHz
HFRCODPLLCAL15	64 MHz
HFRCODPLLCAL16	80 MHz

Table 8.2. HFRCOEM23 Calibration Frequencies

DEVINFO Location	Target Frequency
HFRCOEM23CAL0	4 MHz
HFRCOEM23CAL1	5 MHz
HFRCOEM23CAL3	7 MHz
HFRCOEM23CAL4	10 MHz
HFRCOEM23CAL6	13 MHz
HFRCOEM23CAL7	16 MHz
HFRCOEM23CAL8	19 MHz (default)
HFRCOEM23CAL9	20 MHz
HFRCOEM23CAL10	26 MHz
HFRCOEM23CAL11	32 MHz
HFRCOEM23CAL12	40 MHz

8.3.3.4 Interrupts

HFRCO has one interrupt: IF.RDY. RDY is triggered when the timeout has finished and the qualified HFRCO clock is ready. The clock is gated until it is ready.

8.3.3.5 Status Flags

8.3.3.5.1 FREQBSY

The FREQBSY bit indicates the HFRCO is busy updating its frequency after writing to the CAL register. The FREQBSY bit should be used whenever frequency is changed. E.g. After software writes to the CAL register, FREQBSY would assert immediately. Software should wait for FREQBSY to be zero before attempting to write to the CAL register again.

For band-change, FREQBSY would not de-assert until after the timeout upon being re-enabled.

For normal start-up, FREQBSY would not assert.

When DPLL is on, FREQBSY would not assert as the frequency change is not caused by writing to the CAL register. When disabling DPLL the last tuning value is written back to the CAL register, which will assert FREQBSY.

8.3.3.5.2 ENS

ENS indicates the HFRCO is enabled. This flag is used to check if the HFRCO is enabled by any requester.

Note: When a band change occurs, the HFRCO is disabled and re-enabled. This will cause the ENS bit to briefly de-assert.

8.3.3.5.3 RDY

RDY indicates HFRCO is enabled and start-up timeout has exceeded. Used to check if the HFRCO clock is ready after enable.

Changing bands will de-assert RDY as the oscillator must reset and start up again.

8.3.3.5.4 SYNCBUSY

SYNCBUSY indicates ongoing synchronization of CAL register fields. Same as all other modules.

8.3.3.6 Forced Oscillator Control

The HFRCO can be forced on and off using the FORCEEN and DISONDEMAND bits in the CTRL register.

Setting FORCEEN will force the oscillator core to run, but peripherals will still need to request the clock to un-gate the clock signal.

8.3.3.7 Oscillator Modes

The HFRCO has three modes of operation, an **on-demand** mode (which is the normal software use case), a **force on** and a **force off** mode.

In **on-demand** mode the oscillator will start whenever a peripheral requests it. Which in most cases is whenever the peripheral is enabled.

In **force on** mode the analog core will run independently of whether it is requested or not. This can be useful for measuring analog current without any digital load on the clocks.

In **force off** mode, the analog core will be shut off independently of whether it is requested or not. This can be useful for changing analog test settings without risking glitches on the clock.

The DISONDEMAND bit can also be used to give software full control over the clock for exceptional cases where software control is desired.

Table 8.3. Oscillator modes

Bit Field	FORCEEN	DISONDEMAND
On-Demand (normal operation)	0	0
Forced On	1	X
Forced Off	0	1

8.3.4 HFRCO Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	HFRCO_IPVERSION	R	IP Version ID
0x004	HFRCO_CTRL	RW	Ctrl Register
0x008	HFRCO_CAL	RWH SYNC	Calibration Register
0x00C	HFRCO_STATUS	RH	Status Register
0x010	HFRCO_IF	RWH INTFLAG	Interrupt Flag Register
0x014	HFRCO_IEN	RW	Interrupt Enable Register
0x01C	HFRCO_LOCK	W	Lock Register
0x1000	HFRCO_IPVERSION_SET	R	IP Version ID
0x1004	HFRCO_CTRL_SET	RW	Ctrl Register
0x1008	HFRCO_CAL_SET	RWH SYNC	Calibration Register
0x100C	HFRCO_STATUS_SET	RH	Status Register
0x1010	HFRCO_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1014	HFRCO_IEN_SET	RW	Interrupt Enable Register
0x101C	HFRCO_LOCK_SET	W	Lock Register
0x2000	HFRCO_IPVERSION_CLR	R	IP Version ID
0x2004	HFRCO_CTRL_CLR	RW	Ctrl Register
0x2008	HFRCO_CAL_CLR	RWH SYNC	Calibration Register
0x200C	HFRCO_STATUS_CLR	RH	Status Register
0x2010	HFRCO_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2014	HFRCO_IEN_CLR	RW	Interrupt Enable Register
0x201C	HFRCO_LOCK_CLR	W	Lock Register
0x3000	HFRCO_IPVERSION_TGL	R	IP Version ID
0x3004	HFRCO_CTRL_TGL	RW	Ctrl Register
0x3008	HFRCO_CAL_TGL	RWH SYNC	Calibration Register
0x300C	HFRCO_STATUS_TGL	RH	Status Register
0x3010	HFRCO_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3014	HFRCO_IEN_TGL	RW	Interrupt Enable Register
0x301C	HFRCO_LOCK_TGL	W	Lock Register

8.3.5 HFRCO Register Description

8.3.5.1 HFRCO_IPVERSION - IP Version ID

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x2																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x2	R	**IP Version**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

8.3.5.2 HFRCO_CTRL - Ctrl Register

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														EM23ONDEMAND	DISONDEMAND	FORCEEN

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	EM23ONDEMAND	0x0	RW	**EM23 On-demand**
	Use this bit to prevent EM23 shutdown of the HFRCOEM23 low power domain (PD0C) upon EM23 entry. Set this bit to 1 if on-demand requests are supposed to be honored while in EM23.			
1	DISONDEMAND	0x0	RW	**Disable On-demand**
	Setting this bit disable HFRCO on-demand feature			
0	FORCEEN	0x0	RW	**Force Enable**
	Setting this bit force HFRCO enabled			

8.3.5.3 HFRCO_CAL - Calibration Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xA				0x2		0x0		0x3			0x8					0x1		0x1F							0x7F						
Access	RW				RW		RW		RW			RW					RW		RW							RW						
Name	IREFTC				CMPSEL		CLKDIV		CMPBIAS			FREQRANGE					LDOHP		FINETUNING							TUNING						

Bit	Name	Reset	Access	Description
31:28	IREFTC	0xA	RW	**Tempco Trim on Comparator Current**
	Writing this field adjusts the temperature coefficient trim on comparator current.			
27:26	CMPSEL	0x2	RW	**Comparator Load Select**
	Writing this field adjusts the active load for comparators.			
25:24	CLKDIV	0x0	RW	**Locally Divide HFRCO Clock Output**
	Writing this field configures the HFRCO clock output divider.			

Value	Mode	Description
0	DIV1	Divide by 1.
1	DIV2	Divide by 2.
2	DIV4	Divide by 4.

23:21	CMPBIAS	0x3	RW	**Comparator Bias Current**
	Writing this field adjusts the HFRCO comparator bias current.			
20:16	FREQRANGE	0x8	RW	**Frequency Range**
	Writing this field adjusts the HFRCO frequency range.			
15	LDOHP	0x1	RW	**LDO High Power Mode**
	Settings this bit puts the HFRCO LDO in high power mode.			
14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:8	FINETUNING	0x1F	RW	**Fine Tuning Value**
	Writing this field adjusts the HFRCO fine tuning value. Higher value means lower frequency.			
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:0	TUNING	0x7F	RW	**Tuning Value**
	Writing this field adjusts the HFRCO tuning value. Higher value means lower frequency.			

8.3.5.4 HFRCO_STATUS - Status Register

Offset	Bit Position																														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0															0x0														0x0	0x0	0x0
Access	R															R														R	R	R
Name	LOCK															ENS														SYNCBUSY	FREQBSY	RDY

Bit	Name	Reset	Access	Description
31	LOCK	0x0	R	**Lock Status**
	If set, all HFRCO lockable registers are locked.			

Value	Mode	Description
0	UNLOCKED	HFRCO is unlocked
1	LOCKED	HFRCO is locked

30:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	ENS	0x0	R	**Enable Status**
	HFRCO is enabled.			
15:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	SYNCBUSY	0x0	R	**Synchronization Busy**
	This bit is set when there is an ongoing synchronization of CAL register bitfields.			
1	FREQBSY	0x0	R	**Frequency Updating Busy**
	HFRCO is busy updating frequency.			
0	RDY	0x0	R	**Ready**
	HFRCO is enabled and start-up time has exceeded.			

8.3.5.5 HFRCO_IF - Interrupt Flag Register

Offset	Bit Position																														
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																RDY

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	RDY	0x0	RW	**Ready Interrupt Flag**
	Set when HFRCO is ready (start-up time exceeded).			

8.3.5.6 HFRCO_IEN - Interrupt Enable Register

Offset	Bit Position																														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																RDY

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	RDY	0x0	RW	**RDY Interrupt Enable**
	Enable/disable the RDY interrupt			

8.3.5.7 HFRCO_LOCK - Lock Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x8195						
Access																									W						
Name																									LOCKKEY						

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x8195	W	**Lock Key**

Write any other value than the unlock code to lock registers from editing. Write the unlock code to unlock.

Value	Mode	Description
33173	UNLOCK	Unlock code

8.4 DPLL - Digital Phased Locked Loop

8.4.1 Introduction

The Digital Phase-Locked Loop (DPLL) uses a reference clock to generate a desired clock frequency at a specified ratio to the reference clock.

8.4.2 Features

- Frequency Lock Mode
- Phase-Lock Mode
- Output frequency = F_{REF}*(N+1)/(M+1), where N and M are 12-bit values
- Very fast lock time
- Very fast transient tracking
- Low output jitter
- Lock detection with an interrupt
- Lock fail detection with interrupts

8.4.3 Functional Description

8.4.3.1 Enabling and Disabling

The DPLL can be enabled and disabled by software via the DPLL_EN register. Before enabling DPLL, software should:

1. Select reference clock by setting the CLKSEL field in CMU_DPLLREFCLKCTRL.
2. The CMU should not be running from the HFRCO. If necessary, the CMU should switch to the FSRCO until after the DPLL has locked to avoid over-clocking due to overshoot. If necessary, select FSRCO or HFXO in the CMU_SYSCLKCTRL register CLKSEL field.
3. Configure the DPLL.
4. Make certain that the ENS bit in DPLL_STATUS is low.

The DPLL is disabled automatically when entering EM2, EM3, or EM4. Note that disabling the DPLL will not automatically turn off the reference clock. The CLKSEF field in CMU_DPLLREFCLKCTRL must be set to DISABLED before entering EM2 or the selected REFCLK may continue to run in EM2.

8.4.3.2 Lock Modes

The DPLL provides two lock modes, referred to as frequency-lock loop mode (FREQLL) and phase-lock loop mode (PHASELL). FREQLL mode keeps the DCO frequency-locked to the reference clock, which means the DCO frequency will be accurate. However, the phase error can accumulate over time and cause a non-zero average frequency error. FREQLL mode also provides better jitter and transient performance. PHASELL mode keeps the DCO phase-locked to the reference clock, which means the phase error does not accumulate over time, which makes the average frequency error zero. FREQLL mode is usually sufficient unless specific phase requirement exists.

8.4.3.3 Configurations

The formula for the DPLL output frequqncy is FREF*(N+1)/(M+1). The user should calculate N and M in DPLL_CFG1 to achieve the target frequency. Note that with a larger value of N, the DCO lock time would increase and DCO jitter would decrease. Both effects are approximately linear. This relationship can be used to select N for a given application to strike a compromise between lock time and output jitter. For example if an ratio of 3 is desired, the DPLL could be configured as {N=599, M=199} for fast lock time but high jitter, or as {N=2999, M=999} for lower jitter but longer lock time.

Note: All configuration settings should be done before enabling the DPLL. They should not be changed when DPLL is running. The final tuning values can be read back from TUNING and FINETUNING in HFRCO_CAL, after DPLL is disabled and DPLLENS in DPLL_STATUS is low.

8.4.3.4 Lock Detection

The DPLL has 3 different types of output events: ready, lock fail due to period underflow, and lock fail due to period overflow. Each of the events has its own interrupt flag. DPLLRDY is set when DPLL successfully locks to the reference clock based on the software configuration. DPLLLOCKFAILLOW is set when the DPLL fails to lock because the period lower boundary is hit. DPLLLOCKFAILHIGH is set when the DPLL fails to lock because the period upper boundary is hit. If the interrupt flags are set and the corresponding interrupt enable bits in DPLL_IEN are set, the DPLL will request an interrupt. Based on different interrupt events, software should take different actions:

- If the DPLLRDY interrupt is received first, it means target clock is ready and it is safe to switch to use DCO's output.
- If the DPLLLOCKFAILLOW interrupt is received first, it indicates the RANGE in HFRCO_CAL is too small. Software should disable the DPLL and write a larger value to RANGE, then enable the DPLL again to lock.
- If the DPLLLOCKFAILHIGH interrupt is received first, it indicates the RANGE in HFRCO_CAL is too large. Software should disable DPLL and write a smaller value to RANGE, then enable DPLL again to lock.
- If the DPLLRDY interrupt is received first and then DPLLLOCKFAILLOW or DPLLLOCKFAILHIGH is received later, it means reference clock drifted over 1% and the DPLL has lost its locked status.
  - If AUTORECOVER in DPLL_CFG is not set, software should disable the DPLL and enable DPLL again to lock.
  - If AUTORECOVER in DPLL_CFG is set, hardware will re-lock automatically. When the target frequency is near the boundary of a range, the drift may cause underflow or overflow. In this case the fail interrupt will still be received. Software should disable the DPLL and modify RANGE in HFRCO_CAL in corresponding direction, depending on whether the DPLLLOCKFAILLOW or DPLLLOCKFAILHIGH bit is set. Then enable DPLL again to lock.

8.4.4 DPLL Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	DPLL_IPVERSION	R	IP Version
0x004	DPLL_EN	RW ENABLE	Enable
0x008	DPLL_CFG	RW CONFIG	Config
0x00C	DPLL_CFG1	RW CONFIG	Config1
0x010	DPLL_IF	RWH INTFLAG	Interrupt Flag
0x014	DPLL_IEN	RW	Interrupt Enable
0x018	DPLL_STATUS	RH	Status
0x024	DPLL_LOCK	W	Lock
0x1000	DPLL_IPVERSION_SET	R	IP Version
0x1004	DPLL_EN_SET	RW ENABLE	Enable
0x1008	DPLL_CFG_SET	RW CONFIG	Config
0x100C	DPLL_CFG1_SET	RW CONFIG	Config1
0x1010	DPLL_IF_SET	RWH INTFLAG	Interrupt Flag
0x1014	DPLL_IEN_SET	RW	Interrupt Enable
0x1018	DPLL_STATUS_SET	RH	Status
0x1024	DPLL_LOCK_SET	W	Lock
0x2000	DPLL_IPVERSION_CLR	R	IP Version
0x2004	DPLL_EN_CLR	RW ENABLE	Enable
0x2008	DPLL_CFG_CLR	RW CONFIG	Config
0x200C	DPLL_CFG1_CLR	RW CONFIG	Config1
0x2010	DPLL_IF_CLR	RWH INTFLAG	Interrupt Flag
0x2014	DPLL_IEN_CLR	RW	Interrupt Enable
0x2018	DPLL_STATUS_CLR	RH	Status
0x2024	DPLL_LOCK_CLR	W	Lock
0x3000	DPLL_IPVERSION_TGL	R	IP Version
0x3004	DPLL_EN_TGL	RW ENABLE	Enable
0x3008	DPLL_CFG_TGL	RW CONFIG	Config
0x300C	DPLL_CFG1_TGL	RW CONFIG	Config1
0x3010	DPLL_IF_TGL	RWH INTFLAG	Interrupt Flag
0x3014	DPLL_IEN_TGL	RW	Interrupt Enable
0x3018	DPLL_STATUS_TGL	RH	Status
0x3024	DPLL_LOCK_TGL	W	Lock

8.4.5 DPLL Register Description

8.4.5.1 DPLL_IPVERSION - IP Version

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP Version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

8.4.5.2 DPLL_EN - Enable

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement Busy Status**
	When EN is cleared, DISABLING status is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs and FIFO			
0	EN	0x0	RW	**Module Enable**
	The ENABLE bit enables the module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit.			

8.4.5.3 DPLL_CFG - Config

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																										0x0				0x0	0x0	0x0
Access																										RW				RW	RW	RW
Name																										DITHEN				AUTORECOVER	EDGESEL	MODE

Bit	Name	Reset	Access	Description
31:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	DITHEN	0x0	RW	**Dither Enable Control**
	Set to enable dither function			
5:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	AUTORECOVER	0x0	RW	**Automatic Recovery Control**
	Set to enable automatic recovery function			
1	EDGESEL	0x0	RW	**Reference Edge Select**
	This bit controls which edge of reference is detected			
0	MODE	0x0	RW	**Operating Mode Control**
	This bit controls which mode DPLL is operating when enabled			

Value	Mode	Description
0	FLL	Frequency Lock Mode
1	PLL	Phase Lock Mode

8.4.5.4 DPLL_CFG1 - Config1

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0																0x0											
Access					RW																RW											
Name					N																M										

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:16	N	0x0	RW	**Factor N**
	The locked DCO frequency is given by: Fdco = Fref * (N + 1)/(M+1). N is required to be larger than 300.			
15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:0	M	0x0	RW	**Factor M**
	The locked DCO frequency is given by: Fdco = Fref * (N + 1)/(M+1). M can be any value.			

8.4.5.5 DPLL_IF - Interrupt Flag

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														LOCKFAILHIGH	LOCKFAILLOW	LOCK

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	LOCKFAILHIGH	0x0	RW	**Lock Failure High Interrupt Flag**
	Set when DPLL fail to lock because of period overflow.			
1	LOCKFAILLOW	0x0	RW	**Lock Failure Low Interrupt Flag**
	Set when DPLL fail to lock because of period underflow.			
0	LOCK	0x0	RW	**Lock Interrupt Flag**
	Set when DPLL achieve the lock.			

8.4.5.6 DPLL_IEN - Interrupt Enable

Offset	Bit Position																														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														LOCKFAILHIGH	LOCKFAILLOW	LOCK

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	LOCKFAILHIGH	0x0	RW	**LOCKFAILHIGH Interrupt Enable**
	LOCKFAILHIGH Interrupt Enable			
1	LOCKFAILLOW	0x0	RW	**LOCKFAILLOW Interrupe Enable**
	LOCKFAILLOW Interrupe Enable			
0	LOCK	0x0	RW	**LOCK interrupt Enable**
	LOCK interrupt Enable			

8.4.5.7 DPLL_STATUS - Status

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																														0x0	0x0
Access	R																														R	R
Name	LOCK																														ENS	RDY

Bit	Name	Reset	Access	Description
31	LOCK	0x0	R	**Lock Status**
	Indicates the current status of configuration lock			

Value	Mode	Description
0	UNLOCKED	DPLL is unlocked
1	LOCKED	DPLL is locked

30:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	ENS	0x0	R	**Enable Status**
	DPLL is enabled.			
0	RDY	0x0	R	**Ready Status**
	DPLL is enabled and locked.			

8.4.5.8 DPLL_LOCK - Lock

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x7102															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x7102	W	**Lock Key**
	Write any other value than the unlock code to lock registers from editing. Write the unlock code to unlock.			

Value	Mode	Description
28930	UNLOCK	Unlock code

8.5 LFXO - Low-Frequency Crystal Oscillator

8.5.1 Introduction

The Low Frequency Crystal Oscillator (LFXO) uses an external 32.768 kHz crystal to provide an accurate low-frequency clock. The module is available in all energy modes, except EM3. The main interaction is with the CMU through the clock requesting mechanism.

8.5.2 Features

High-level features.

- Crystal calibration
- Functional in all energy modes, except EM3
- Failure detection and EM4WU
- External CMOS mode
- Edge interrupts and EM2WU
- On-demand oscillator enabling

8.5.3 Functional Description

8.5.3.1 Modes

The LFXO can be used in three different modes. The mode can be programmed by setting MODE bit field in the LFXO_CFG register. If MODE is set to XTAL, the LFXO is programmed to operate in crystal mode and a 32.768 kHz crystal oscillator should be connected to LF crystal pads, LFXTAL_I and LFXTAL_O (see the device data sheet for details). If MODE is set to BUFEXTCLK, the LFXO is programmed to operate in external sine mode and the sine wave should be supplied to LFXTAL_I pin. If MODE is set to DIGEXTCLK, LFXO is programmed to operate in external CMOS mode and the external 32.768 kHz clock should be provided on LFXTAL_I pin. See the register descriptions for more details.

8.5.3.2 Enabling

There are two ways to turn on the LFXO clock. One is to turn it on in FORCEON mode by setting FORCEEN bit to 1 in LFXO_CTRL register. Another is to keep it ready to be turned on in ONDEMAND mode by setting FORCEEN bit to 0 and DISONDEMAND bit to 0 in LFXO_CTRL register. This means that the oscillator will be off unless its clock requested. When a peripheral requests the clock, hardware will automatically enable the LFXO without any software intervention. The oscillator will remain on as long as the peripheral requests it. DISONDEMAND setting does not have any impact when FORCEEN set to 1. LFXO is in FORCEOFF mode when FORCEEN set to 0 and DISONDEMAND set to 1. In FORCEOFF mode all requests are blocked and LFXO will not generate the clock. The LFXO clock is available in all energy modes, except EM3.

8.5.3.3 Clock Qualification

Once the LFXO is enabled, the clock should not be used until it has had time to stabilize. Therefore, a number of cycles are required to qualify the clock. Before the clock is qualified, no clock requesters will receive the LFXO clock. The number of cycles used to qualify the clock can be programmed by setting the TIMEOUT bit field in the LFXO_CFG register. The TIMEOUT default value is set to 32,728 cycles, which is much more than necessary for stabilization. The stabilization time required will depend on the particular crystal, oscillator settings, and frequency accuracy requirements. A value of 4096 clocks is generally recommended for most applications. A low timeout of 2 cycles may be used in DIGEXTCLK mode in order to filter out the first glitch from the pad. The 2 clock cycle timeout should not be used with crystals. There are two status bits and one interrupt associated with enabling the oscillator and qualifying its clock. Once the oscillator gets enabled the ENS bit in LFXO_STATUS register will be set high. Note that due to the nature of on demand clocking, the oscillator can be enabled anytime, so if software reads ENS low it is not safe to assume that ENS stays low during the next instruction. It is only safe to assume that oscillator is OFF at the time ENS is being read. Similarly, if software reads ENS high it is not safe to assume that ENS stays high during the next instruction. Once the clock is qualified, the RDY status is set high in the LFXO_STATUS register. The same uncertainties also apply to the RDY bit. However, software can wait for RDY bit to go high to detect that LFXO clock is qualified. Or it can enable the interrupt with RDYIEN in LFXO_IEN register and receive RDYIF interrupt available in LFXO_IF register. RDYIF also acts as EM2 wakeup source if RDYIEN set high. If put into FORCEON mode, the LFXO will start the qualification and once qualified it will gate off the clock but immediately start with no qualification upon receiving a request. If in ONDEMAND mode, the LFXO starts the qualification every time it is switched from off to on due to clock requests. The qualification can take up to 32k cycles. Note that only enabling RDY interrupt does not act as a clock request.

8.5.3.4 Edge Detection Interrupts

There is a possibility for software to detect rising or falling edges of the LFXO clock. The edge detection is enabled if any of POSEDGEIEN and NEGEDGEIEN is set to 1. The corresponding flags are available in POSEDGEIF and NEGEDGEIF. If none of the interrupts are enabled, the edge detection is disabled and POSEDGEIF and NEGEDGEIF hold their last value until cleared or set by software. Disabling the edge detection is only allowed on NEGEDGEIF. Both flags act as EM2 wakeup sources if the corresponding IEN is set high.

8.5.3.5 Clock Failure

In case the oscillator or crystal stops or does not output clock when expected, a failure interrupt can be raised. The failure occurs if fewer than 3 LFXO clock positive edges happen during one 1ms. The failure detection is enabled by setting FAILDETEN to 1 in LFXO_CTRL register. This bit acts as a clock requester. Once enabled, failure detection status can be checked by reading FAILIF in LFXO_IF register. If FAILIEN is set high, failure will generate both interrupt and EM2 wakeup. Failure detection is also implemented as EM4 wakeup source. To wakeup from EM4 on LFXO failure detection, set FAILDETEM4WUEN high in LFXO_CTRL.

8.5.3.6 Automatic Gain Control

AGC and HIGHAMPL in LFXO_CFG are settings applied to the LFXO oscillator. Both settings provide higher crystal oscillation amplitude. This will improve duty cycle in the output clock and give lower sensitivity to noise, but at the cost of higher current consumption. The AGC bit is used to enable the Automatic Gain Control module that adjusts the amplitude of the oscillations. It is enabled by default. When disabled, the LFXO will run at the start-up current and the crystal will oscillate rail-to-rail or limited by the start-up current. The HIGHAMPL bit will have no effect when AGC is disabled. When AGC is enabled setting the HIGHAMPL bit will give about 70% higher crystal oscillation amplitude.

8.5.3.7 Force Off

It is not allowed to write to LFXO_CFG unless LFXO is in FORCEOFF mode. If this guideline is violated, the write access is blocked and a bus fault is generated. Writing to CFG registers has no effect in DIGEXTCLK mode. Note: when putting the oscillators to FORCEOFF mode, wait for ENS status to go low for the oscillator to completely shut off. Once the oscillator is forced off, it is safe to write to the LFXO_CFG register.

8.5.3.8 Register Synchronization

While the CFG registers are static LFXO configuration, LFXO_CAL register has GAIN and CAPTUNE bit fields which can be written to while the oscillator is running. This is used to calibrate the LFXO clock. These registers are allowed to be written only if CALBSY in LFXO_SYNCBUSY register is low. If this guideline is violated, the write access is blocked and a bus fault is generated. CALBSY is guaranteed to be low in FORCEOFF mode. When exiting FORCEOFF mode, CALBSY will go high and stay high until the initial internal synchronization is done. CALBSY is also guaranteed to be low in DIGEXTCLK mode since writing to CAL register has no effect in DIGEXTCLK mode. CAPTUNE is allowed to be incremented or decremented by one LSB when not in FORCEOFF mode. Note that CAPTUNE tunes the internal capacitors connected to LFXTAL_I and LFXTAL_O pads (see Register map for more details). By programming GAIN bit field it is possible to optimize start-up time and power consumption for a given crystal. Internal capacitances are not provided on all chips (see the device data sheet for more details).

8.5.3.9 Register Lock

See the LFXO_LOCK register on how to lock certain registers. Registers LFXO_CTRL, LFXO_CFG, and LFXO_CAL are lockable. The LOCK bit in LFXO_STATUS register is available to check whether the registers are locked. If locked, all updates to these registers are blocked and bus faults are issued.

8.5.3.10 Reset Behavior

Upon reset, the LFXO is configured for the safe crystal start-up. The TIMEOUT is set to 32k cycles, The MODE is set to XTAL and the reset state is FORCEOFF. In order to minimize the start-up time and power consumption for a given crystal, it is possible to adjust the start-up gain in the oscillator by programming GAIN in LFXO_CAL. All controls are retained in EM4, except LFXO_IEN register which is reset after EM4 wakeup.

8.5.4 LFXO Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	LFXO_IPVERSION	R	LFXO IP Version
0x004	LFXO_CTRL	RW	LFXO Control Register
0x008	LFXO_CFG	RW	LFXO Configuration Register
0x010	LFXO_STATUS	RH	LFXO Status Register
0x014	LFXO_CAL	RW LFSYNC	LFXO Calibration Register
0x018	LFXO_IF	RWH INTFLAG	Interrupt Flag Register
0x01C	LFXO_IEN	RW	Interrupt Enable Register
0x020	LFXO_SYNCBUSY	RH	LFXO Sync Busy Register
0x024	LFXO_LOCK	W	Configuration Lock Register
0x1000	LFXO_IPVERSION_SET	R	LFXO IP Version
0x1004	LFXO_CTRL_SET	RW	LFXO Control Register
0x1008	LFXO_CFG_SET	RW	LFXO Configuration Register
0x1010	LFXO_STATUS_SET	RH	LFXO Status Register
0x1014	LFXO_CAL_SET	RW LFSYNC	LFXO Calibration Register
0x1018	LFXO_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x101C	LFXO_IEN_SET	RW	Interrupt Enable Register
0x1020	LFXO_SYNCBUSY_SET	RH	LFXO Sync Busy Register
0x1024	LFXO_LOCK_SET	W	Configuration Lock Register
0x2000	LFXO_IPVERSION_CLR	R	LFXO IP Version
0x2004	LFXO_CTRL_CLR	RW	LFXO Control Register
0x2008	LFXO_CFG_CLR	RW	LFXO Configuration Register
0x2010	LFXO_STATUS_CLR	RH	LFXO Status Register
0x2014	LFXO_CAL_CLR	RW LFSYNC	LFXO Calibration Register
0x2018	LFXO_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x201C	LFXO_IEN_CLR	RW	Interrupt Enable Register
0x2020	LFXO_SYNCBUSY_CLR	RH	LFXO Sync Busy Register
0x2024	LFXO_LOCK_CLR	W	Configuration Lock Register
0x3000	LFXO_IPVERSION_TGL	R	LFXO IP Version
0x3004	LFXO_CTRL_TGL	RW	LFXO Control Register
0x3008	LFXO_CFG_TGL	RW	LFXO Configuration Register
0x3010	LFXO_STATUS_TGL	RH	LFXO Status Register
0x3014	LFXO_CAL_TGL	RW LFSYNC	LFXO Calibration Register
0x3018	LFXO_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x301C	LFXO_IEN_TGL	RW	Interrupt Enable Register
0x3020	LFXO_SYNCBUSY_TGL	RH	LFXO Sync Busy Register

Offset	Name	Type	Description
0x3024	LFXO_LOCK_TGL	W	Configuration Lock Register

8.5.5 LFXO Register Description

8.5.5.1 LFXO_IPVERSION - LFXO IP Version

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP Version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

8.5.5.2 LFXO_CTRL - LFXO Control Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0			0x1	0x0
Access																											RW	RW			RW	RW
Name																											FAILDETEM4WUEN	FAILDETEN			DISONDEMAND	FORCEEN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	FAILDETEM4WUEN	0x0	RW	**LFXO Failure Detection EM4WU Enable**
	Set this bit to enable EM4 exit on the oscillator failure detection.			
4	FAILDETEN	0x0	RW	**LFXO Failure Detection Enable**
	Set this bit to enable the oscillator failure detection feature. Note that setting this bit will enable the oscillator core.			
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISONDEMAND	0x1	RW	**LFXO Disable On-demand requests**
	Set this bit to disable On-demand requests.			
0	FORCEEN	0x0	RW	**LFXO Force Enable**
	Set this bit to enable the oscillator core. The oscillator core is enabled regardless of On-demand requests.			

8.5.5.3 LFXO_CFG - LFXO Configuration Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x7				0x0				0x0	0x1
Access																							RW				RW				RW	RW
Name																							TIMEOUT				MODE				HIGHAMPL	AGC

Bit	Name	Reset	Access	Description
31:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10:8	TIMEOUT	0x7	RW	**LFXO Start-up Delay**

Configures the start-up delay for LFXO.

Value	Mode	Description
0	CYCLES2	Timeout period of 2 cycles
1	CYCLES256	Timeout period of 256 cycles
2	CYCLES1K	Timeout period of 1024 cycles
3	CYCLES2K	Timeout period of 2048 cycles
4	CYCLES4K	Timeout period of 4096 cycles
5	CYCLES8K	Timeout period of 8192 cycles
6	CYCLES16K	Timeout period of 16384 cycles
7	CYCLES32K	Timeout period of 32768 cycles

Bit	Name	Reset	Access	Description
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	MODE	0x0	RW	**LFXO Mode**

Selects the LFXO mode.

Value	Mode	Description
0	XTAL	A 32768Hz crystal should be connected to the LF crystal pads. Voltage must not exceed VDDIO.
1	BUFEXTCLK	An external sine source with minimum amplitude 100mv (zero-to-peak) and maximum amplitude 500mV (zero-to-peak) should be connected in series with LFXTAL_I pin. Minimum voltage should be larger than ground and maximum voltage smaller than VDDIO. The sine source does not need to be ac coupled externally as it is ac couples inside LFXO. LFXTAL_O is free to be used as a general purpose GPIO.
2	DIGEXTCLK	An external 32KHz CMOS clock should be provided on LFXTAL_I. LFXTAL_O is free to be used as a general purpose GPIO.

Bit	Name	Reset	Access	Description
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	HIGHAMPL	0x0	RW	**LFXO High Amplitude Enable**
	Set this bit to enable high XTAL oscillation amplitude.			
0	AGC	0x1	RW	**LFXO AGC Enable**
	Set this bit to enable automatic gain control which limits XTAL oscillation amplitude.			

8.5.5.4 LFXO_STATUS - LFXO Status Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0															0x0																0x0
Access	R															R																R
Name	LOCK															ENS																RDY

Bit	Name	Reset	Access	Description
31	LOCK	0x0	R	**LFXO Locked Status**
	If set, all LFXO lockable registers are locked.			

Value	Mode	Description
0	UNLOCKED	LFXO lockable registers are not locked
1	LOCKED	LFXO lockable registers are locked

30:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	ENS	0x0	R	**LFXO Enable Status**
	LFXO is enabled.			
15:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	RDY	0x0	R	**LFXO Ready Status**
	LFXO is enabled and start-up time has exceeded.			

8.5.5.5 LFXO_CAL - LFXO Calibration Register

Offset	Bit Position																															
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x1			0x0						
Access																							RW			RW						
Name																							GAIN			CAPTUNE						

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	GAIN	0x1	RW	**LFXO Startup Gain** The optimal value depends on the chosen crystal.
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:0	CAPTUNE	0x0	RW	**Internal Capacitance Tuning** Program internal load capacitance connected between X_N pin and ground and X_P pin and ground. The bus affects tuning capacitances on both pins symmetrically. CAPTUNE value must not exceed 0x4F. When updating CAPTUNE, its value must only be incremented or decremented by 1 which provides a tuning step of 0.25pF. The maximum value is estimated to be 20pF. Please refer to the device Datasheet for more information.

8.5.5.6 LFXO_IF - Interrupt Flag Register

Offset	Bit Position																															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													RW	RW	RW	RW
Name																													FAIL	NEGEDGE	POSEDGE	RDY

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	FAIL	0x0	RW	**LFXO Failure Interrupt Flag**
	Set when LFXO failure is detected. Write 1 to clear the interrupt flag.			
2	NEGEDGE	0x0	RW	**Falling Edge Interrupt Flag**
	Triggers on every negative edge of the LFXO clock.			
1	POSEDGE	0x0	RW	**Rising Edge Interrupt Flag**
	Triggers on every positive edge of the LFXO clock.			
0	RDY	0x0	RW	**LFXO Ready Interrupt Flag**
	Set when LFXO is ready (start-up time exceeded). Write 1 to clear the interrupt flag.			

8.5.5.7 LFXO_IEN - Interrupt Enable Register

Offset	Bit Position																															
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													RW	RW	RW	RW
Name																													FAIL	NEGEDGE	POSEDGE	RDY

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	FAIL	0x0	RW	**LFXO Failure Interrupt Enable**
	Write 1 to enable FAILIF			
2	NEGEDGE	0x0	RW	**Falling Edge Interrupt Enable**
	Write 1 to enable NEGEDGEIF.			
1	POSEDGE	0x0	RW	**Rising Edge Interrupt Enable**
	Write 1 to enable POSEDGEIF.			
0	RDY	0x0	RW	**LFXO Ready Interrupt Enable**
	Write 1 to enable RDYIF.			

8.5.5.8 LFXO_SYNCBUSY - LFXO Sync Busy Register

Offset	Bit Position																															
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																R
Name																																CAL

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	CAL	0x0	R	**LFXO Synchronization status**
	This bit is set when there is an ongoing synchronization of CAL register bitfields. Do not write to CAL register while this bit is set.			

8.5.5.9 LFXO_LOCK - Configuration Lock Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x1A20								
Access																								W								
Name																								LOCKKEY								

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x1A20	W	**Lock Key**

Write any other value than UNLOCK to lock CTRL, CFG and CAL registers. Write UNLOCK value to unlock the registers.

Value	Mode	Description
6688	UNLOCK	Unlock LFXO lockable registers

8.6 LFRCO - Low-Frequency RC Oscillator

8.6.1 Introduction

The LFRCO is an integrated low-frequency (32.768 kHz) RC oscillator that may be used as a timing reference in low energy modes when crystal accuracy is not required.

8.6.2 Features

- 32.768 kHz oscillator
- High Accuracy
- Available in all energy modes, except EM3
- On-demand
- EM2 wakeup interrupt for oscillator ready
- EM2 wakeup interrupts for rising and falling edges of the clock
- Lockable registers
- Trim bit synchronization

8.6.3 Functional Description

8.6.3.1 Start-up

The LFRCO has a fast start-up time (refer to the data sheet electrical specifications for the exact start-up time). When the oscillator has started up and is ready to use, the RDY status bit will go high and the RDY interrupt will be triggered. After start-up, it may take two clock cycles for the clock to propagate through the CMU to the peripherals.

8.6.3.2 On-Demand Clocking

Software may forceably enable the LFRCO by setting the LFRCO_CTRL.FORCEEN bit field. However, by default, the LFRCO is configured to be enabled only when required by hardware, and to shut down when no hardware request is present (i.e. LFRCO_CTRL.DISONDEMAND=0 and LFRCO_CTRL.FORCEEN=0). This is known as on-demand clocking and allows the oscillator to be controlled without any software intervention.

8.6.3.3 Calibration

The LFRCO is trimmed in production and the trim values are automatically written to the FREQTRIM field in the LFRCO_CAL register, before user code execution. Normally, software does not need to modify the to the LFRCO_CAL register. However, it is possible for software to re-calibrate the LFRCO by modifying the FREQTRIM value. This might be desired, for example if re-calibration is needed at a specific temperature, or there is a desire to use different trim values at different temperatures.

It is possible to recalibrate the LFRCO by modifying the FREQTRIM value in the LFRCO_CAL register. Software may modify the LFRCO_CAL register while it is running. However, the LFRCO_CAL has hardware synchronization, and should only be written after checking that SYNCBUSY_CALBSY is not set.

8.6.3.4 Interrupts

LFRCO has three interrupts, RDYIF, POSEDGEIF and NEGEDGEIF. Each will trigger an EM2 wakeup if the corresponding IEN is set.

RDYIF is triggered after start-up, when the LFRCO startup sequence is complete and the oscillator is ready to use.

POSEDGEIF and NEGEDGEIF are triggered by the rising and falling edge of LFRCO respectively. These flags will only get set if either of the interrupts are enabled (with POSEDGEIEN or NEGEDGEIEN), as the interrupt enable acts as a clock requester and keeps the oscillator running.

8.6.4 LFRCO Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	LFRCO_IPVERSION	R	IP Version
0x008	LFRCO_STATUS	RH	Status Register
0x00C	LFRCO_CAL	RW	Calibration Register
0x014	LFRCO_IF	RWH INTFLAG	Interrupt Flag Register
0x018	LFRCO_IEN	RW	Interrupt Enable Register
0x01C	LFRCO_SYNCBUSY	RH	Synchronization Busy Register
0x020	LFRCO_LOCK	W	Configuration Lock Register
0x1000	LFRCO_IPVERSION_SET	R	IP Version
0x1008	LFRCO_STATUS_SET	RH	Status Register
0x100C	LFRCO_CAL_SET	RW	Calibration Register
0x1014	LFRCO_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1018	LFRCO_IEN_SET	RW	Interrupt Enable Register
0x101C	LFRCO_SYNCBUSY_SET	RH	Synchronization Busy Register
0x1020	LFRCO_LOCK_SET	W	Configuration Lock Register
0x2000	LFRCO_IPVERSION_CLR	R	IP Version
0x2008	LFRCO_STATUS_CLR	RH	Status Register
0x200C	LFRCO_CAL_CLR	RW	Calibration Register
0x2014	LFRCO_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2018	LFRCO_IEN_CLR	RW	Interrupt Enable Register
0x201C	LFRCO_SYNCBUSY_CLR	RH	Synchronization Busy Register
0x2020	LFRCO_LOCK_CLR	W	Configuration Lock Register
0x3000	LFRCO_IPVERSION_TGL	R	IP Version
0x3008	LFRCO_STATUS_TGL	RH	Status Register
0x300C	LFRCO_CAL_TGL	RW	Calibration Register
0x3014	LFRCO_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3018	LFRCO_IEN_TGL	RW	Interrupt Enable Register
0x301C	LFRCO_SYNCBUSY_TGL	RH	Synchronization Busy Register
0x3020	LFRCO_LOCK_TGL	W	Configuration Lock Register

8.6.5 LFRCO Register Description

8.6.5.1 LFRCO_IPVERSION - IP Version

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x0	R	**IP version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

8.6.5.2 LFRCO_STATUS - Status Register

Offset	Bit Position																															
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0															0x0																0x0
Access	R															R																R
Name	LOCK															ENS																RDY

Bit	Name	Reset	Access	Description
31	LOCK	0x0	R	**Lock Status**
	This bit is set when LFRCO is locked.			

Value	Mode	Description
0	UNLOCKED	Access to configuration registers not locked
1	LOCKED	Access to configuration registers locked

30:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*

Bit	Name	Reset	Access	Description
16	ENS	0x0	R	**Enabled Status**
	This bit is set when LFRCO is enabling the analog core.			
15:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	RDY	0x0	R	**Ready Status**
	This bit is set when qualification is done and LFRCO is ready.			

8.6.5.3 LFRCO_CAL - Calibration Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0xA5				
Access																												RW				
Name																												FREQTRIM				

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	FREQTRIM	0xA5	RW	**Frequency Trim**
	Trims the clock frequency of the LFRCO			

8.6.5.4 LFRCO_IF - Interrupt Flag Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														NEGEDGE	POSEDGE	RDY

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	NEGEDGE	0x0	RW	**Falling Edge Interrupt Flag**
	Triggers on every negative edge of the LFRCO clock.			
1	POSEDGE	0x0	RW	**Rising Edge Interrupt Flag**
	Triggers on every positive edge of the LFRCO clock.			
0	RDY	0x0	RW	**Ready Interrupt Flag**
	Triggers when the oscillator becomes ready			

8.6.5.5 LFRCO_IEN - Interrupt Enable Register

Offset	Bit Position																														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														NEGEDGE	POSEDGE	RDY

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	NEGEDGE	0x0	RW	**Falling Edge Interrupt Enable**
	Enables the negedge interrupt and will cause the oscillator to run. EM2 wakeup source.			
1	POSEDGE	0x0	RW	**Rising Edge Interrupt Enable**
	Enables the posedge interrupt and will cause the oscillator to run. EM2 wakeup source.			
0	RDY	0x0	RW	**Ready Interrupt Enable**
	Enables the ready interrupt. EM2 wakeup source.			

8.6.5.6 LFRCO_SYNCBUSY - Synchronization Busy Register

Offset	Bit Position																														
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																R
Name																																CAL

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	CAL	0x0	R	**CAL Busy**
	CAL register synchronization busy bit			

8.6.5.7 LFRCO_LOCK - Configuration Lock Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x2603								
Access																								W								
Name																								LOCKKEY								

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x2603	W	**Lock Key**

Writing the lock key will unlock the lfrco configuration registers (CAL, CTRL and TEST). Writing any other value will lock them.

Value	Mode	Description
0	LOCK	Lock Configuration Registers
9731	UNLOCK	Unlock Configuaration Registers

8.7 FSRCO - Fast Start RCO

8.7.1 Introduction

This is an RC oscillator which can start and stop very fast. It is a fixed frequency oscillator, with no frequency configurability and as such any user of this clock can rely on it being a specific frequency independent of the system state. This is the first oscillator used during power up and hence it minimizes dependency to other blocks.

8.7.2 Features

- 20 MHz nominal frequency
- Low energy consumption

8.7.3 Functional Description

There are no programmable registers in this module. Software can choose to use this as system clock in the CMU block. the only way to enable or disable the FSRCO is by requesting it as a clock source in the CMU clock select registers.

8.7.4 FSRCO Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	FSRCO_IPVERSION	R	IP Version
0x1000	FSRCO_IPVERSION_SET	R	IP Version
0x2000	FSRCO_IPVERSION_CLR	R	IP Version
0x3000	FSRCO_IPVERSION_TGL	R	IP Version

8.7.5 FSRCO Register Description

8.7.5.1 FSRCO_IPVERSION - IP Version

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x0	R	**IP Version**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

8.8 ULFRCO - Ultra Low Frequency RC Oscillator

8.8.1 Introduction

The ULFRCO is an ultra low power 1 kHz oscillator which is available in all energy modes. The ULFRCO is available to many low-frequency peripherals as a lower power alternative to one of the 32 kHz oscillators. This oscillator is also used for internal bias and housekeeping tasks in EM0-EM3.

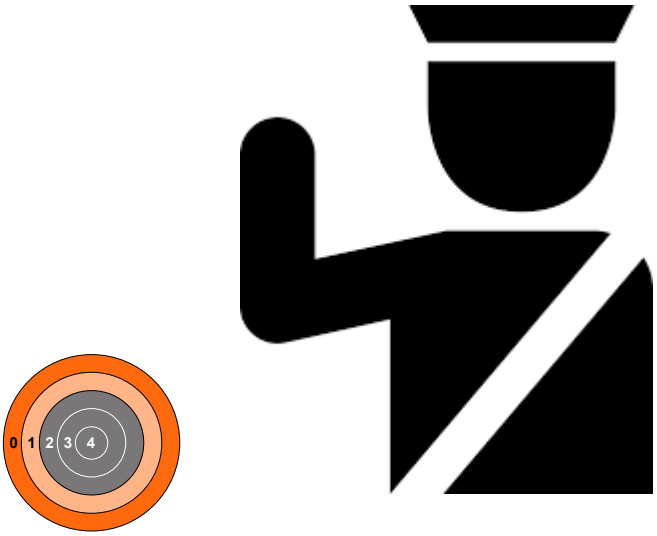
8.8.2 Features

- 1 kHz nominal frequency
- Low energy consumption

8.8.3 Functional Description

There are no user programmable registers in this module. The oscillator is always on in all energy modes except EM4. In EM4, the oscillator is available on-demand by peripheral requests.

9. SMU - Security Management Unit



Quick Facts

What?

The Security Management Unit (SMU) provides configuration and status reporting for ARM TrustZone on the EFM32PG28.

Why?

Enables a robust solution at the system level.

How?

Hardware context switching and enhanced security provided by ARM TrustZone. Extension of the ARM MPU to control peripheral access.

9.1 Introduction

The Security Management Unit is used to configure and extend TrustZone bus level security provided by the Cortex®-M33. In addition it increases the effective MPU regions by providing MPU control over peripheral access.

9.2 Features

- Per peripheral privileged and secure attributes
- Per manager privileged and secure attributes
- Separate interrupt flags for privileged, secure, or instruction access exceptions.
- Separate interrupt flag for secure manager access exceptions
- Secure and Privileged exception IRQs
- Configurable secure, non-secure, and non-secure-callable memory regions.

9.3 Functional Description

9.3.1 Bus Level Security

Bus level security is the ability to control the flow of information on the device. The components of bus level security are the Cortex®-M33, the Bus Manager Protect Unit (BMPU), and the Peripheral Protection Unit (PPU) as highlighted in Figure 9.1 Bus Level Security Implementation on page 242. The SMU controls and configures all the components used in bus level security.

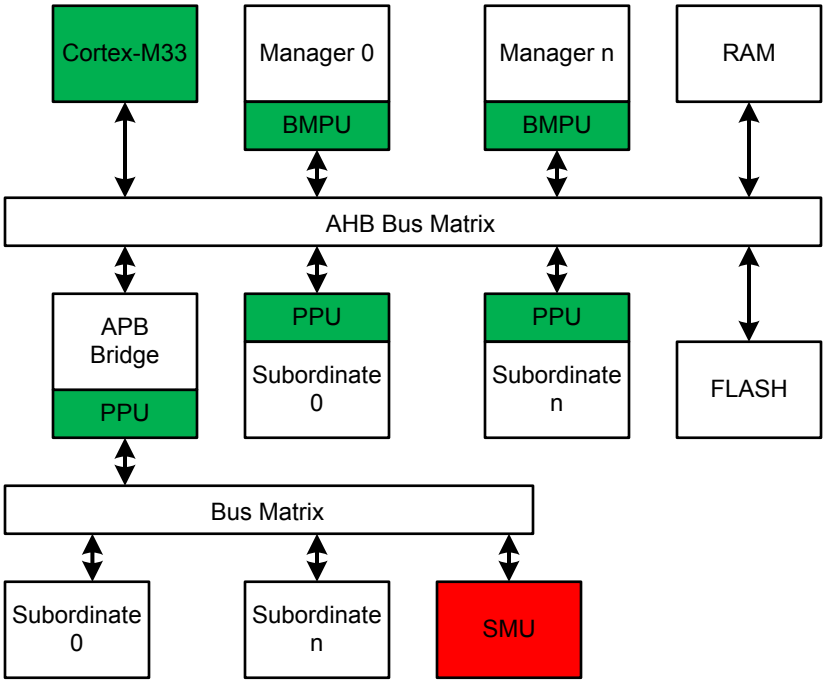


Figure 9.1. Bus Level Security Implementation

The BMPU is responsible for preventing managers (CPU, DMA, Etc..) from accessing secure addresses without authorization. For example, if a DMA configured as non-secure tries to access memory that is marked secure the BMPU will prevent access and set the corresponding interrupt flag. The BMPU prevents access of secure addresses by non-secure managers. The Cortex®-M33 has BMPU functionality built into the TrustZone implementation.

The PPU is primarily responsible for blocking access to privileged peripherals from unprivileged managers. In addition, it also ensures that secure and non-secure peripherals are only accessible at the appropriate secure or non-secure addresses as described in 9.3.6 Configuring Peripherals.

Since FLASH and RAM have no PPU, bus managers of any privilege state may access those resources. The Cortex®-M33 has an MPU which prevents execution of privileged memory when the CPU is in an unprivileged state. For more information on the MPU refer to the ARM Cortex®-M33 documentation.

9.3.2 Privileged Access Control

The Cortex®-M33 and all other managers can be in either the privileged or unprivileged state. All bus access to peripherals are tested for privilege level by the PPU and resolved as shown in Table 9.1 Privileged Access Table on page 243.

If an exception is detected on a write, the write will be ignored and the appropriate interrupt flag set. If an exception is detected on a read 0x0 will be returned and the appropriate interrupt flag set.

Table 9.1. Privileged Access Table

Manager Attribute	Peripheral Attribute	Result
privileged	privileged	Success
privileged	unprivileged	Success
unprivileged	privileged	Exception
unprivileged	unprivileged	Success

9.3.3 Secure Access Control

The Cortex®-M33 and all other managers can be in either the secure or non-secure state. All bus accesses are tested for security status by the BMPUs and PPUs and resolve as shown in Table 9.2 Secure Access Table on page 243 Secure access is computed using the secure attribute of the manager and the address region being accessed. If a peripheral is being accessed, the secure attribute of the peripheral is also used. For more information on the relationship between the address regions and peripheral security attributes please see 9.3.6 Configuring Peripherals

If an exception is detected on a write the write will be ignored and the appropriate interrupt flag set. If an exception is detected on a read 0x0 will be returned and the appropriate interrupt flag set.

Table 9.2. Secure Access Table

Manager Attribute	Address Attribute	Peripheral Attribute	Result
secure	secure	N/A	Success
secure	secure	secure	Success
secure	secure	non-secure	Exception
secure	non-secure	N/A	Exception
secure	non-secure	secure	Exception
secure	non-secure	non-secure	Success
non-secure	secure	N/A	Exception
non-secure	secure	secure	Exception
non-secure	secure	non-secure	Exception
non-secure	non-secure	N/A	Success
non-secure	non-secure	secure	Exception
non-secure	non-secure	non-secure	Success

9.3.4 ARM TrustZone

ARM TrustZone is used to control what addresses are accessible by the CPU at any given time. There are two security states: secure and non-secure. In addition the MPU provides two privilege levels: privileged and unprivileged. This results in 4 possible states: secure-privileged, non-secure-privileged, secure-unprivileged and non-secure-unprivileged.

Non-secure code may not directly call secure code. To call secure code, non-secure code must first call a shim located in specially marked non-secure-callable memory. Unprivileged code may invoke privileged code and change the processor state to privileged by either issuing an SVC instruction or taking an interrupt. The processor is returned to unprivileged state when software manually reconfigures the security state or exits an interrupt.

For more information on secure/non-secure and privileged/unprivileged state transitions see the ARM Cortex®-M33 documentation.

There are two primary use cases for TrustZone and the MPU. The first is simply partitioning a monolithic application in to the 4 states to protect some pieces of the system from bugs or attacks on others. The second is to use a RTOS to isolate several tasks from each other. In this case the RTOS itself normally consumes the privileged states with all other code running in the unprivileged states. Whenever a task switch occurs the RTOS can reconfigure the device so the new task has access to only the components it requires, protecting other tasks from interference.

In both use cases the TrustZone and MPU feature of the Cortex®-M33 both secures and accelerates mode transitions while the SMU provides the ability to configure the security and privilege attributes of peripherals and memory.

The core is in secure-privileged state after a reset.

9.3.5 Configuring Managers

The SMU provides the ability to configure the current secure and privileged attribute of all bus managers except for the CPU which is controlled as described in 9.3.4 ARM TrustZone.

To configure the privileged attribute of a manager set the appropriate bit in SMU_BMPUPATDn. To configure the secure attribute of a manager set the appropriate bit in SMU_BMPUPSATDn.

9.3.6 Configuring Peripherals

The SMU provides the ability to configure the current secure and privileged state of all peripherals. To configure the privileged attribute of a peripheral set the appropriate bit in SMU_PPUPATDn.

Each peripheral is accessible at one of two addresses: A secure address and an non-secure address. Which address is valid depends on the security attribute of the peripheral configured in the SMU. When configured as secure a peripheral may only be accessed at its secure address and when configured as non-secure the peripheral may only be accessed at its non-secure address. This forces code to be aware of the security attribute of the peripheral being accessed, preventing secure code from accessing a non-secure peripheral unintentionally.

The device memory map contains two regions of fixed length and fixed security attribute to facilitate the secure access of peripherals. There is one secure (0x40000000) and one non-secure (0x50000000) region for peripherals. Each peripheral memory region can be configured independently.

To configure the security attribute of a peripheral set the appropriate bit in SMU_PPUSATDn.

9.3.7 Configuring Memory

The SMU provides the ability to configure the security attribute of memory. There are 13 configurable regions in total. There are three regions in FLASH (0 - 2) and three in RAM (4-6) which have pre-determined secure attributes and user selectable sizes. Regions 3 and 11 cover the flash info page and ARM EPPB space respectively and have a fixed size. These regions can be configured as secure or non-secure by setting ESAUR3NS in SMU_ESAURTYPES0 and ESAUR11NS in SMU_ESAURTYPES1 respectively.

The size of the FLASH and RAM regions are controlled by the SMU_ESAUMRBRxy registers as shown in Table 9.3 Memory Configuration Regions on page 245. Region sizes are adjusted in 4 kB increments with the lower 12 bits of SMU_ESAUMRBRxy ignored. The non-secure-callable regions may be set to size 0 but secure and non-secure regions must be at least 4 kB.

Table 9.3. Memory Configuration Regions

Region	Memory	Attributes	Start	End
0	FLASH	secure	0x00000000	SMU_ESAURMBR01-1
1	FLASH	non-secure-callable	SMU_ESAURMBR01	SMU_ESAURMBR12-1
2	FLASH	non-secure	SMU_ESAURMBR12	0x0FDFFFFF
3	FLASH (info page)	secure or non-secure	0x0FE00000	0x0FFFFFFF
4	RAM	secure	0x20000000	SMU_ESAURMBR45-1
5	RAM	non-secure-callable	SMU_ESAURMBR45	SMU_ESAURMBR56-1
6	RAM	non-secure	SMU_ESAURMBR56	0x2FFFFFFF
7	Peripherals	secure	0x40000000	0x4FFFFFFF
8	Peripherals	non-secure	0x50000000	0x5FFFFFFF
9	RESERVED	secure	0xA0000000	0xAFFFFFFF
10	RESERVED	non-secure	0xB0000000	0xBFFFFFFF
11	EPPB	secure or non-secure	0xE0044000	0xE00FDFFF
12	Cortex®-M33 Processor ROM table	exempt	0xE00FE000	0xE00FEFFF

9.3.8 Cortex®-M33 Integration

In addition to the SMU based access controls the Cortex®-M33 has additional security features for controlling both secure and privileged access.

The Security Attribution Unit (SAU) provides that ability to setup secure memory regions in addition to those configured by the SMU. To disable the SAU and rely entirely on the SMU for security management clear ENABLE and set ALLNS in the SAU CTRL register. To enable a combination of SMU and SAU control set ENABLE in the SAU CTRL register. If both ENABLE and ALLNS are cleared all Cortex®-M33 will treat all transactions as secure.

When both SAU and SMU are in use, a memory address is considered secure if either the SAU or SMU have it configured as secure. When enabled the SAU applies ONLY to access by the Cortex®-M33 and does not effect any other managers. For more information on the SAU refer to ARM documentation.

Note: It is highly recommended that systems avoid using the SAU unless necessary. Since the SAU does not affect any managers outside the Cortex®-M33, extreme care must be taken to ensure the SAU regions can not be trivially by bypassed through use of another manager such as the DMA.

In addition to the Cortex®-M33 MPU provides the ability to control which regions of FLASH and RAM are marked as privileged and prevent execution of privileged code by a CPU in unprivileged state. For more information on the configuration and use of the MPU refer to ARM documentation.

9.3.9 Exception Handling

When a BMPU detects a non-secure manager attempting to access a secure address, the BMPUSECIF in SMU_IF is set and the ID of the Manager block is written to SMU_BMPUFS. If BMPUSECIEN is set and the SMU's Secure IRQ enabled, the CPU will be interrupted.

When a PPU detects an access to a secure peripheral at its non-secure address or an access to a non-secure peripheral at its secure address, PPUSECIF in SMU_IF is set and the ID of the peripheral being accessed is written to SMU_PPUFS. If PPUSECIEN is set and the SMU's Secure IRQ enabled, the CPU will be interrupted.

If a PPU detects an attempt to fetch an instruction from a peripheral, PPUINSTIF in SMU_IF will be set and the ID of the peripheral being accessed is written to SMU_PPUFS. If PPUINSTIEN is set and the SMU's Privileged IRQ enabled, the CPU will be interrupted.

If a PPU detects an attempt to access a privileged peripheral by an unprivileged manager, PPUPRIVIF in SMU_IF will be set and the ID of the peripheral being accessed is written to SMU_PPUFS. If PPUPRIVIEN is set and the SMU's Privileged IRQ enabled, the CPU will be interrupted.

When any IRQ is trigged the Cortex®-M33 is automatically placed in the privileged state. The security state is determined by configuration inside the Cortex®-M33. Refer to ARM's documentation for more details.

If the SMU is configured in an inconsistent way, the SMUPRGERR flag in SMU_STATUS will be set. One example of an invalid configuration is setting SMU_ESAUMRBR01 to a value larger than SMU_ESAUMRBR23. SMUPRGERR should be checked after the SMU is configured.

9.3.10 SMU Lock

The SMU registers can be locked to prevent unintended modifications. SMULOCK in SMU_STATUS indicates if the SMU is currently locked. To unlock the SMU write 0xACCE55 to the SMU_LOCK register. To lock write any other value to SMU_LOCK.

In addition to locking the SMU registers the SMU can prevent access to the Cortex®-M33 ASU, MPU, SMPU, VTOR and VTAIRCR registers. To lock access to one or more of these blocks set the corresponding bit in SMU_M33CTRL.

9.4 SMU Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	SMU_IPVERSION	R	IP Version
0x004	SMU_STATUS	RH	Status Register
0x008	SMU_LOCK	W	Lock Register
0x00C	SMU_IF	RWH INTFLAG	Interrupt Flag Register
0x010	SMU_IEN	RW	Interrupt Enable Register
0x020	SMU_M33CTRL	RW	M33 Control Settings
0x040	SMU_PPUPATD0	RW	Privileged Access
0x044	SMU_PPUPATD1	RW	Privileged Access
0x060	SMU_PPUSATD0	RW	Secure Access
0x064	SMU_PPUSATD1	RW	Secure Access
0x140	SMU_PPUFS	RH	Fault Status
0x150	SMU_BMPUPATD0	RW	Privileged Attribute
0x170	SMU_BMPUSATD0	RW	Secure Attribute
0x250	SMU_BMPUFS	RH	Fault Status
0x254	SMU_BMPUFSADDR	RH	Fault Status Address
0x260	SMU_ESAURTYPES0	RW	Region Types 0
0x264	SMU_ESAURTYPES1	RW	Region Types 1
0x270	SMU_ESAUMRB01	RW	Movable Region Boundary
0x274	SMU_ESAUMRB12	RW	Movable Region Boundary
0x280	SMU_ESAUMRB45	RW	Movable Region Boundary
0x284	SMU_ESAUMRB56	RW	Movable Region Boundary
0x1000	SMU_IPVERSION_SET	R	IP Version
0x1004	SMU_STATUS_SET	RH	Status Register
0x1008	SMU_LOCK_SET	W	Lock Register
0x100C	SMU_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1010	SMU_IEN_SET	RW	Interrupt Enable Register
0x1020	SMU_M33CTRL_SET	RW	M33 Control Settings
0x1040	SMU_PPUPATD0_SET	RW	Privileged Access
0x1044	SMU_PPUPATD1_SET	RW	Privileged Access
0x1060	SMU_PPUSATD0_SET	RW	Secure Access
0x1064	SMU_PPUSATD1_SET	RW	Secure Access
0x1140	SMU_PPUFS_SET	RH	Fault Status
0x1150	SMU_BMPUPATD0_SET	RW	Privileged Attribute
0x1170	SMU_BMPUSATD0_SET	RW	Secure Attribute
0x1250	SMU_BMPUFS_SET	RH	Fault Status

Offset	Name	Type	Description
0x1254	SMU_BMPUFSADDR_SET	RH	Fault Status Address
0x1260	SMU_ESAURTYPES0_SET	RW	Region Types 0
0x1264	SMU_ESAURTYPES1_SET	RW	Region Types 1
0x1270	SMU_ESAUMRB01_SET	RW	Movable Region Boundary
0x1274	SMU_ESAUMRB12_SET	RW	Movable Region Boundary
0x1280	SMU_ESAUMRB45_SET	RW	Movable Region Boundary
0x1284	SMU_ESAUMRB56_SET	RW	Movable Region Boundary
0x2000	SMU_IPVERSION_CLR	R	IP Version
0x2004	SMU_STATUS_CLR	RH	Status Register
0x2008	SMU_LOCK_CLR	W	Lock Register
0x200C	SMU_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2010	SMU_IEN_CLR	RW	Interrupt Enable Register
0x2020	SMU_M33CTRL_CLR	RW	M33 Control Settings
0x2040	SMU_PPUPATD0_CLR	RW	Privileged Access
0x2044	SMU_PPUPATD1_CLR	RW	Privileged Access
0x2060	SMU_PPUSATD0_CLR	RW	Secure Access
0x2064	SMU_PPUSATD1_CLR	RW	Secure Access
0x2140	SMU_PPUFS_CLR	RH	Fault Status
0x2150	SMU_BMPUPATD0_CLR	RW	Privileged Attribute
0x2170	SMU_BMPUSATD0_CLR	RW	Secure Attribute
0x2250	SMU_BMPUFS_CLR	RH	Fault Status
0x2254	SMU_BMPUFSADDR_CLR	RH	Fault Status Address
0x2260	SMU_ESAURTYPES0_CLR	RW	Region Types 0
0x2264	SMU_ESAURTYPES1_CLR	RW	Region Types 1
0x2270	SMU_ESAUMRB01_CLR	RW	Movable Region Boundary
0x2274	SMU_ESAUMRB12_CLR	RW	Movable Region Boundary
0x2280	SMU_ESAUMRB45_CLR	RW	Movable Region Boundary
0x2284	SMU_ESAUMRB56_CLR	RW	Movable Region Boundary
0x3000	SMU_IPVERSION_TGL	R	IP Version
0x3004	SMU_STATUS_TGL	RH	Status Register
0x3008	SMU_LOCK_TGL	W	Lock Register
0x300C	SMU_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3010	SMU_IEN_TGL	RW	Interrupt Enable Register
0x3020	SMU_M33CTRL_TGL	RW	M33 Control Settings
0x3040	SMU_PPUPATD0_TGL	RW	Privileged Access
0x3044	SMU_PPUPATD1_TGL	RW	Privileged Access
0x3060	SMU_PPUSATD0_TGL	RW	Secure Access

Offset	Name	Type	Description
0x3064	SMU_PPUSATD1_TGL	RW	Secure Access
0x3140	SMU_PPUFS_TGL	RH	Fault Status
0x3150	SMU_BMPUPATD0_TGL	RW	Privileged Attribute
0x3170	SMU_BMPUSATD0_TGL	RW	Secure Attribute
0x3250	SMU_BMPUFS_TGL	RH	Fault Status
0x3254	SMU_BMPUFSADDR_TGL	RH	Fault Status Address
0x3260	SMU_ESAURTYPES0_TGL	RW	Region Types 0
0x3264	SMU_ESAURTYPES1_TGL	RW	Region Types 1
0x3270	SMU_ESAUMRB01_TGL	RW	Movable Region Boundary
0x3274	SMU_ESAUMRB12_TGL	RW	Movable Region Boundary
0x3280	SMU_ESAUMRB45_TGL	RW	Movable Region Boundary
0x3284	SMU_ESAUMRB56_TGL	RW	Movable Region Boundary

9.5 SMU Register Description

9.5.1 SMU_IPVERSION - IP Version

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x6																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x6	R	**IP Version**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

9.5.2 SMU_STATUS - Status Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	R
Name																															SMUPRGERR	SMULOCK

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	SMUPRGERR	0x0	R	**SMU Programming Error**
	Indicates if SMU Registers were programmed incorrectly.			
0	SMULOCK	0x0	R	**SMU Lock**
	Indicates if SMU Registers are locked.			

Value	Mode	Description
0	UNLOCKED	
1	LOCKED	

9.5.3 SMU_LOCK - Lock Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																							
Access									W																							
Name									SMULOCKKEY																							

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:0	SMULOCKKEY	0x0	W	
	Write anything but UNLOCK to lock registers.			

Value	Mode	Description
11325013	UNLOCK	Unlocks Registers

9.5.4 SMU_IF - Interrupt Flag Register

Offset	Bit Position																														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0	0x0														0x0		0x0
Access															RW	RW														RW		RW
Name															BMPUSEC	PPUSEC														PPUINST		PPUPRIV

Bit	Name	Reset	Access	Description
31:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	BMPUSEC	0x0	RW	**BMPU Security Interrupt Flag**
	Triggered when a security fault occurs in the Bus Manager Protection Unit			
16	PPUSEC	0x0	RW	**PPU Security Interrupt Flag**
	Triggered when a security fault occurs in the Peripheral Protection Unit			
15:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	PPUINST	0x0	RW	**PPU Instruction Interrupt Flag**
	Triggered when a instruction fault occurs in the Peripheral Protection Unit			
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	PPUPRIV	0x0	RW	**PPU Privilege Interrupt Flag**
	Triggered when a privilege fault occurs in the Peripheral Protection Unit			

9.5.5 SMU_IEN - Interrupt Enable Register

Offset	Bit Position																														
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0	0x0														0x0		0x0
Access															RW	RW														RW		RW
Name															BMPUSEC	PPUSEC														PPUINST		PPUPRIV

Bit	Name	Reset	Access	Description
31:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	BMPUSEC	0x0	RW	**BMPU Security Interrupt Enable**
	Set to enable the BMPUSECIF Interrupt			
16	PPUSEC	0x0	RW	**PPU Security Interrupt Enable**
	Set to enable the PPUSECIF Interrupt			
15:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	PPUINST	0x0	RW	**PPU Instruction Interrupt Enable**
	Set to enable the PPUINSTIF Interrupt			
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	PPUPRIV	0x0	RW	**PPU Privilege Interrupt Enable**
	Set to enable the PPUPRIVIF Interrupt			

9.5.6 SMU_M33CTRL - M33 Control Settings

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												LOCKSAU	LOCKNSMPU	LOCKSMPU	LOCKNSVTOR	LOCKSVTAIRCR

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	LOCKSAU	0x0	RW	**New BitField**
	Set to 1 lock security attribution unit			
3	LOCKNSMPU	0x0	RW	**New BitField**
	Set to 1 lock non-secure MPU configuration			
2	LOCKSMPU	0x0	RW	**New BitField**
	Set to 1 lock secure MPU configuration			
1	LOCKNSVTOR	0x0	RW	**New BitField**
	Set to 1 lock non-secure VTOR			
0	LOCKSVTAIRCR	0x0	RW	**New BitField**
	Set to 1 lock secure VTAIRCR			

9.5.7 SMU_PPUPATD0 - Privileged Access

Offset	Bit Position																															
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1		0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	
Access	RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Name	EUSART2	EUSART1	HOSTMAILBOX	DCDC	GPCRC	BURAM	SYSCFG	SYSCFGCFGNS		I2C1	BURTC	USART0	TIMER4	TIMER3	TIMER2	TIMER1	TIMER0	LDMAXBAR	LDMA	GPIO	PRS	ICACHE0	MSC	ULFRCO	LFRCO	LFXO	DPLL0	FSRCO	HFRCO0	CMU	EMU	

Bit	Name	Reset	Access	Description
31	EUSART2	0x1	RW	**EUSART2 Privileged Access**
	EUSART2 Privileged Access			
30	EUSART1	0x1	RW	**EUSART1 Privileged Access**
	EUSART1 Privileged Access			
29	HOSTMAILBOX	0x1	RW	**HOSTMAILBOX Privileged Access**
	HOSTMAILBOX Privileged Access			
28	DCDC	0x1	RW	**DCDC Privileged Access**
	DCDC Privileged Access			
27	GPCRC	0x1	RW	**GPCRC Privileged Access**
	GPCRC Privileged Access			
26	BURAM	0x1	RW	**BURAM Privileged Access**
	BURAM Privileged Access			
25	SYSCFG	0x1	RW	**SYSCFG Privileged Access**
	SYSCFG Privileged Access			
24	SYSCFGCFGNS	0x1	RW	**SYSCFGCFGNS Privileged Access**
	SYSCFGCFGNS Privileged Access			
23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22	I2C1	0x1	RW	**I2C1 Privileged Access**
	I2C1 Privileged Access			
21	BURTC	0x1	RW	**BURTC Privileged Access**
	BURTC Privileged Access			
20	USART0	0x1	RW	**USART0 Privileged Access**
	USART0 Privileged Access			
19	TIMER4	0x1	RW	**TIMER4 Privileged Access**
	TIMER4 Privileged Access			
18	TIMER3	0x1	RW	**TIMER3 Privileged Access**

Bit	Name	Reset	Access	Description
	TIMER3 Privileged Access			
17	TIMER2	0x1	RW	**TIMER2 Privileged Access**
	TIMER2 Privileged Access			
16	TIMER1	0x1	RW	**TIMER1 Privileged Access**
	TIMER1 Privileged Access			
15	TIMER0	0x1	RW	**TIMER0 Privileged Access**
	TIMER0 Privileged Access			
14	LDMAXBAR	0x1	RW	**LDMAXBAR Privileged Access**
	LDMAXBAR Privileged Access			
13	LDMA	0x1	RW	**LDMA Privileged Access**
	LDMA Privileged Access			
12	GPIO	0x1	RW	**GPIO Privileged Access**
	GPIO Privileged Access			
11	PRS	0x1	RW	**PRS Privileged Access**
	PRS0 Privileged Access			
10	ICACHE0	0x1	RW	**ICACHE0 Privileged Access**
	ICACHE0 Privileged Access			
9	MSC	0x1	RW	**MSC Privileged Access**
	IMEM Privileged Access			
8	ULFRCO	0x1	RW	**ULFRCO Privileged Access**
	ULFRCO Privileged Access			
7	LFRCO	0x1	RW	**LFRCO Privileged Access**
	LFRCO Privileged Access			
6	LFXO	0x1	RW	**LFXO Privileged Access**
	LFXO Privileged Access			
5	DPLL0	0x1	RW	**DPLL0 Privileged Access**
	DPLL0 Privileged Access			
4	FSRCO	0x1	RW	**FSRCO Privileged Access**
	FSRCO Privileged Access			
3	HFRCO0	0x1	RW	**HFRCO0 Privileged Access**
	HFRCO0 Privileged Access			
2	CMU	0x1	RW	**CMU Privileged Access**
	CMU Privileged Access			
1	EMU	0x1	RW	**EMU Privileged Access**
	EMU Privileged Access			
0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

9.5.8 SMU_PPUPATD1 - Privileged Access

Offset	Bit Position																													
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset								0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1				0x1	0x1	0x1	0x1
Access								RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW				RW	RW	RW	RW
Name								MVP	SEMAILBOX	EUSART0	WDOG1	WDOG0	I2C0	HFXO0	HFRCO1	LESENSE	PCNT	VDAC0	AMUXCP0	ACMP1	ACMP0	IADC0	LETIMER0	SMUCFGNS	SMU				DMEM	KEYSCAN	LCD	SYSRTC

Bit	Name	Reset	Access	Description
31:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24	MVP	0x1	RW	**MVP Privileged Access**
	MVP Privileged Access			
23	SEMAILBOX	0x1	RW	**SEMAILBOX Privileged Access**
	SEMAILBOX Privileged Access			
22	EUSART0	0x1	RW	**EUSART0 Privileged Access**
	EUSART0 Privileged Access			
21	WDOG1	0x1	RW	**WDOG1 Privileged Access**
	WDOG1 Privileged Access			
20	WDOG0	0x1	RW	**WDOG0 Privileged Access**
	WDOG0 Privileged Access			
19	I2C0	0x1	RW	**I2C0 Privileged Access**
	I2C0 Privileged Access			
18	HFXO0	0x1	RW	**HFXO0 Privileged Access**
	HFXO0 Privileged Access			
17	HFRCO1	0x1	RW	**HFRCO1 Privileged Access**
	HFRCO1 Privileged Access			
16	LESENSE	0x1	RW	**LESENSE Privileged Access**
	LESENSE Privileged Access			
15	PCNT	0x1	RW	**PCNT Privileged Access**
	PCNT Privileged Access			
14	VDAC0	0x1	RW	**VDAC0 Privileged Access**
	VDAC0 Privileged Access			
13	AMUXCP0	0x1	RW	**AMUXCP0 Privileged Access**
	AMUXCP0 Privileged Access			
12	ACMP1	0x1	RW	**ACMP1 Privileged Access**
	ACMP1 Privileged Access			

Bit	Name	Reset	Access	Description
11	ACMP0	0x1	RW	**ACMP0 Privileged Access**
	ACMP0 Privileged Access			
10	IADC0	0x1	RW	**IADC0 Privileged Access**
	IADC0 Privileged Access			
9	LETIMER0	0x1	RW	**LETIMER0 Privileged Access**
	LETIMER0 Privileged Access			
8	SMUCFGNS	0x1	RW	**SMUCFGNS Privileged Access**
	SMUCFGNS Privileged Access			
7	SMU	0x1	RW	**SMU Privileged Access**
	SMU Privileged Access			
6:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	DMEM	0x1	RW	**DMEM Privileged Access**
	DMEM Privileged Access			
2	KEYSCAN	0x1	RW	**KEYSCAN Privileged Access**
	KEYSCAN Privileged Access			
1	LCD	0x1	RW	**LCD Privileged Access**
	LCD Privileged Access			
0	SYSRTC	0x1	RW	**SYSRTC Privileged Access**
	SYSRTC Privileged Access			

9.5.9 SMU_PPUSATD0 - Secure Access

Offset	Bit Position																															
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1		0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	
Access	RW	RW	RW	RW	RW	RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Name	EUSART2	EUSART1	HOSTMAILBOX	DCDC	GPCRC	BURAM	SYSCFG	SYSCFGCFGNS		I2C1	BURTC	USART0	TIMER4	TIMER3	TIMER2	TIMER1	TIMER0	LDMAXBAR	LDMA	GPIO	PRS	ICACHE0	MSC	ULFRCO	LFRCO	LFXO	DPLL0	FSRCO	HFRCO0	CMU	EMU	

Bit	Name	Reset	Access	Description
31	EUSART2	0x1	RW	**EUSART2 Secure Access**
	EUSART2 Secure Access			
30	EUSART1	0x1	RW	**EUSART1 Secure Access**
	EUSART1 Secure Access			
29	HOSTMAILBOX	0x1	RW	**HOSTMAILBOX Secure Access**
	HOSTMAILBOX Secure Access			
28	DCDC	0x1	RW	**DCDC Secure Access**
	DCDC Secure Access			
27	GPCRC	0x1	RW	**GPCRC Secure Access**
	GPCRC Secure Access			
26	BURAM	0x1	RW	**BURAM Secure Access**
	BURAM Secure Access			
25	SYSCFG	0x1	RW	**SYSCFG Secure Access**
	SYSCFG Secure Access			
24	SYSCFGCFGNS	0x1	RW	**SYSCFGCFGNS Secure Access**
	SYSCFGCFGNS Secure Access			
23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22	I2C1	0x1	RW	**I2C1 Secure Access**
	I2C1 Secure Access			
21	BURTC	0x1	RW	**BURTC Secure Access**
	BURTC Secure Access			
20	USART0	0x1	RW	**USART0 Secure Access**
	USART0 Secure Access			
19	TIMER4	0x1	RW	**TIMER4 Secure Access**
	TIMER4 Secure Access			
18	TIMER3	0x1	RW	**TIMER3 Secure Access**

Bit	Name	Reset	Access	Description
	TIMER3 Secure Access			
17	TIMER2	0x1	RW	**TIMER2 Secure Access**
	TIMER2 Secure Access			
16	TIMER1	0x1	RW	**TIMER1 Secure Access**
	TIMER1 Secure Access			
15	TIMER0	0x1	RW	**TIMER0 Secure Access**
	TIMER0 Secure Access			
14	LDMAXBAR	0x1	RW	**LDMAXBAR Secure Access**
	LDMAXBAR Secure Access			
13	LDMA	0x1	RW	**LDMA Secure Access**
	LDMA Secure Access			
12	GPIO	0x1	RW	**GPIO Secure Access**
	GPIO Secure Access			
11	PRS	0x1	RW	**PRS Secure Access**
	PRS Secure Access			
10	ICACHE0	0x1	RW	**ICACHE0 Secure Access**
	ICACHE0 Secure Access			
9	MSC	0x1	RW	**MSC Secure Access**
	MSC Secure Access			
8	ULFRCO	0x1	RW	**ULFRCO Secure Access**
	ULFRCO Secure Access			
7	LFRCO	0x1	RW	**LFRCO Secure Access**
	LFRCO Secure Access			
6	LFXO	0x1	RW	**LFXO Secure Access**
	LFXO Secure Access			
5	DPLL0	0x1	RW	**DPLL0 Secure Access**
	DPLL0 Secure Access			
4	FSRCO	0x1	RW	**FSRCO Secure Access**
	FSRCO Secure Access			
3	HFRCO0	0x1	RW	**HFRCO0 Secure Access**
	HFRCO0 Secure Access			
2	CMU	0x1	RW	**CMU Secure Access**
	CMU Secure Access			
1	EMU	0x1	RW	**EMU Secure Access**
	EMU Secure Access			
0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

9.5.10 SMU_PPUSATD1 - Secure Access

Offset	Bit Position																													
0x064	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset								0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1				0x1	0x1	0x1	0x1
Access								RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW				RW	RW	RW	RW
Name								MVP	SEMAILBOX	EUSART0	WDOG1	WDOG0	I2C0	HFXO0	HFRCO1	LESENSE	PCNT	VDAC0	AMUXCP0	ACMP1	ACMP0	IADC0	LETIMER0	SMUCFGNS	SMU				DMEM	KEYSCAN	LCD	SYSRTC

Bit	Name	Reset	Access	Description
31:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24	MVP	0x1	RW	**MVP Secure Access**
	MVP Secure Access			
23	SEMAILBOX	0x1	RW	**SEMAILBOX Secure Access**
	SEMAILBOX Secure Access			
22	EUSART0	0x1	RW	**EUSART0 Secure Access**
	EUSART0 Secure Access			
21	WDOG1	0x1	RW	**WDOG1 Secure Access**
	WDOG1 Secure Access			
20	WDOG0	0x1	RW	**WDOG0 Secure Access**
	WDOG0 Secure Access			
19	I2C0	0x1	RW	**I2C0 Secure Access**
	I2C0 Secure Access			
18	HFXO0	0x1	RW	**HFXO0 Secure Access**
	HFXO0 Secure Access			
17	HFRCO1	0x1	RW	**HFRCO1 Secure Access**
	HFRCO1 Secure Access			
16	LESENSE	0x1	RW	**LESENSE Secure Access**
	LESENSE Secure Access			
15	PCNT	0x1	RW	**PCNT Secure Access**
	PCNT Secure Access			
14	VDAC0	0x1	RW	**VDAC0 Secure Access**
	VDAC0 Secure Access			
13	AMUXCP0	0x1	RW	**AMUXCP0 Secure Access**
	AMUXCP0 Secure Access			
12	ACMP1	0x1	RW	**ACMP1 Secure Access**
	ACMP1 Secure Access			

Bit	Name	Reset	Access	Description
11	ACMP0	0x1	RW	**ACMP0 Secure Access**
	ACMP0 Secure Access			
10	IADC0	0x1	RW	**IADC0 Secure Access**
	IADC0 Secure Access			
9	LETIMER0	0x1	RW	**LETIMER0 Secure Access**
	LETIMER0 Secure Access			
8	SMUCFGNS	0x1	RW	**SMUCFGNS Secure Access**
	SMUCFGNS Secure Access			
7	SMU	0x1	RW	**SMU Secure Access**
	SMU Secure Access			
6:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	DMEM	0x1	RW	**DMEM Secure Access**
	DMEM Secure Access			
2	KEYSCAN	0x1	RW	**KEYSCAN Secure Access**
	KEYSCAN Secure Access			
1	LCD	0x1	RW	**LCD Secure Access**
	LCD Secure Access			
0	SYSRTC	0x1	RW	**SYSRTC Secure Access**
	SYSRTC Secure Access			

9.5.11 SMU_PPUFS - Fault Status

Offset	Bit Position																													
0x140	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0				
Access																												R				
Name																												PPUFSPERIPHID				

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	PPUFSPERIPHID	0x0	R	**Peripheral ID**
	ID of the peripheral that caused the fault.			

9.5.12 SMU_BMPUPATD0 - Privileged Attribute

Offset	Bit Position																														
0x150	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x1			0x1	0x1	0x1	0x1		
Access																								RW			RW	RW	RW	RW		
Name																								SEEXTDMA			MVPAHBDATA2	MVPAHBDATA1	MVPAHBDATA0	LDMA		

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	SEEXTDMA	0x1	RW	**SEEXTDMA privileged mode**
	SEEXTDMA privileged mode			
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	MVPAHBDATA2	0x1	RW	**MVPAHBDATA2 privileged mode**
	MVPAHBDATA2 privileged mode			
4	MVPAHBDATA1	0x1	RW	**MVPAHBDATA1 privileged mode**
	MVPAHBDATA1 privileged mode			
3	MVPAHBDATA0	0x1	RW	**MVPAHBDATA0 privileged mode**
	MVPAHBDATA0 privileged mode			
2	LDMA	0x1	RW	**MCU LDMA privileged mode**
	MCU LDMA privileged mode			
1:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

9.5.13 SMU_BMPUSATD0 - Secure Attribute

Offset	Bit Position																														
0x170	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x1			0x1	0x1	0x1	0x1		
Access																								RW			RW	RW	RW	RW		
Name																								SEEXTDMA			MVPAHBDATA2	MVPAHBDATA1	MVPAHBDATA0	LDMA		

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	SEEXTDMA	0x1	RW	**SEEXTDMA secure mode**
	SEEXTDMA secure mode			
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	MVPAHBDATA2	0x1	RW	**MVPAHBDATA2 secure mode**
	MVPAHBDATA2 secure mode			
4	MVPAHBDATA1	0x1	RW	**MVPAHBDATA1 secure mode**
	MVPAHBDATA1 secure mode			
3	MVPAHBDATA0	0x1	RW	**MVPAHBDATA0 secure mode**
	MVPAHBDATA0 secure mode			
2	LDMA	0x1	RW	**MCU LDMA secure mode**
	MCU LDMA secure mode			
1:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

9.5.14 SMU_BMPUFS - Fault Status

Offset	Bit Position																															
0x250	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									BMPUFSMASTERID							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	BMPUFSMASTERID	0x0	R	**Bus Manager ID**
	ID of Bus Manager that triggered fault			

9.5.15 SMU_BMPUFSADDR - Fault Status Address

Offset	Bit Position																															
0x254	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x0																
Access																R																
Name																BMPUFSADDR																

Bit	Name	Reset	Access	Description
31:0	BMPUFSADDR	0x0	R	**Fault Address**
	Access address that triggered fault			

9.5.16 SMU_ESAURTYPES0 - Region Types 0

Offset	Bit Position																															
0x260	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																				0x0												
Access																				RW												
Name																				ESAUR3NS												

Bit	Name	Reset	Access	Description
31:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	ESAUR3NS	0x0	RW	**Region 3 Non-Secure** Set to 1 to configure Region 3 as Non-secure
11:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

9.5.17 SMU_ESAURTYPES1 - Region Types 1

Offset	Bit Position																															
0x264	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																				0x0												
Access																				RW												
Name																				ESAUR11NS												

Bit	Name	Reset	Access	Description
31:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	ESAUR11NS	0x0	RW	**Region 11 Non-Secure** Set to 1 to configure Region 11 as Non-secure
11:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

9.5.18 SMU_ESAUMRB01 - Movable Region Boundary

Offset	Bit Position																															
0x270	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0xA000																											
Access					RW																											
Name					ESAUMRB01																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:12	ESAUMRB01	0xA000	RW	**Moveable Region Boundary**
	Moveable Region Boundary between Region 0 and Region 1. Address Represents the start of Region 1 at a 4kB offset.			
11:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

9.5.19 SMU_ESAUMRB12 - Movable Region Boundary

Offset	Bit Position																															
0x274	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0xC000																											
Access					RW																											
Name					ESAUMRB12																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:12	ESAUMRB12	0xC000	RW	**Moveable Region Boundary**
	Moveable Region Boundary between Region 1 and Region 2. Address Represents the start of Region 2 at a 4kB offset.			
11:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

9.5.20 SMU_ESAUMRB45 - Movable Region Boundary

Offset	Bit Position																															
0x280	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x2000																											
Access					RW																											
Name					ESAUMRB45																											

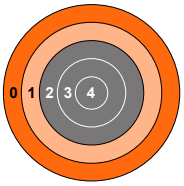
Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:12	ESAUMRB45	0x2000	RW	**Moveable Region Boundary**
	Moveable Region Boundary between Regions 4 and 5. This represents the starting address of Region 5.			
11:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

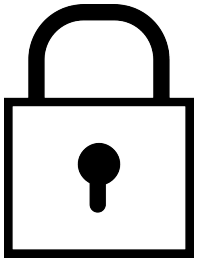
9.5.21 SMU_ESAUMRB56 - Movable Region Boundary

Offset	Bit Position																															
0x284	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x4000																											
Access					RW																											
Name					ESAUMRB56																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:12	ESAUMRB56	0x4000	RW	**Moveable Region Boundary**
	Moveable Region Boundary between Regions 5 and 6. This represents the starting address of Region 6.			
11:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

10. SE - Secure Engine Subsystem





Quick Facts

What?

The Secure Engine Subsystem encapsulates security peripherals providing both improved system security and ease of use.

Why?

Isolation of security hardware from the Cortex®-M33 protects the SE system from exploits that target the main CPU. The subsystem also provides autonomous cryptographic operations allowing the main CPU to perform other tasks or enter EM1 to save power.

How?

Security peripherals are completely isolated from the main CPU and controlled with a processor internal to the SE subsystem.

10.1 Introduction

The Secure Engine (SE) provides several security features and acts as a barrier protecting the security hardware from activity on the Cortex®-M33. It also enables autonomous operation of security features.

Available features include:

- Secure Boot with Root of Trust and Secure Loader (RTSL)
- Hardware Cryptographic Acceleration with DPA countermeasures for AES128/256, SHA-1, SHA-2 (up to 256-bit), ECC (up to 256-bit), ECDSA, ECDH and J-Pake
- True Random Number Generator (TRNG) compliant with NIST SP800-90 and AIS-31
- ARM® TrustZone®
- Secure Debug with lock/unlock

All Secure Engine functions are enabled by software. These functions are fully described in the Secure Engine emlib online documentation located at the following link:

https://docs.silabs.com/mcu/latest/efr32mg21/group-SE

10.2 Security Features

10.2.1 Security Features Overview

- Acceleration of cryptographic functions
  - AES encryption and decryption with 128, 192, or 256-bit keys
  - Supported block cipher modes of operation for AES include: ECB, CTR, CBC, CFB, CBC-MAC, CMAC, CCM, GCM and GMAC.
  - ECC over GF(P) up to 256-bit
  - Supported ECC NIST recommended curves include P-192 and P-256
  - SHA-1 and SHA-2 up to 256-bit
- True Random Number Generation
  - Entropy Source complies to NIST 800-90B requirements
  - Online Health tests comply to NIST 800-90 and AIS31 requirements
  - Random Data Passes NIST 800-22 and NIST 800-90B test suites
- Secure Boot Loader (First Stage Boot Loader)

10.2.2 Secure Boot with Root of Trust and Secure Loader (RTSL)

The Secure Boot with RTSL authenticates a chain of trusted firmware that begins from an immutable memory (ROM).

It prevents malware injection, prevents rollback, ensures that only authentic firmware is executed, and protects Over The Air updates.

For more information about this feature, see AN1218: Series 2 Secure Boot with RTSL.

10.2.3 Secure Debug

The SE provides a secure debug unlock function that allows users to grant debug access to locked devices on a device by device basis. To use this function the device must be programmed with a public Command key by the user. To unlock a device, a unique challenge (a device-unique persistent random set of bytes) must be read out and signed by the private key associated with public Command key creating an unlock token. The device can then be unlocked by providing the valid unlock token. The token can be used to unlock the device any number of times. There is also a command to force the device to update its challenge, which revokes the previously-generated token.

More information on Secure Debug can be found in the AN1190: Secure Debug application note.

Note: Secure debug locking a device will limit the capability for Silicon Labs to perform failure analysis on the device. Provide secure debug tokens for each device when submitting parts for failure analysis.

10.2.4 Cryptographic Accelerator

The Cryptographic Accelerator in Secure Engine is an autonomous hardware accelerator with Differential Power Analysis (DPA) countermeasures to protect keys.

It supports AES encryption and decryption with 128/192/256-bit keys, Elliptic Curve Cryptography(ECC) to support public key operations and hashes.

Supported block cipher modes of operation for AES include:

- ECB (Electronic Code Book)
- CTR (Counter Mode)
- CBC (Cipher Block Chaining)
- CFB (Cipher Feedback)
- GCM (Galois Counter Mode)
- CCM (Counter with CBC-MAC)
- CBC-MAC (Cipher Block Chaining Message Authentication Code)
- GMAC (Galois Message Authentication Code)

The Cryptographic Accelerator accelerates Elliptical Curve Cryptography and supports the NIST (National Institute of Standards and Technology) recommended curves including P-192 and P-256 for ECDH(Elliptic Curve Diffie-Hellman) key derivation and ECDSA (Elliptic Curve Digital Signature Algorithm) sign and verify operations.

Secure Engine also supports ECJ-PAKE (Elliptic Curve variant of Password Authenticated Key Exchange by Juggling).

Supported hashes include SHA-1, SHA2/224, and SHA-2/256.

This implementation provides a fast and energy efficient solution to state of the art cryptographic needs.

Note: AES_ECB, AES_CBC, AES_CBCMAC, and SHA-1 are provided for legacy compatibility and are not recommended for cryptographic purposes without thoroughly understanding their potential security weaknesses.

10.2.5 True Random Number Generation

The SE provides access to a non-deterministic random number generator based on a full hardware solution. The TRNG output passes the NIST 800-22 and AIS31 test suites. The TRNG module includes several built-in self tests to detect issues with the noise source, ensure entropy, and meet cryptography standards. The Repetition Count Test and Adaptive Proportion Test with window sizes of 64 and 4096 bits described in section 6.5.1.2 of NIST-800-90B are implemented in hardware and run continuously on the data.

http://csrc.nist.gov/publications/drafts/800-90/draft-sp800-90b.pdf

The AIS31 Online Test described in section 5.5.3 of AIS 31 is also implemented in hardware, and runs continuously on the data.

https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_Functionality_classes_for_random_number_generators_e.pdf

10.3 SE Mailbox

All communication with the Secure Engine Subsystem takes place through the SE Mailbox. Operations are performed by using the mailbox to sending a command and then receive the SE response. The mailbox is a bidirectional 64 word FIFO.

10.3.1 Sending Commands

The TX FIFO has two status flags in SE_TX_STATUS register. TXFULL is set when the FIFO is full and TXINT is set if there is space in the FIFO for at least 16 words. If TXINTEN in SE_CONFIGURATION is set an interrupt will be generated when TXINT is set.

Writing to any SE_DATAn register will result in data being placed in the FIFO. For example, to write 16 words to the FIFO software may write SE_DATA0 16 times, or may make a single write to each of the 16 SE_DATAn registers. If the FIFO is written when no space is available, the CPU will be stalled until spaces becomes available and the write can be completed.

To send a command, first check TXINT to ensure that there is space available in the FIFO. Then write SE_TX_HEADER with the command length and protection bit. Finally, write the command data into the SE_DATAn registers. While the command is being written, BYTERM in SE_TX_STATUS will contain the number of bytes remaining in the command. To ensure minimal performance impact, software should ensure that space exists in the FIFO before writing to it.

10.3.2 Receiving Responses

The RX FIFO has two status flags in SE_RX_STATUS register. RXEMPTY is set when the FIFO is empty and RXINT is set if there are at least 4 words in the FIFO or if the final word of the message is present in the FIFO. If RXINTEN in SE_CONFIGURATION is set, an interrupt will be generated when RXINT is set.

Reading from any SE_DATAn register will result in data being read from the FIFO. For example, to read 16 words from the FIFO, software may read SE_DATA0 16 times, or may make a single read from each of the 16 SE_DATAn registers. If the FIFO is read when it is empty and no message is available, a 0x0 will be read. If the FIFO is read when empty and a message is being processed, the CPU will be stalled until data becomes available.

Software may check for responses by polling RXINT, RXEMPTY, or RXHEADER in SE_RX_STATUS. The RXINT interrupt may also be used to notify the CPU when data is available. To receive a response first read the response header from SE_RX_HEADER. Software may read the message size from SE_RX_HEADER, or use BYTERM in SE_RX_STATUS, which contains the number of words remaining in the response.

The command status is available in both SE_RX_STATUS and SE_RX_HEADER and indicates if the command completed successfully.

10.3.3 MAILBOX Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	MAILBOX_MSGPTRx	RW	Message Pointer
0x040	MAILBOX_IF	RW INTFLAG	Interrupt Flag Register
0x044	MAILBOX_IEN	RW	Interrupt Enable Register
0x1000	MAILBOX_MSGPTRx_SET	RW	Message Pointer
0x1040	MAILBOX_IF_SET	RW INTFLAG	Interrupt Flag Register
0x1044	MAILBOX_IEN_SET	RW	Interrupt Enable Register
0x2000	MAILBOX_MSGPTRx_CLR	RW	Message Pointer
0x2040	MAILBOX_IF_CLR	RW INTFLAG	Interrupt Flag Register
0x2044	MAILBOX_IEN_CLR	RW	Interrupt Enable Register
0x3000	MAILBOX_MSGPTRx_TGL	RW	Message Pointer
0x3040	MAILBOX_IF_TGL	RW INTFLAG	Interrupt Flag Register
0x3044	MAILBOX_IEN_TGL	RW	Interrupt Enable Register

10.3.4 MAILBOX Register Description

10.3.4.1 MAILBOX_MSGPTRx - Message Pointer

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	PTR																															

Bit	Name	Reset	Access	Description
31:0	PTR	0x0	RW	**Pointer**
	The Memory Address of the message.			

10.3.4.2 MAILBOX_IF - Interrupt Flag Register

Offset	Bit Position																													
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													RW	RW	RW	RW
Name																													MBOXIF3	MBOXIF2	MBOXIF1	MBOXIF0

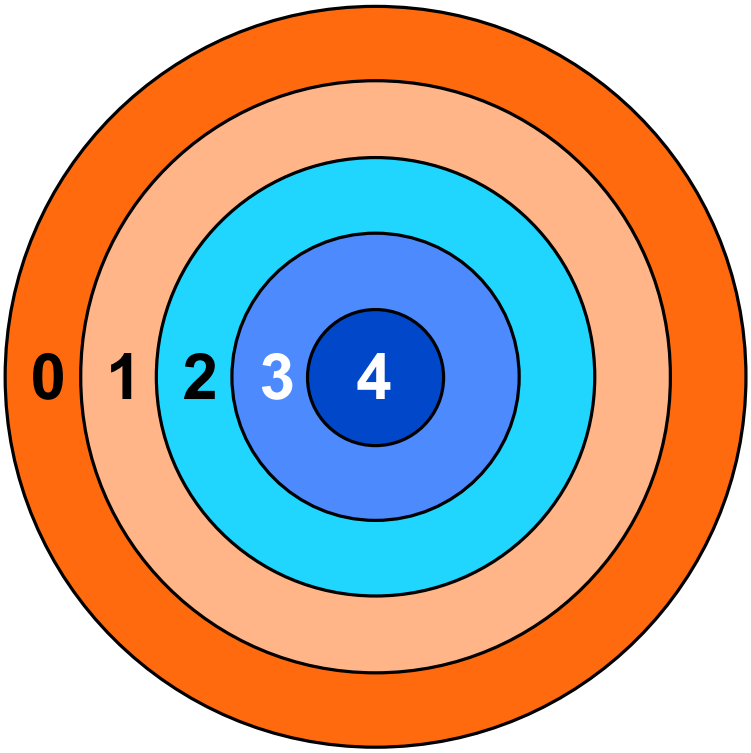
Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	MBOXIF3	0x0	RW	**Mailbox Interupt Flag**
	When set, Mailbox Message Availble.			
2	MBOXIF2	0x0	RW	**Mailbox Interupt Flag**
	When set, Mailbox Message Availble.			
1	MBOXIF1	0x0	RW	**Mailbox Interupt Flag**
	When set, Mailbox Message Availble.			
0	MBOXIF0	0x0	RW	**Mailbox Interupt Flag**
	When set, Mailbox Message Availble.			

10.3.4.3 MAILBOX_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													RW	RW	RW	RW
Name																													MBOXIEN3	MBOXIEN2	MBOXIEN1	MBOXIEN0

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	MBOXIEN3	0x0	RW	**Mailbox Interrupt Enable**
	Set to enable the MBOXIF Interrupt			
2	MBOXIEN2	0x0	RW	**Mailbox Interrupt Enable**
	Set to enable the MBOXIF Interrupt			
1	MBOXIEN1	0x0	RW	**Mailbox Interrupt Enable**
	Set to enable the MBOXIF Interrupt			
0	MBOXIEN0	0x0	RW	**Mailbox Interrupt Enable**
	Set to enable the MBOXIF Interrupt			

11. EMU - Energy Management Unit



Quick Facts

What?

The EMU (Energy Management Unit) handles the different low energy modes in EFM32PG28

Why?

The need for performance and peripheral functions varies over time in most applications. By efficiently scaling the available resources in real time to match the demands of the application, the energy consumption can be kept at a minimum.

How?

With a broad selection of energy modes, a high number of low-energy peripherals available even in EM2, and short wake-up time, applications can dynamically minimize energy consumption during program execution.

11.1 Introduction

The Energy Management Unit (EMU) manages all the low energy modes (EM) in EFM32PG28. Each energy mode manages whether the CPU and the various peripherals are available. The energy modes range from EM0 to EM4. EM0 mode provides the highest amount of features, enabling the CPU and peripherals with the highest clock frequency. EM4 Mode provides the lowest power state, allowing the part to return to EM0 on a wake-up condition. The EMU also controls the internal regulators settings and voltage monitoring needed for optimal power configuration and protection.

11.2 Features

The primary features of the EMU are listed below:

- Energy Modes control
  - Entry into EM4
  - Configuration of regulators and clocks for each Energy Mode
  - Configuration of various EM4 wake-up conditions
  - Configuration of GPIO retention settings
- Power routing configurations
  - DCDC control and bypass
- Temperature sensor
- Brown Out Detection
- Supply voltage scaling
  - EM0 / EM1 voltage scaling
  - EM2 / EM3 voltage scaling
- Reset Management
  - Power-on Reset (POR)
  - Brown-out Detection (BOD) on the following power domains:
    - Analog Unregulated Power Domain AVDD
    - Digital Unregulated Power Domain DVDD
    - I/O Unregulated Power Domain IOVDDx
    - Regulated Digital Domain DECOUPLE (DEC)
  - RESETn pin reset
  - Watchdog (WDOG) reset
  - Software triggered reset (SYSRESETREQ)
  - Core LOCKUP condition
  - EM4 Detection
  - EM4 wakeup reset from GPIO pin
  - Configurable reset levels
  - A software readable register indicates the cause of the last reset

11.3 Functional Description

The EMU is responsible for managing the wide range of energy modes available in EFM32PG28. The block works in harmony with the entire platform to easily transition between energy modes in the most efficient manner possible. The following diagram Figure 11.1 EMU Overview on page 276, shows the relative connectivity to the various blocks in the system.

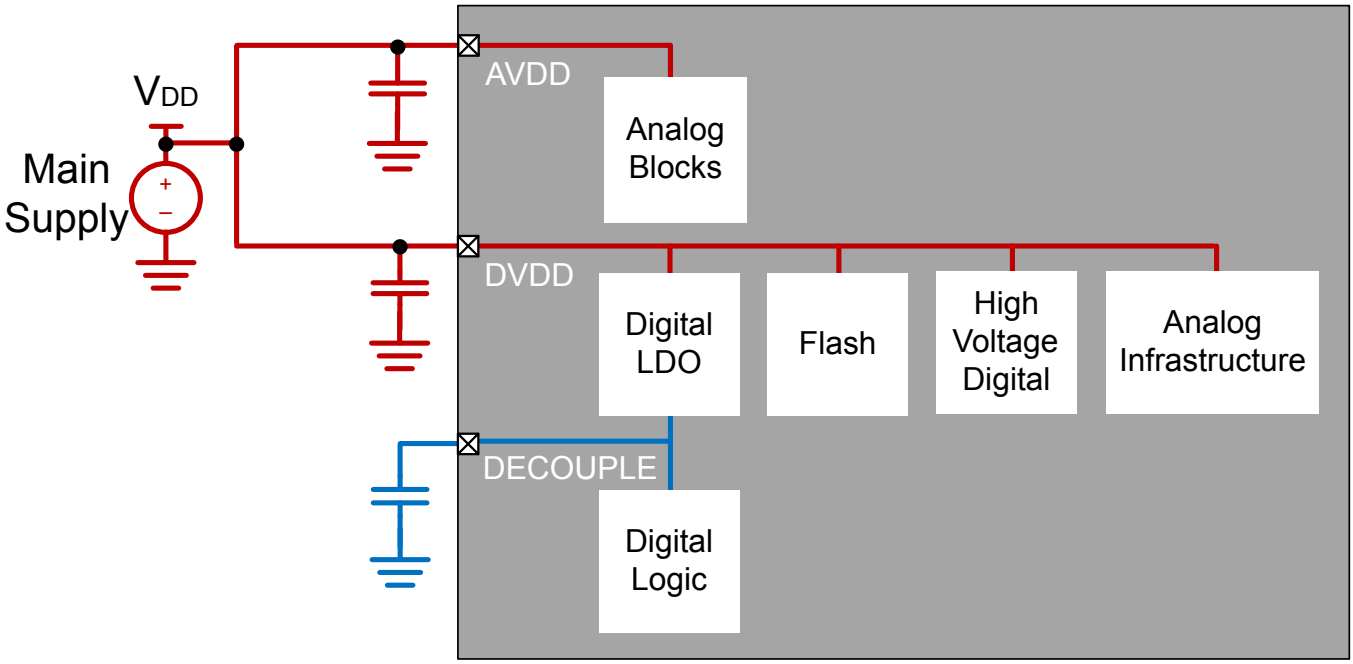


Figure 11.1. EMU Overview

The EMU is available on the peripheral bus. The energy management state machine controls the internal voltage regulators, oscillators, memories, and interrupt system. Events, interrupts, and resets can trigger the energy management state machine to return to the active state. This is further described in the following sections.

11.3.1 Energy Modes

EFM32PG28 features five main energy modes, referred to as Energy Mode 0 (EM0) through Energy Mode 4 (EM4). The Cortex®-M33 is only available for program execution in EM0. In EM0 Active/EM1 Sleep any peripheral function can be enabled. EM2 through EM4, also referred to as low energy modes, provide a significantly reduced energy consumption while still allowing a rich set of peripheral functionality. The following Table 11.1 table on page 277 shows the possible transitions between different energy modes.

Table 11.1. Energy Mode Transitions

Current Mode	EM Transition Action				
	Enter EM0	Enter EM1	Enter EM2	Enter EM3	Enter EM4
EM0		Sleep (WFI, WFE)	Deep Sleep (WFI, WFE)	Deep Sleep (WFI, WFE)	EM4 Entry
EM1	IRQ		Peripheral wake up done[1]	Peripheral wake up done [1]	
EM2	IRQ	Peripheral wake up req [1]			
EM3	IRQ	Peripheral wake up req [1]			
EM4	Wake Up				

Note:
1. Peripheral wake-up from EM2/3 to EM1 and then automatically back to EM2/3 when done.

Certain peripherals have the ability to temporarily turn on additional logic in EM2 or EM3 to receive and transmit data or trigger LDMA transfers without intervention from the M33 core. The system automatically returns to the original energy mode when such operations are complete.

The Core can always request to go to EM1 with the WFI or WFE command during EM0. The core will be prevented from entering EM2 or EM3 if flash is programming or erasing.

An overview of supported energy modes and available functionality is shown in the following table. For each energy mode, the system will typically default to its lowest power configuration, with non-essential clocks and peripherals disabled. Functionality may be then selectively enabled by software.

Modules with EM2/3/4 capability exist in a Low Power Domain (e.g., PD0x or PDHV). Refer to 11.3.4 Power Domains for more details.

Table 11.2. Energy Modes

	EM0 / EM1	EM2	EM3	EM4
Cortex®-M33 Core Active	Yes, in EM0 only	-	-	-
Debug	Available	See Note [1]	See Note [1]	-
Digital logic and system RAM retained	Yes	Yes	Yes	-
Flash Memory Access	Available	-	-	-
LDMA (Linked DMA Controller)	Available	Available [2]	Available [2]	-
High Frequency Oscillators (HFXO,HFRCODPLL) and Clocks (BUSCLK, HCLK, PCLK, RADIOCLK, EM01GRPACLK, EM01GRPCCLK)	Available	-	-	-

	EM0 / EM1	EM2	EM3	EM4
Fast Startup RC Oscillator (FSRCO), EM2/3 High Frequency Oscillator (HFRCOEM23), ADC Clock (IADCCLK), and VDAC0 Clock (VDAC0CLK)	Available	Available [3]	Available [3]	-
Low Frequency Oscillators (LFRCO, LFXO)	Available	Available	-	Available [4]
Low Energy Clocks (EM23GRPACLK, WDOGnCLK, SYSRTCCCLK, LCDCLK, PCNT0CLK, EUSART0CLK)	Available	Available	Available [5]	-
EM4 Clock (EM4GRPACLK)	Available	Available	Available [5]	Available [4]
ULFRCO (Ultra Low Frequency Oscillator)	On	On	On	Available[4]
GPCRC (General Purpose Cyclic Redundancy Check)	Available	-	-	-
BURTC	Available	Available	Available [5]	Available
SYSRTC	Available	Available	Available[5]	-
BURAM	Available	Available	Available	Available
MVP	Available	-	-	-
LCD	Available	Available	Available[5]	
USART (UART/SPI)	Available	-	-	-
I^2C	Available	Available [7 12]	Available [7 12]	-
EUSART	Available	Available[8 12]	Available[8 12]	-
TIMER (Timer/Counter)	Available	-	-	-
LETIMER (Low Energy Timer)	Available	Available [12]	Available [5 12]	-
WDOG (Watchdog)	Available	Available [12]	Available [5 12]	-
ACMP (Analog Comparator)	Available	Available [6 12]	Available [6 12]	-
IADC (Analog to Digital Converter)	Available	Available [2 12]	Available [2 12]	-
VDAC (Digital-to-Analog Converter)	Available	Available [2 12]	Available [2 12]	
LESENSE (Low energy Sense)	Available	Available [2 12]	Available [2 12]	
PCNT (Pulse Counter)	Available	Available [2 12]	Available [2 12]	
KEYSCAN	Available	Available [12]	Available [12]	-
DC-DC	Available	Available	Available	-
EMU Temperature Change	Available	Available	Available	-
Brown-Out Detect/Power-on Reset	Available	Available	Available	Available
Pin Reset	Available	Available	Available	Available

	EM0 / EM1	EM2	EM3	EM4
PRS (Peripheral Reflex System)	Available	Available[12]	Available[12]	-
GPIO Pin Interrupts	Available	Available[12]	Available[12]	Available [10]
GPIO Pin State Retention	Yes	Yes	Yes	Available [11]

Note:

1. Leaving the debugger connected when in EM2 or EM3 will cause the system to enter a higher power EM2 mode in which the high frequency clocks are still enabled and certain core functionality is still powered-up in order to maintain debug-functionality.
2. The LDMA can be used with some low power peripherals (e.g., IADC) in EM2/3. Features required by the LDMA which are not supported in EM2/3 (e.g., HCLK), will be automatically enabled prior to the LDMA transfer and then automatically disabled afterwards.
3. Default off, but kept active if used by the IADC or VDAC.
4. Default off, but kept active if used by the BURTC
5. Must be using ULFRCO
6. ACMP functionality in EM2/3 limited to edge interrupt
7. I2C0 only. Not supported on all GPIO Ports. Functionality limited to receive address recognition
8. EUSART0 only. Not supported on all GPIO Ports. Functionality limited to low-frequency UART or SPI secondary interface
9. Wake on any key press supported in EM2/3. Full key scanning operates in EM0/1.
10. Pin wake-up in EM4 supported only on GPIO_EM4WUx pins. Consult data sheet for complete list of pins.
11. If enabled in EMU->EM4CTRL.EM4IORETMODE.
12. Module is in a PD0x Low Power Domain. Refer to 11.3.4 Power Domains for more detail.

The different energy modes are summarized in the following sections.

11.3.1.1 EM0

EM0 provides all system features.

- Cortex®-M33 is executing code
- High and low frequency clock trees are active
- All oscillators are available
- All peripheral functionality is available

11.3.1.2 EM1

EM1 disables the core but leaves the remaining system fully available.

- Cortex®-M33 is in sleep mode. Clocks to the core are off
- High and low frequency clock trees are active
- All oscillators are available
- All peripheral functionality is available

11.3.1.3 EM2

This is the first level into the low power energy modes. Most of the high frequency peripherals are disabled or have reduced functionality. Memory and registers retain their values.

- Cortex®-M33 is in sleep mode. Clocks to the core are off.
- High frequency clock tree is inactive
- Low frequency clock tree is active
- The following oscillators are available
  - LFRCO, LFXO, ULFRCO
  - On-demand if used by peripherals: FSRCO, HFRCOEM23
- The following low frequency peripherals are available
  - SYSRTC, BURTC, WDOG, LETIMER, PCNT, LESENSE, I2C0, EUSART0 (UART or SPI secondary only), KEYSCAN, and LCD
- The following analog peripherals are available (with potential limitations on functionality)
  - ACMP, IADC, VDAC
- Wake-up to EM0 through
  - Peripheral interrupt, reset pin, power on reset, asynchronous pin interrupt, I2C0 address recognition
- Wake-up to EM1 through
  - Peripheral data transfer request
  - Part returns to EM2 when transfers are complete
- RAM and register values are preserved
  - RAM blocks may be optionally powered down for lower power
- GPIO pin state is retained
- SYSRTC memory is retained
- Debug connectivity is unavailable by default to reduce current consumption. Debug connectivity can be enabled by setting the EM2DBGEN bit in the EMU_CTRL register, and will consume about 0.5 uA extra supply current.

11.3.1.4 EM3

In this low energy mode, all low frequency oscillators (LFXO, LFRCO) and all low frequency clocks derived from them are stopped, as well as all high frequency clocks. Most peripherals are disabled or have reduced functionality. Memory and registers retain their values.

- Cortex®-M33 is in sleep mode. Clocks to the core are off.
- High frequency clock tree is inactive
- All low frequency clock trees derived from the low frequency oscillators (LFXO, LFRCO) are inactive
- The following oscillators are available
  - ULFRCO
  - On-demand if used by peripherals: FSRCO, HFRCOEM23
- The following low frequency peripherals are available if clocked by the ULFRCO
  - SYSRTC, BURTC, and WDOG
  - SYSRTC, BURTC, WDOG, LETIMER, PCNT, LESENSE, I2C0, EUSART0 (SPI secondary only), KEYSCAN, and LCD
- The following analog peripherals are available (with potential limitations on functionality)
  - ACMP, IADC, VDAC
- Wake-up to EM0 through
  - Peripheral interrupt, reset pin, power on reset, asynchronous pin interrupt, I2C0 address recognition
- Wake-up to EM1 through
  - Peripheral data transfer request
  - Part returns to EM3 when transfers are complete
- RAM and register values are preserved
  - RAM blocks may be optionally powered down for lower power
- GPIO pin state is retained
- SYSRTC memory is retained
- Debug connectivity is unavailable by default to reduce current consumption. Debug connectivity can be enabled by setting the EM2DBGEN bit in the EMU_CTRL register, and will consume about 0.5 uA extra supply current.

11.3.1.5 EM4

EM4 is the lowest energy mode of the part. There is no retention except for GPIO PAD state and BURAM values. Wake-up from EM4 requires a reset to the system, returning it back to EM0.

- Cortex®-M33 is off
- High frequency clock tree is off
- Low frequency clock tree may be active
- No RAM or register values are retained, except for the BURAM.
- The following oscillators are on if used by the BURTC:
  - LFRCO, LFXO, ULFRCO
- The following low frequency peripherals are available
  - BURTC
- Wake-up to EM0 through
  - BURTC interrupt, reset pin, power on reset, asynchronous pin interrupt (on GPIO_EM4WUx pins only), or RFSENSE
- GPIO pin state may be retained (depending on EMU->EM4CTRL.EM4IORETMODE configuration)

11.3.2 Entering Low Energy Modes

The following sections describe the requirements for entering the various energy modes.

11.3.2.1 Entry Into EM1

Energy mode EM1 is entered when the Cortex®-M33 executes the Wait For Interrupt (WFI) or Wait For Event (WFE) instruction while the SLEEPDEEP bit in the Cortex®-M33 System Control Register is cleared. The MCU can re-enter sleep automatically out of an Interrupt Service Routine (ISR) if the SLEEPONEXIT bit in the Cortex®-M33 System Control Register is set. Refer to ARM documentation on entering Sleep modes.

Alternatively, EM1 can be entered from either EM2 or EM3 due to certain peripheral wake-up requests, allowing transfers from the peripheral to system RAM. The system will return back to EM2 or EM3 once the peripheral has completed its transfers and processing.

11.3.2.2 Entry Into EM2 or EM3

Energy mode EM2 or EM3 may be entered when **all** of the following conditions are true:

- Cortex®-M33 (if present) is in DEEPSLEEP state
- Flash Program/Erase Inactive
- DMA done with all current requests
- A debugger is not currently connected.

  Note: The device will still enter a sleep state which emulates the behavior of EM2 or EM3 when an active debug connection is present, but it will draw more than the specified sleep current.

Energy mode EM2 is entered from EM0 when the Cortex®-M33 executes the Wait For Interrupt (WFI) or Wait For Event (WFE) instruction while the SLEEPDEEP bit in the Cortex®-M33 System Control Register is set. The MCU can re-enter DeepSleep automatically out of an Interrupt Service Routine (ISR) if the SLEEPONEXIT bit in the Cortex®-M33 System Control Register is set. Refer to ARM documentation on entering Sleep modes.

Alternately, EM2 or EM3 is entered from EM1 upon the completion of a Peripheral Wake-Up Request from capable peripherals if no EM0 wake-up happens in the meantime.

When entering EM2 or EM3, if any peripheral on an auxiliary low power domain (PD0B, PD0C, etc.) is enabled, that auxiliary low power domain will be powered, causing higher current draw. Otherwise, the auxiliary power domain will be powered down. See 11.3.4 Power Domains for more information.

11.3.2.3 Entry Into EM4

Energy mode EM4 is entered through register access.

Software must ensure no modules are active when entering EM4.

Software may enter EM4 from EM0 by writing the sequence 2,3,2,3,2,3,2,3,2 to EM4CTRL->EM4ENTRY bit field. If the EM4BLOCK bit in WDOGn_CTRL is set, the CPU will be prevented from entering EM4 by software request.

An active debugger connection will prevent entry into EM4.

11.3.3 Exiting a Low Energy Mode

A system in EM2 and EM3 can be woken up to EM0 through regular interrupt requests from active peripherals. Since state and RAM retention is available, the EFM32 Series 2 is fully restored and can continue to operate as before it went into the Low Energy Mode.

Wake-up from EM4 is performed through a reset. Wake-up from a specific module must be enabled in that module's EM4WUEN register.

Enabled interrupts that can cause wake-up from EM2, EM3, and EM4 are shown in the following table. The wake-up triggers always return the device to EM0. Additionally, any reset source will return to EM0.

Table 11.3. Wake-Up Triggers from Low Energy Modes

Peripheral	Wake-Up Trigger	EM2	EM3	EM4
LETIMER	Any enabled interrupt	Yes	-	-
LFXO	Ready Interrupt	Yes	-	-
LFRCO	Ready Interrupt	Yes	-	-
WDOG	Any enabled interrupt	Yes	Yes	-
I^2C0	Receive address recognition	Yes	Yes	-
EUSART0	Any enabled interrupt	Yes	-	-
IADC	SCAN and SINGLE FIFO events, Window comparator events	Yes	Yes	
ACMP	Any enabled edge interrupt	Yes	Yes	-
VDAC	Any enabled interrupt except ABUS conflict and allocation interrupts	Yes	Yes	
SYSRTC	Any enabled interrupt	Yes	Yes	-
BURTC	Timeout	Yes	Yes	Yes [1]
EMU Temperature Sensor	Measured temperature outside the defined limits	Yes	Yes	-
Pin Interrupts	Transition	Yes[2]	Yes[2]	Yes [1 3]
Reset Pin	Assertion	Yes	Yes	Yes
Power	Cycle Off/On	Yes	Yes	Yes

Note:

1. Corresponding bit in the module's EM4WUEN must be set.
2. Available on Port A, Port B, and all EM4WU pins.
3. Only available on EM4WU pins.

11.3.4 Power Domains

Peripherals may exist on several independent power domains which are powered down to minimize supply current when not in use. Power domains are managed automatically by the EMU.

The lowest-energy power domain is the "high-voltage" power domain (PDHV), which supports extremely low-energy infrastructure and peripherals. Circuits powered from PDHV are always on and available in all energy modes down to EM4.

The next power domain is the low power domain (PD0), which is further divided to power subsets of peripherals. All PD0 power domains are shut down in EM4. Circuits powered from PD0 power domains may be available in EM0, EM1, EM2, and EM3.

Low power domain A (PD0A) is the base power domain for EM2 and EM3 and will always remain on in EM0-EM3. It powers the most commonly-used EM2 and EM3-capable peripherals and infrastructure required to operate in EM2 and EM3. The lowest-power EM2 and EM3 operation is achieved when only the base PD0A power domain is active. Auxiliary PD0 power domains (PD0B, PD0C, PD0D, PD0E) power additional EM2 and EM3-capable peripherals on demand. If any peripherals on one of the auxiliary power domains is enabled, that power domain will be active in EM2 and EM3. Otherwise, the auxiliary PD0 power domains will be shut down to reduce current.

Note: Power domain PD0E is also turned on when peripherals on PD0B, PD0C, or PD0D are used.

The active power domain (PD1) powers the rest of the device circuitry, including the CPU core and EM0 / EM1 peripherals. PD1 is always powered on in EM0 and EM1. PD1 is always shut down in EM2, EM3, and EM4.

Table 11.4 Peripheral Power Subdomains on page 283 shows the peripherals on the PDHV and PD0x domains. Any peripheral not listed is on PD1.

Table 11.4. Peripheral Power Subdomains

Always On in EM2/EM3		Selectively On in EM2/3			
PDHV[1]	PD0A	PD0B[2]	PD0C[2]	PD0D[2]	PD0E
LFRCO	SYSRTC	LETIMER0	HFRCOEM23	DEBUG	GPIO
LFXO	FSRCO	IADC0	HFXO	WDOG0	KEYSCAN
BURTC	LCD	PCNT0		WDOG1	PRS
ULFRCO		ACMP0		EUSART0	
		ACMP1		I2C0	
		LESENSE			
		VDAC0			

Note:
1. Peripherals on PDHV are also available in EM4.
2. If any of PD0B, PD0C, or PD0D are enabled, PD0E will also be automatically enabled.

11.3.5 Voltage Scaling

The EFM32PG28 supports supply voltage scaling for the LDO powering DECOUPLE. Voltage scaling helps to optimize the energy efficiency of the system by operating at lower voltages when possible. Three supply voltage operating points are available:

Table 11.5. Voltage Scaling Options

VSCALE Setting	DECOUPLE Voltage	Operating Conditions
VSCALE2	1.1 V	EM0/EM1 Operation up to 80 MHz EM2 and EM3
VSCALE1	1.0 V	EM0/EM1 Operation up to 40 MHz EM2 and EM3
VSCALE0	0.9 V	EM2 and EM3 Only

11.3.5.1 Voltage Scaling in EM0 and EM1

In EM0 and EM1, the voltage scaling value should be set according to the desired operating frequency. The system defaults to VSCALE2 out of reset. To operate above 40 MHz, VSCALE2 should always be used. If the system will operate below 40 MHz, VSCALE1 may be used to save energy.

The voltage scaling value for EM0 and EM1 is changed via software command bits in the EMU_CMD register. Setting EMU_CMD_EM01VSCALE1 will switch to VSCALE1, and setting EMU_CMD_EM01VSCALE2 will switch to VSCALE2.

The command initiates a voltage change operation, but some time is needed before the new supply voltage is reached. When changing between VSCALE values in EM0, it takes approximately 150 us to ramp the voltage down and approximately 32 us to ramp the voltage up to the new values (see the data sheet specifications for exact numbers). During this time, SRAM access is prohibited by the hardware and any accesses to SRAM from the CPU or DMA will be blocked until the operation is complete. The EMU_STATUS_VSCALEBUSY bit indicates when a voltage scale change is in progress. When the operation is complete the EMU_IF_VSCALEDONEIF flag will be set.

Note: Because SRAM access is blocked during a voltage scaling operation, it is recommended to configure the desired EM0 / EM1 voltage scaling once during initial boot-up for systems operating at VSCALE1.

The current VSCALE setting can be read at any time from the EMU_STATUS_VSCALE field.

11.3.5.2 Voltage Scaling in EM2 and EM3

A separate voltage scaling value is used during EM2 and EM3. This allows the core to run at a higher voltage when in EM0 / EM1 and reduce the voltage in EM2 and EM3 for power savings, or maintain the same voltage for faster wakeup. The voltage scale level for EM2 and EM3 is defined by the EMU_CTRL_EMU23VSCALE field. The new voltage scaling level will be applied when the system is in EM2 or EM3, and return to the EM0 / EM1 voltage scaling level automatically when the system exits the low energy mode.

Hardware will only allow the VSCALE level to remain the same or be reduced when entering EM2 and EM3. If EMU_CTRL_EMU23VSCALE is set to a higher VSCALE setting than the current EM0 / EM1 VSCALE level, the DECOUPLE voltage will remain the same as the EM0 / EM1 setting.

If the voltage scaling level for EM2 / EM3 is lower than the level set for EM0 / EM1, additional time is needed to wake up from the low powered state (see the device data sheet for specific timing). The lowest current during sleep will be obtained by setting EMU23VSCALE to VSCALE0, and the fastest wake times will be obtained when EMU23VSCALE is equal to or higher than the EM0 / EM1 voltage scaling value.

11.3.6 EM0 / EM1 Peripheral Register Retention

When the device enters EM2 or EM3, all peripherals will retain their register configurations by default. Retention for peripherals on the PD1 power domain (i.e. those which do not operate in EM2 and EM3), can optionally be disabled by setting bit 0 of the EMU_PD1PARETCTRL_PD1PARETDIS field. Disabling retention reduces the supply current in EM2 and EM3 slightly. However, the peripheral register interfaces will be reset upon exit to EM0.

Important: This feature is not currently supported by Silicon Labs software stacks. It is the responsibility of the user software to reconfigure any peripherals as necessary when the device wakes to EM0.

11.3.7 Power Configurations

In order to provide the lowest power solutions, the EFM32PG28 comes with a DC-DC module to power internal circuits. The EFM32PG28 may be operated with or without the DC-DC. When used, the DC-DC requires an external inductor and capacitor (refer to the data sheet for recommended values).

The EFM32PG28 has multiple power supply rails: a DC-DC regulator input (VREGVDD), IO Supply (IOVDD), Analog (AVDD), RF Analog Supply (RFVDD), RF Power Amplifier Supply (PAVDD), Digital LDO and flash (DVDD), and Low Voltage Digital Supply (DECOUPLE). Additional detail for each configuration and option is given in the following sections.

Due to on-chip circuitry (e.g., diodes), some power supply pins have a dependent relationship with one or more other power supply pins. These internal relationships between the external voltages applied to the various EFR32 supply pins are defined below. Exceeding the below constraints can result in damage to the device and/or increased current draw.

- VREGVDD >= DVDD

  Note: In systems not using the DC-DC converter, VREGVDD must be shorted to DVDD external to the device.

- PAVDD >= RFVDD
- DVDD >= DECOUPLE
- AVDD, IOVDD: No supply sequencing dependency. Additional leakage may occur if DVDD remains unpowered with power applied to these supplies.

Additionally, there are other system-level considerations when assigning power supplies.

- The usable range for analog signals connected to GPIO (such as IADC inputs) will be limited to the lower of AVDD and IOVDD.
- The RESETn pin has an internal pullup to the DVDD supply. If RESETn is driven by external circuitry above DVDD, additional current may flow into the pin due to this pullup.

11.3.7.1 Power Configuration 0: STARTUP

Upon power-on reset (POR), the system is configured in a safe Startup Configuration that supports all of the available Power Configurations. The Startup Configuration is shown in the simplified diagram below.

In the Startup configuration the DC-DC converter's Bypass switch is ON (i.e., the VREGVDD pin is shorted internally to the DVDD pin).

After power on, firmware can elect to turn on the DC-DC if the external hardware configuration supports it.

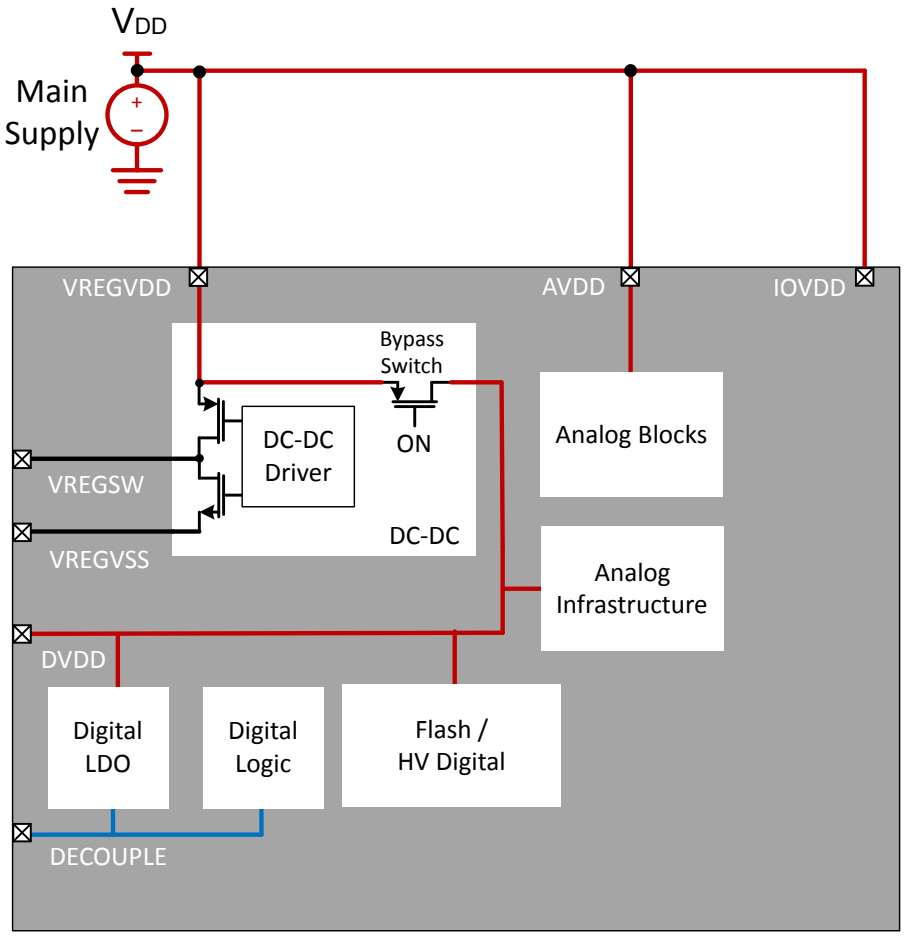


Figure 11.2. Startup Power Configuration

11.3.7.2 Power Configuration 1: No DC-DC

In Power Configuration 1, the DC-DC converter is unused, and all power is supplied by external sources. The DVDD pin must be shorted to VREGVDD.

Other supplies may be supplied by the same supply as VREGIN and DVDD (as shown in 11.3.7.2 Power Configuration 1: No DC-DC), or they may be powered from a separate source.

VREGSW must be left disconnected in this configuration.

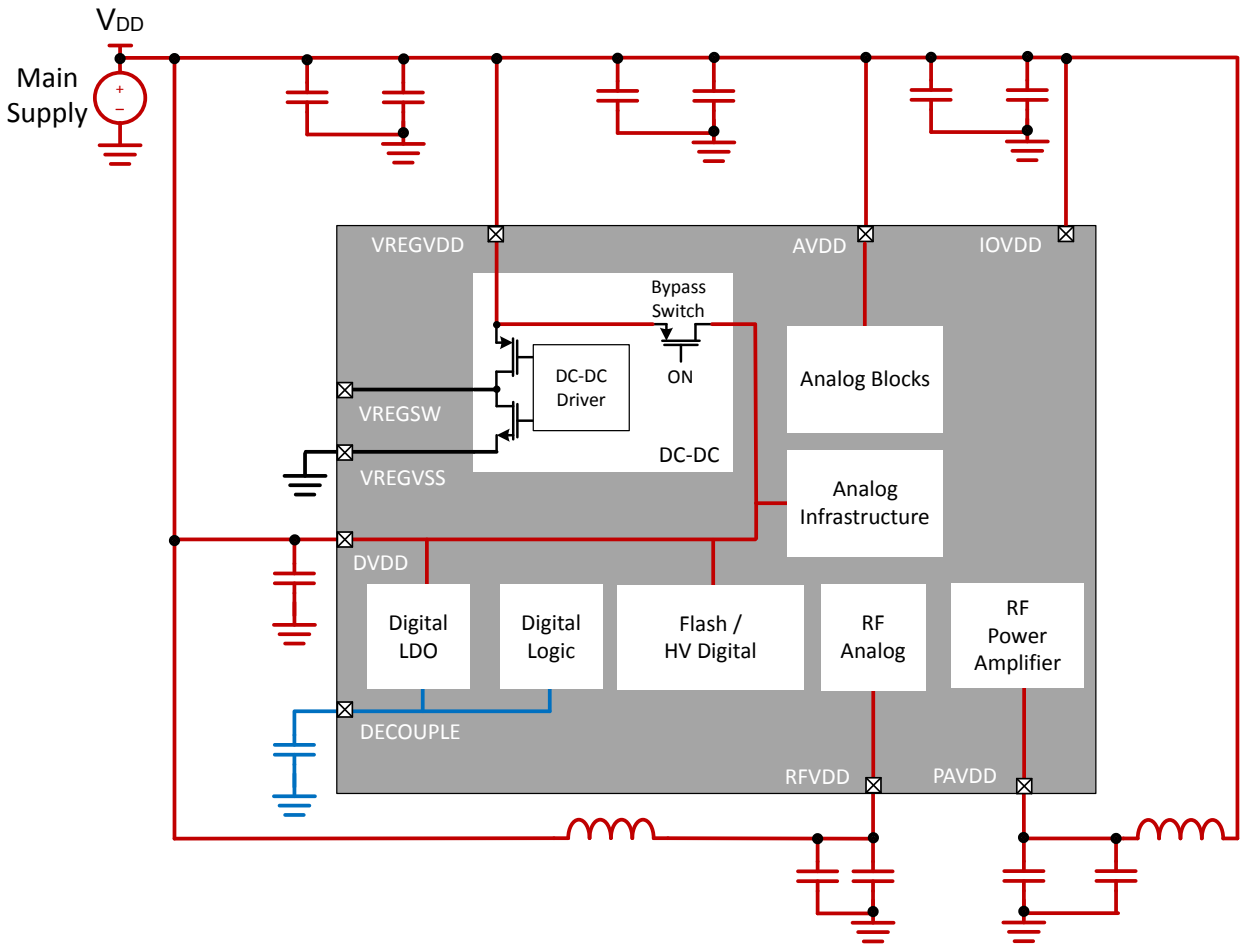


Figure 11.3. DC-DC Off Power Configuration

11.3.7.3 Power Configuration 2: DC-DC

For the lowest power applications, the DC-DC converter can be used to power the rest of the supplies on the device. When the DC-DC converter is used to regulate the voltage at DVDD, the maximum supply voltage may be limited by the operating temperature and/or the average lifetime load conditions. Refer to the device datasheet for additional details.

In Power Configuration 2, the DC-DC Output (V_{DCDC}) is connected to DVDD and optionally, to all the other supplies on the chip. In the configuration shown in Figure 11.4 DC-DC Power Configuration on page 288, the AVDD and IOVDD supplies are connected to the main supply to support higher voltage external interfaces.

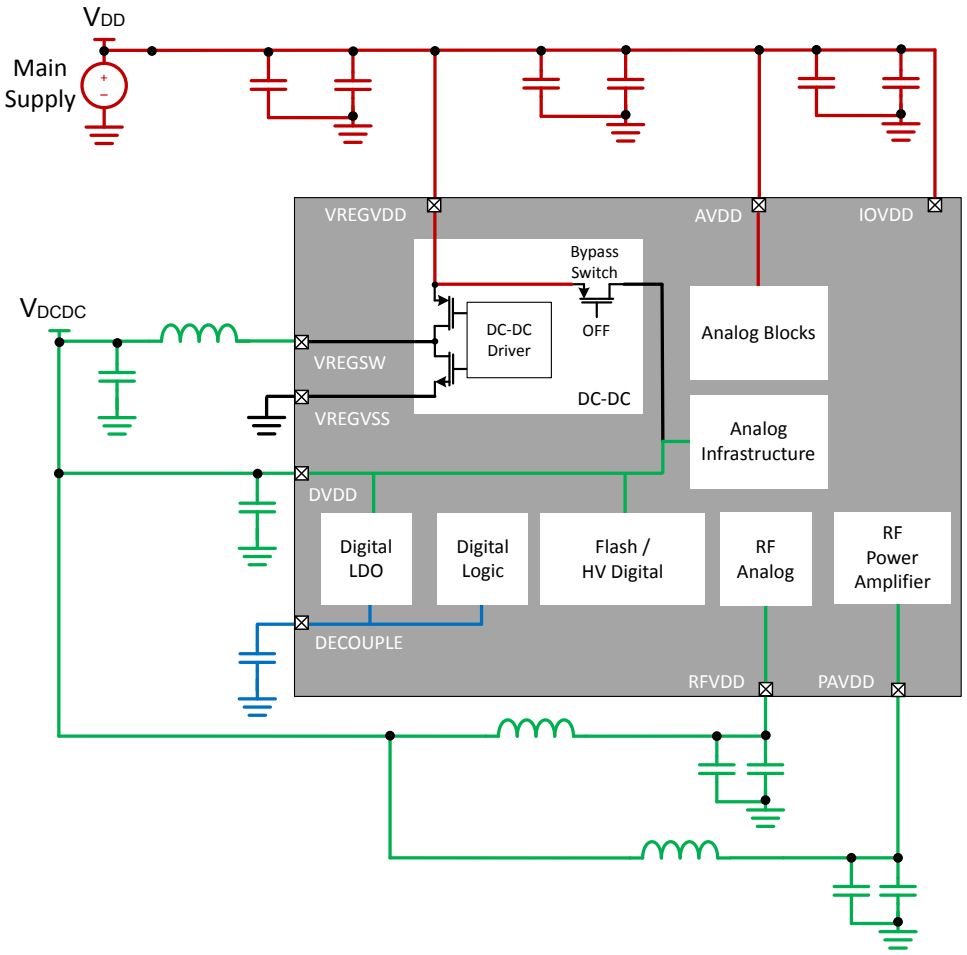


Figure 11.4. DC-DC Power Configuration

As the Main Supply voltage approaches the DC-DC output voltage, it eventually reaches a point where becomes inefficient (or impossible) for the DC-DC module to regulate V_{DCDC}. At this point, firmware can enable bypass mode, which effectively disables the DC-DC and shorts the Main Supply voltage directly to the DC-DC output. If and when sufficient voltage margin on the Main Supply returns, the system can be switched back into DC-DC regulation mode.

11.3.8 Buck DC-DC Interface

The EFM32PG28 devices feature a DC-DC buck converter which requires a single external inductor and a single external capacitor. The input supply is the VREGVDD pin, and the DC-DC converter will produce a nominal 1.8 V output at the DVDD pin to power MCU functions. The DC-DC converter is an efficient PFM (Pulse Frequency Modulation) architecture. In addition, the DC-DC converter supports an unregulated bypass mode, in which the input voltage is directly shorted to the DC-DC output. An integrated programmable supply monitor and dedicated interrupt allows software to enable the bypass switch when the VREGVDD supply voltage is below the minimum allowable voltage for the output current load.

The input supply VREGVDD has a maximum range between 1.8 V and 3.8 V, but is limited by application parameters, including transient current load, operating junction temperature, and the lifetime average current load.

Refer to the device datasheet for more details on the input supply voltage range.

11.3.8.1 Buck DC-DC Mode Bypass and VREGVDD Comparator

The buck DC-DC converter implements a bypass mode which shorts the VREGVDD input voltage directly to the DC-DC converter output through an internal switch. Bypass mode is enabled automatically during a power-on-reset. Bypass mode can also be enabled and disabled through software, using the DCDC_CTRL_MODE field. When set to BYPASS, the bypass switch is enabled and DC-DC regulation will be disabled. Consult the data sheet for the bypass switch impedance specification.

The EFM32PG28 includes a supply comparator circuit to help software determine when the VREGVDD supply is high enough to enable the buck DC-DC, or when to change to bypass mode. The THRESSEL field in the EMU_VREGVDDCMPCTRL register sets the comparator threshold between 2.0 and 2.3 V, and the VREGINCMPEN bit is used to enable the supply comparator. When the VREGVDD comparator is used, DCDC_STATUS_VREGIN can be read by software to determine whether VREGVDD is above or below the established threshold.

The VREGVDD comparator can also generate interrupt events when the input supply is above or below the specified threshold. The VREGINHIGHIEN and VREGINLOWIEN bits in DCDC_IEN are used to enable the above / below threshold interrupts, respectively. The VREGVDD comparator will be active and generate interrupts in EM0 and EM1 only.

The VREGVDD Comparator status is always captured and stored in RMURSTCAUSE.VREGIN on any reset event, even if the reset is not caused by VREGVDD being too low. At startup, the firmware should determine if the last reset was caused by a low VREGVDD condition by checking the following:

EMU_RSTCAUSE_VREGIN & (EMU_RMURSTCAUSE_DVDDBOD | EMU_RMURSTCAUSE_DVDDLEBOD)

If true, the part should remain in bypass mode with the DCDC disabled.

11.3.8.2 Buck DC-DC Startup

Out of power-on-reset (POR), the DC-DC converter defaults to bypass mode and the DC-DC block is disabled. Before enabling the DC-DC, software should first configure and enable the VREGVDD comparator. Once the thresholds for the VREGVDD comparator have been configured and the comparator enabled, the DCDC_STATUS.VREGIN bit should be checked to ensure that the input supply is above the threshold. When the input supply is sufficient, the DC-DC may be configured and enabled. The following steps outline this procedure:

1. Set VREGVDD comparator threshold with EMU_VREGVDDCMPCTRL.THRESSEL
2. Enable VREGVDD comparator with EMU_VREGVDDCMPCTRL.VREGINCMPEN
3. Check DCDC_STATUS.VREGIN:
   - If low, VREGIN is above the programmed threshold and it is safe to enter DC-DC mode
   - If high, VREGIN is below the programmed threshold and firmware should remain in bypass mode
4. Enable the DC-DC module with DCDC_EN_EN = 1
5. Configure the IPKVAL and DRVSPEED settings in DCDC_EM01CTRL0 and DCDC_EM23CTRL0.
6. Enable any required interrupts via DCDC_IEN.
7. Start the DC-DC by setting DCDC_CTRL.MODE to DCDCREGULATION.

The DC-DC will enter a warmup phase for approximately 100 us, then disable the bypass switch and begin using the DC-DC core to regulate the output voltage. The DCDC_IF.RUNNINGIF interrupt flag will indicate when the switch from bypass to DC-DC is complete, however this does not indicate that the output is regulated. Until the output capacitor discharges due to normal current draw from the system, the voltage may be higher than 1.8 V. The DCDC_IF.REGULATIONIF interrupt flag will indicate when the DC-DC has reached regulation and is providing the desired output voltage.

If the VREGINLOWIF interrupt occurs, software should immediately switch back to bypass mode by clearing DCDC_CTRL.MODE to BYPASS.

11.3.8.3 Buck DC-DC Recommended Configuration Settings

Certain DC-DC parameters are adjustable for fine-tuning of performance, but the majority of applications will not need to use any other than the recommended settings. All datasheet parameters are specified using the recommended settings detailed in this section. The configuration settings must be set before DC-DC regulation is started, and must not be changed while the DC-DC is active.

The DCDC_EM01CTRL0 and DCDC_EM23CTRL0 registers each have an IPKVAL field to adjust the maximum peak / load current, and a DRVSPEED field to adjust the driver speed. DCDC_EM01CTRL0 sets the configuration for EM0 and EM1 operation while DCDC_EM23CTRL0 sets the configuration for EM2 and EM3 operation. The DCDC_CTRL.IPKTMAXCTRL field adjusts the maximum time for peak current detection, which impacts the voltage ripple at the DC-DC output. The recommended settings are shown in Table 11.6 DRVSPEED, IPKVAL, and IPKMAXCTRL Recommended Settings for buck DC-DC on page 290.

Table 11.6. DRVSPEED, IPKVAL, and IPKMAXCTRL Recommended Settings for buck DC-DC

Bit Field	Recommended Setting
DCDC_EM01CTRL0.IPKVAL	9 (LOAD60MA)
DCDC_EM01CTRL0.DRVSPEED	1 (DEFAULT_SETTING)
DCDC_EM23CTRL0.IPKVAL	3 (LOAD5MA)
DCDC_EM23CTRL0.DRVSPEED	1 (DEFAULT_SETTING)
DCDC_CTRL.IPKTMAXCTRL	16 (1.19 us)

11.3.8.4 Buck DC-DC EM4 Entry

The buck DC-DC is available in all energy modes except for EM4. If the system wants to enter EM4, the DC-DC converter must first be turned off and switched over to bypass mode. The system will not enter EM4 if the DC-DC is active. If an attempt is made to go into EM4 with DC-DC active, it will be blocked, and the DCDC_IF_EM4ERR flag will be set.

11.3.9 EFP01 Communication

The EFP01 Energy Friendly Power Management IC (PMIC) is an extremely flexible, highly efficient, multi-output power management IC, providing complete system power and primary cell battery Coulomb counting for EFM32PG28 devices. The dual-DCDC converter outputs available on certain EFP01 OPNs can, for example, provide power to both the 1.8 V supplies (e.g., DVDD/AVDD/IOVDD) as well as the 1.1/1.0/0.9 V supply (DECOUPLE) for improved efficiency. EFP01 uses an I2C interface for communciation and also has a unidirectional, open-drain IRQ# output to indicate status flag changes. Consult EFP01 Datasheet for more detailed information and available OPNs.

The EFM32PG28 has additional built-in hardware support for the EFP01 Energy Friendly PMICs, including:

- Direct Mode Energy Mode transition supporting all energy modes (including EM4) on dedicated pins (PC1 / PC2)
- Hardware IRQ with (dedicated IRQ vector) in all energy modes (included EM4) on dedicated pin (PC5)

EFM32PG28's EFP01 hardware support must be enabled by setting one (or both) of the EFPDRVDECOUPLE or the EFPDRVDVDD bits in the EMU_CTRL register:

1. EFPDRVDECOUPLE: Set this bit if EFP01's DCDC output will be powering EFM32PG28's DECOUPLE supply. Once set, EFM32PG28's internal LDOs will be disabled, and any voltage changes (due to voltage-scaling and/or energy mode transitions) will be managed by EFP01. Note that because this bit disables in the internal LDO's powering the core, it should be set until after EFP01's DECOUPLE output has been configured and enabled.
2. EFPDRVDVDD: Set this bit if EFP01's DCDC output is powering EFM32PG28's DVDD supply (or DVDD along with other 1.8V supply inputs). This mode assumes that EFM32PG28's internal DCDC is not being used, so the EFM32PG28 VREGVDD and PAVDD pins should be shorted together on the PCB.

EFM32PG28 provides a dedicated hardware IRQ vector for the EFP01's IRQ output. To use the EFM32PG28's hardware support for EFP01's IRQ output:

1. The PC5 pin should be configured as an input with no pull-up/pull-down enabled and connected on the PCB to EFP01's IRQ pin. Note that although this pin exists on Port C, which typically doesn't support EM2/3 operation, when used as a EFP01 IRQ input the PC5 pin can operate in EM2/3. In addition, the PC5 pin can operate in EM4, without the need to be configured as a EM4Wakeup.
2. EFP01 Hardware support must be enabled by setting either the EFPDRVDECOUPLE or the EFPDRVDVDD bits as described above.

Once enabled, the EFP01 interrupt flag in the EMU_EFPIF register will be set whenever the EFP01 IRQ line goes low. A processor interrupt can be generated to the EMUEFP_IRQHandler() by setting the EFPIEN bit in the EMU_EFPIEN register.

EFM32PG28 includes hardware support for EFP01's optional Direct Mode interface to allow fast-energy mode transitions into and out of all energy modes (EM0/1, EM2/3, EM4). Ordinarilly, I2C transactions are used to manage EFP01's energy mode state - however, a single I2C transaction can take over 100 us. In Direct Mode, the EFM32PG28 retasks the I2C pins as push-pull outputs with pull-ups disabled to control the EFP01's energy mode state directly, allowing much faster energy mode transistions. State definitions are defined in Table 11.7 Direct Mode Energy Mode States on page 291. Because the Direct Mode feature is non-I2C compliant, it should be enabled only during periods when no communciation between EFM32PG28 and EFP01 is required (e.g., an energy mode transition from EM0 to EM2/4), and it is recommended that EFP01 be the only I2C device on the bus. It is also recommended for firmware to wait for the I2C STOP interrupt ensure no I2C transaction is in progress before switching to Direct Mode.

Table 11.7. Direct Mode Energy Mode States

Direct Mode State	I2C SCL Level	I2C SDA Level	Allowed State Transitions
EM0	1	1	• EM2 • I^2C Start Condition
EM2	0	1	• EM0 • EM4
EM4	0	0	• EM2[2]
I^2C Start Condition	1	0	

1 Direct mode transitions between EM0 and EM4 are not allowed. The system must briefly go through the EM2 state on EM4 exit or entrance.

2 Direct mode transitions between EM0 and EM4 are not allowed. The system must briefly go through the EM2 state on EM4 exit or entrance.

To enable Direct Mode:

1. The I2C1 module must be used to communicate with EFP01. I2C0 is not supported.
2. The I2C1_SDA function must be routed to the PC1 pin and connected on the PCB to EFP01's I2C_SDA pin.
3. The I2C1_SCL function must be routed to the PC2 pin and connected on the PCB to EFP01's I2C_SCL pin.
4. Direct Mode must be enabled by setting the EFPDIRECTMODEN bit in the EMU_CTRL register. The EMU will automatically disable the I2C1 internal pull-ups when in Direct Mode
5. EFP01 Hardware support must be enabled by setting either the EFPDRVDECOUPLE or the EFPDRVDVDD bits as described above.

11.3.10 Brown Out Detector (BOD)

Brown out detectors ensure that the minimum supply required for the chip to operate properly and safely is provided to the EFM32PG28. Once triggered, a BOD will generate a system reset.

All BODs detect when the supply falls below a programmed threshold except DECOVMBOD (Over Voltage Monitoring),which detects when the supply goes above a predefined threshold.

All BODs except DVDDBOD and DVDDLEBOD can be individually enabled by firmware.

Table 11.8. EFM32PG28 BODs

BOD	Control Register	Supported Energy Modes	Function
DVDDBOD	n/a	EM0/1	Monitors the DVDD supply in EM0 and EM1. Hardware enables this BOD automatically in EM0/EM1 and disables it in EM2/EM3/EM4
DVDDLEBOD	n/a	EM2/3/4	Low Energy BOD monitors the DVDD supply in EM2/EM3/ EM4. DVDDLEBOD is automatically masked by hardware for ~100us after it is enabled to allow it to settle
DECBOD	EMU_DECBOD	EM0/1/2/3	Monitors the DECOUPLE supply. DECBOD is automatically masked by hardware for ~20us after it is enabled to allow it to settle.
DECOVMBOD	EMU_DECBOD	EM0/1/2/3	Monitors the DECOUPLE supply Over Voltage by detecting DECOUPLE going over a specified threshold. DECOVMBOD is automatically masked by hardware for ~20us after it is enabled to allow it to settle.
AVDDBOD	EMU_BOD3SENSE	EM0/1/2/3/4	Monitors the AVDD supply. Automatically masked by hardware for ~100us after it is enabled to allow it to settle.
IOVDDBOD	EMU_BOD3SENSE	EM0/1/2/3/4	Monitors the IOVDD supply. Automatically masked by hardware for ~100us after it is enabled to allow it to settle. (Note that some devices may have multiple IOVDD supplies.)

11.3.11 Reset Management Unit

EMU RMU (Reset Management Unit) ensures correct reset operation. It is responsible for connecting the different reset sources to the reset lines of the EFM32PG28. After reset, the M33 loads the stack pointer and program entry point from memory and start execution.

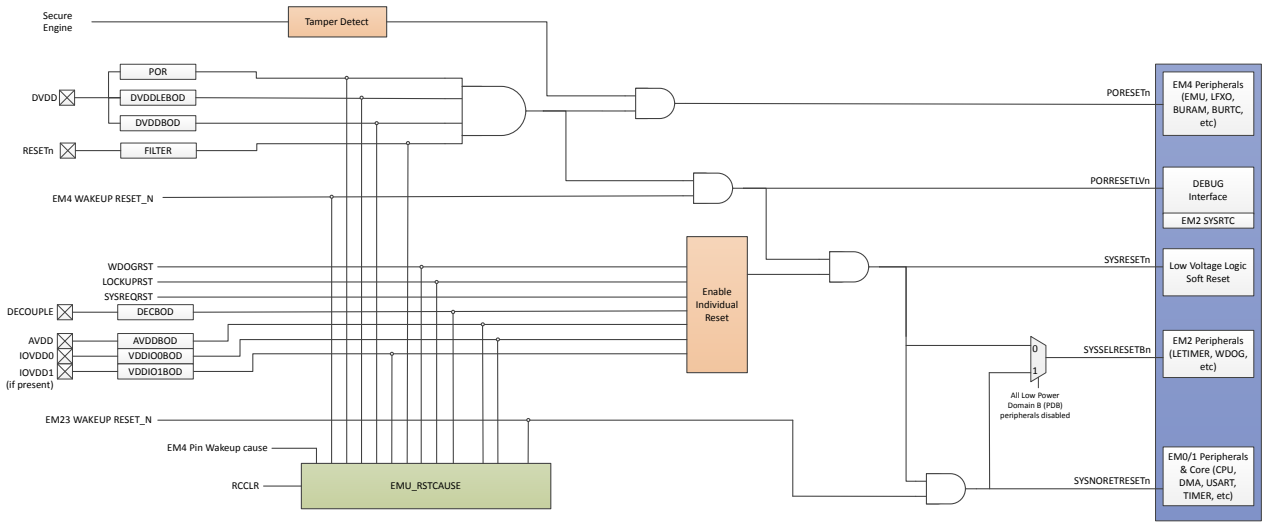


Figure 11.5. Reset Tree

There are two types of reset:

- HARD resets. Resets the entire chip. After a hard reset, the EFM32PG28 goes through its power up sequence. For reset timing specifications, please refer to the device datasheet.
- SOFT resets. Resets only some of the digital low voltage logic. Resets the MCU subsystems and peripherals but doesn't affect digital HV logic (e.g., Power control, BURTC). For reset timing specifications, please refer to the device datasheet.

EFM32PG28 Reset sources

- Power-on Reset (POR)
  - The POR ensures that EFM32PG28 does not start up before the supply voltage DVDD has reached the threshold voltage VPORthr (see Device Datasheet Electrical Characteristics for details). Before the threshold voltage is reached, EFM32PG28 is kept in reset state.
- RESET pin Reset
  - The RESETn pin includes an on-chip pull-up resistor to the DVDD supply, and can therefore be left unconnected if no external reset source is needed. Also connected to the RESETn line is a filter which prevents glitches from resetting the EFM32PG28.
- EM4 wakeup
  - System reset following EM4 exit.
- Watchdog reset
  - The Watchdog circuit is a timer which (when enabled) must be cleared by software regularly. If software does not clear it, a Watchdog reset is activated. This functionality provides recovery from a software stalemate. Refer to the Watchdog section for specifications and description.
- Core lockup condition
  - A MCU lockup is the result of the core being locked up because of an unrecoverable exception following the activation of the processor's built-in system state protection hardware.
- Software triggered reset
  - Software may initiate a reset (e.g. if it finds itself in a non-recoverable state). By asserting the SYSRESETREQ in the Application Interrupt and Reset Control Register, a reset is issued.
- Brown-Out Detection (BOD)
  - EFM32PG28 has multiple built in Brown-out detection (BOD) circuits, which monitor supply voltage level during operation. BOD circuits compare supply voltage to a programmed threshold level and issue a reset request when triggered.
- Secure Engine Tamper detection
  - Secure Engine may issue a system reset request upon tamper detection.

Whether a reset source trigger event lead to a system reset can be controlled via EMU_RMUCTRL register.

EMU_RSTCAUSE register

User can determine the cause of the last reset by querying the EMU_RSTCAUSE register. Once read, EMU_RSTCAUSE should be cleared via EMU_CMD_RCCLR.

Table 11.9. Reset Sources Summary

RSTCAUSE Bit	Name	Type	Can be Disabled?	Description
0	POR	Hard	No	Power On Reset.
1	PIN	Hard	No	Pin Reset.
2	EM4	Soft	No	EM4 Wakeup
3	WDOG0	Soft	Yes	Watchdog 0
4	WDOG1	Soft	Yes	Watchdog 1
5	LOCKUP	Soft	Yes	M33 Lockup
6	SYSREQ	Soft	Yes	M33 Core System Reset
7	DVDDBOD	Hard	No	DVDD BOD
8	DVDDLEBOD	Hard	No	DVDD LEBOD
9	DECBOD	Hard	Yes	DECOUPLE BOD
10	AVDDBOD	Soft	Yes	AVDD BOD
11	IOVDDBOD	Soft	Yes	IOVDD 0 BOD

11.3.12 Temperature Sensor

EMU provides a low energy periodic temperature measurement. A temperature measurement is taken once every 250 ms, with the 9-bit result stored in TEMP bit-field in EMU_TEMP register. The temperature value is expressed in degrees Kelvin. EMU_TEMP_TEMPLSB represents the measured temperature fractional part (in ¼ degree Kelvin).

Note: The EMU temperature sensor is always periodically taking single temperature measurements, except in EM4 (shutoff) mode.

To obtain better noise resolution, the temperature sensor also implements a hardware averaging function, and averaged results can be requested using the EMU_CMD_TEMPAVGREQ command. When TEMPAVGREQ is set by software, the temperature sensor will take 16 or 64 samples as quickly as possible. The TEMPAVGNUM field in EMU_CTRL determines how many temperature measurements will be averaged. The averaged result is stored in the 11-bit field EMU_TEMP_TEMPAVG, which represents the full temperature with resolution of ¼ degree Kelvin.

The EMU provides the following features around temperature changes:

- Interrupt when temperature is updated (EMU_IF_TEMP)
- Interrupt when averaged temperature result is updated (EMU_IF_TEMPAVG)
- Interrupt from LOW level trip (generate interrupt EMU_IF_TEMPLOWIF when measured temperature in EMU_TEMP_TEMP is below programmed threshold EMU_TEMPLIMITS_TEMPLOW)
- Interrupt from HIGH level trip (generate interrupt EMU_IF_TEMPHIGHIF when measured temperature in EMU_TEMP_TEMP is above programmed threshold EMU_TEMPLIMITS_TEMPHI)

High and Low thresholds are specified as 9-bit degree Kelvin values and compared against the single temperature result (EMU_IF_TEMP).

Measured temperature can be converted to degrees Celsius by subtracting 273.15 ($T_{Celsius} = T_{Kelvin} - 273.15$).

11.3.12.1 Linearization, Offset Correction, and Calibration

The raw value reported by the EMU temperature sensor follows a predictable curve. The output may be linearized and the systematic offset removed to achieve die temperature readings with better than +/- 2.5 degrees C accuracy over the full operating temperature range. Further accuracy can be achieved using in-system calibration.

To linearize the measurement and correct for the systematic offset, a second or third-order polynomial equation representing the nominal curve is used. For example, a third-order correction equation takes the form:

$T_{corr} = a*x^3 + b*x^2 + c*x^1 + d$

Where:

- T_{corr} is the corrected temperature (in degrees Celsius)
- x is the measured temperature (in degrees Celsius)
- a is the x^3 term
- b is the x^2 term
- c is the x^1 term
- d is the x^0 term

Polynomial coefficients for both third and second-order polynomials are shown in Table 11.10 Polynomial Coefficients on page 295. Note that the polynomial coefficients provided assume the raw output (in Kelvin) has been converted to Celsius prior to linearization.

Table 11.10. Polynomial Coefficients

Polynomial Order	x^3 Term	x^2 Term	x^1 Term	x^0 Term
Third Order	-1.613E-6	2.001E-5	1.012	-2.894
Second Order	n/a	-2.037E-4	1.014	-2.683

Additional accuracy may be achieved by performing an in-system calibration at known temperatures and operating conditions after linearization.

11.3.13 Register Locks

EMU EMU_LOCK (for user accessible registers) can be used to control access to the EMU_RMUCTRL, EMU_CTRL, and EMU_DECBOD registers. The DCDC_LOCK register can be used to control access to the DC-DC registers.

11.4 EMU Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x010	EMU_DECBOD	RW	DECOUPLE LVBOD Control Register
0x020	EMU_BOD3SENSE	RW	BOD3SENSE Control Register
0x03C	EMU_VREGVDDCMPCTRL	RW	DC-DC VREGVDD Comparator Control Register
0x040	EMU_PD1PARETCTRL	RW	PD1 Partial Retention Control
0x05C	EMU_IPVERSION	R	IP Version
0x060	EMU_LOCK	W	EMU Configuration Lock Register
0x064	EMU_IF	RWH INTFLAG	Interrupt Flags
0x068	EMU_IEN	RW	Interrupt Enables
0x06C	EMU_EM4CTRL	RW	EM4 Control
0x070	EMU_CMD	W	EMU Command Register
0x074	EMU_CTRL	RW	EMU Control Register
0x078	EMU_TEMPLIMITS	RW	EMU Temperature Thresholds
0x084	EMU_STATUS	RH	EMU Status Register
0x088	EMU_TEMP	RH	Temperature
0x090	EMU_RSTCTRL	RW	Reset Management Control Register
0x094	EMU_RSTCAUSE	RH	Reset Cause
0x098	EMU_TAMPERRSTCAUSE	RH	Tamper Reset Cause
0x0A0	EMU_DGIF	RWH INTFLAG	Interrupt Flags Debug
0x0A4	EMU_DGIEN	RW	Interrupt Enables Debug
0x100	EMU_EFPIF	RWH INTFLAG	EFP Interrupt Register
0x104	EMU_EFPIEN	RW	EFP Interrupt Enable Register
0x1010	EMU_DECBOD_SET	RW	DECOUPLE LVBOD Control Register
0x1020	EMU_BOD3SENSE_SET	RW	BOD3SENSE Control Register
0x103C	EMU_VREGVDDCMPCTRL_SET	RW	DC-DC VREGVDD Comparator Control Register
0x1040	EMU_PD1PARETCTRL_SET	RW	PD1 Partial Retention Control
0x105C	EMU_IPVERSION_SET	R	IP Version
0x1060	EMU_LOCK_SET	W	EMU Configuration Lock Register
0x1064	EMU_IF_SET	RWH INTFLAG	Interrupt Flags
0x1068	EMU_IEN_SET	RW	Interrupt Enables
0x106C	EMU_EM4CTRL_SET	RW	EM4 Control
0x1070	EMU_CMD_SET	W	EMU Command Register
0x1074	EMU_CTRL_SET	RW	EMU Control Register
0x1078	EMU_TEMPLIMITS_SET	RW	EMU Temperature Thresholds
0x1084	EMU_STATUS_SET	RH	EMU Status Register

Offset	Name	Type	Description
0x1088	EMU_TEMP_SET	RH	Temperature
0x1090	EMU_RSTCTRL_SET	RW	Reset Management Control Register
0x1094	EMU_RSTCAUSE_SET	RH	Reset Cause
0x1098	EMU_TAMPERRSTCAUSE_SET	RH	Tamper Reset Cause
0x10A0	EMU_DGIF_SET	RWH INTFLAG	Interrupt Flags Debug
0x10A4	EMU_DGIEN_SET	RW	Interrupt Enables Debug
0x1100	EMU_EFPIF_SET	RWH INTFLAG	EFP Interrupt Register
0x1104	EMU_EFPIEN_SET	RW	EFP Interrupt Enable Register
0x2010	EMU_DECBOD_CLR	RW	DECOUPLE LVBOD Control Register
0x2020	EMU_BOD3SENSE_CLR	RW	BOD3SENSE Control Register
0x203C	EMU_VREGVDDCMPCTRL_CLR	RW	DC-DC VREGVDD Comparator Control Register
0x2040	EMU_PD1PARETCTRL_CLR	RW	PD1 Partial Retention Control
0x205C	EMU_IPVERSION_CLR	R	IP Version
0x2060	EMU_LOCK_CLR	W	EMU Configuration Lock Register
0x2064	EMU_IF_CLR	RWH INTFLAG	Interrupt Flags
0x2068	EMU_IEN_CLR	RW	Interrupt Enables
0x206C	EMU_EM4CTRL_CLR	RW	EM4 Control
0x2070	EMU_CMD_CLR	W	EMU Command Register
0x2074	EMU_CTRL_CLR	RW	EMU Control Register
0x2078	EMU_TEMPLIMITS_CLR	RW	EMU Temperature Thresholds
0x2084	EMU_STATUS_CLR	RH	EMU Status Register
0x2088	EMU_TEMP_CLR	RH	Temperature
0x2090	EMU_RSTCTRL_CLR	RW	Reset Management Control Register
0x2094	EMU_RSTCAUSE_CLR	RH	Reset Cause
0x2098	EMU_TAMPER-RSTCAUSE_CLR	RH	Tamper Reset Cause
0x20A0	EMU_DGIF_CLR	RWH INTFLAG	Interrupt Flags Debug
0x20A4	EMU_DGIEN_CLR	RW	Interrupt Enables Debug
0x2100	EMU_EFPIF_CLR	RWH INTFLAG	EFP Interrupt Register
0x2104	EMU_EFPIEN_CLR	RW	EFP Interrupt Enable Register
0x3010	EMU_DECBOD_TGL	RW	DECOUPLE LVBOD Control Register
0x3020	EMU_BOD3SENSE_TGL	RW	BOD3SENSE Control Register
0x303C	EMU_VREGVDDCMPCTRL_TGL	RW	DC-DC VREGVDD Comparator Control Register
0x3040	EMU_PD1PARETCTRL_TGL	RW	PD1 Partial Retention Control
0x305C	EMU_IPVERSION_TGL	R	IP Version
0x3060	EMU_LOCK_TGL	W	EMU Configuration Lock Register

Offset	Name	Type	Description
0x3064	EMU_IF_TGL	RWH INTFLAG	Interrupt Flags
0x3068	EMU_IEN_TGL	RW	Interrupt Enables
0x306C	EMU_EM4CTRL_TGL	RW	EM4 Control
0x3070	EMU_CMD_TGL	W	EMU Command Register
0x3074	EMU_CTRL_TGL	RW	EMU Control Register
0x3078	EMU_TEMPLIMITS_TGL	RW	EMU Temperature Thresholds
0x3084	EMU_STATUS_TGL	RH	EMU Status Register
0x3088	EMU_TEMP_TGL	RH	Temperature
0x3090	EMU_RSTCTRL_TGL	RW	Reset Management Control Register
0x3094	EMU_RSTCAUSE_TGL	RH	Reset Cause
0x3098	EMU_TAMPERRSTCAUSE_TGL	RH	Tamper Reset Cause
0x30A0	EMU_DGIF_TGL	RWH INTFLAG	Interrupt Flags Debug
0x30A4	EMU_DGIEN_TGL	RW	Interrupt Enables Debug
0x3100	EMU_EFPIF_TGL	RWH INTFLAG	EFP Interrupt Register
0x3104	EMU_EFPIEN_TGL	RW	EFP Interrupt Enable Register

11.5 EMU Register Description

11.5.1 EMU_DECBOD - DECOUPLE LVBOD Control Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x1	0x0			0x1	0x0
Access																											RW	RW			RW	RW
Name																											DECOVMBODMASK	DECOVMBODEN			DECBODMASK	DECBODEN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	DECOVMBODMASK	0x1	RW	**Over Voltage Monitor Mask**
	DECOUPLE BOD Over Voltage Monitor Mask			
4	DECOVMBODEN	0x0	RW	**Over Voltage Monitor enable**
	DECOUPLE BOD Over Voltage Monitor enable. Enables LVBOD below vref high. BOD is masked for 20us after enable			
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DECBODMASK	0x1	RW	**DECBOD Mask**
	DECOUPLE BOD Mask			
0	DECBODEN	0x0	RW	**DECBOD enable**
	DECOUPLE BOD enable. Enables LVBOD above vref low. BOD is masked for 20us after enable			

11.5.2 EMU_BOD3SENSE - BOD3SENSE Control Register

Offset	Bit Position																														
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														VDDIO1BODEN	VDDIO0BODEN	AVDDBODEN

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	VDDIO1BODEN	0x0	RW	**VDDIO1 BOD enable**
	BOD output is automatically masked for 100us by HW after enable is set			
1	VDDIO0BODEN	0x0	RW	**VDDIO0 BOD enable**
	BOD output is automatically masked for 100us by HW after enable is set			
0	AVDDBODEN	0x0	RW	**AVDD BOD enable**
	BOD output is automatically masked for 100us by HW after enable is set			

11.5.3 EMU_VREGVDDCMPCTRL - DC-DC VREGVDD Comparator Control Register

Offset	Bit Position																														
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x3		0x0
Access																														RW		RW
Name																														THRESSEL		VREGINCMPEN

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:1	THRESSEL	0x3	RW	**VREGVDD comparator threshold programming**
	VREGVDD comparator threshold programming: 2.0->2.3V, 0.1V/step			
0	VREGINCMPEN	0x0	RW	**VREGVDD comparator enable**
	VREGVDD comparator enable. Output is masked for 5us after enabled. Automatically disabled in EM2.			

11.5.4 EMU_PD1PARETCTRL - PD1 Partial Retention Control

Offset	Bit Position																													
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									PD1PARETDIS							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	PD1PARETDIS	0x0	RW	**Disable PD1 Partial Retention**

Select PD1 register groups that are NOT retained in EM2/EM3. Each bit controls a register group. MCU core group is always retained. Bit[0]: Disables PD1 retention for MCU Peripherals group. Bit[1]: Disables PD1 retention for RADIO group (only on devices with a radio). Bit [15:2]: Unused. Setting PD1 retention for MCU Peripherals group will also allow PD0B/C/D power domains to be turned OFF in EM23 if all peripherals on those power domains are turned off on EM23 entry

Value	Mode	Description
1	PERIPHNORETAIN	Retain associated registers when in EM2/3
2	RADIONORETAIN	Bit[1]. When set, do not retain RADIO associated registers when in EM2/3

11.5.5 EMU_IPVERSION - IP Version

Offset	Bit Position																													
0x05C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x7																
Access																R																
Name																IPVERSION																

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x7	R	**IP Version**
	IP Version			

11.5.6 EMU_LOCK - EMU Configuration Lock Register

Offset	Bit Position																													
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0xADE8								
Access																								W								
Name																								LOCKKEY								

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0xADE8	W	**Lock Key**

Write any other value than the unlock code to lock

Value	Mode	Description
44520	UNLOCK	Unlock EMU register

11.5.7 EMU_IF - Interrupt Flags

Offset	Bit Position																													
0x064	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0		0x0		0x0	0x0							0x0	0x0																
Access	RW	RW	RW		RW		RW	RW							RW	RW																
Name	TEMPHIGH	TEMPLOW	TEMP		TEMPAVG		VSCALEDONE	EM23WAKEUP							IOVDD0BOD	AVDDBOD																

Bit	Name	Reset	Access	Description
31	TEMPHIGH	0x0	RW	**Temperature high Interrupt flag**
	Measured temperature above threshold			
30	TEMPLOW	0x0	RW	**Temperature low Interrupt flag**
	Measured temperature below threshold			
29	TEMP	0x0	RW	**Temperature Interrupt flag**
	Temperature Update			
28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27	TEMPAVG	0x0	RW	**Temperature Average Interrupt flag**
	Averaged Temperature Update			
26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25	VSCALEDONE	0x0	RW	**Vscale done Interrupt flag**
	Voltage scaling completed. EM0 only.			
24	EM23WAKEUP	0x0	RW	**EM23 Wake up Interrupt flag**
	EM23 wake up			
23:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	IOVDD0BOD	0x0	RW	**VDDIO0 BOD Interrupt flag**
	IOVDD0 BOD triggered			
16	AVDDBOD	0x0	RW	**AVDD BOD Interrupt flag**
	AVDD BOD triggered			
15:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

11.5.8 EMU_IEN - Interrupt Enables

Offset	Bit Position																													
0x068	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0		0x0		0x0	0x0							0x0	0x0																
Access	RW	RW	RW		RW		RW	RW							RW	RW																
Name	TEMPHIGH	TEMPLOW	TEMP		TEMPAVG		VSCALEDONE	EM23WAKEUP							IOVDD0BOD	AVDDBOD																

Bit	Name	Reset	Access	Description
31	TEMPHIGH	0x0	RW	**Temperature high Interrupt enable**
	Measured temperature above threshold Interrupt enable			
30	TEMPLOW	0x0	RW	**Temperature low Interrupt enable**
	Measured temperature below threshold Interrupt enable			
29	TEMP	0x0	RW	**Temperature Interrupt enable**
	Temperature Update Interrupt enable			
28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27	TEMPAVG	0x0	RW	**Temperature Interrupt enable**
	Averaged Temperature Interrupt enable			
26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25	VSCALEDONE	0x0	RW	**Vscale done Interrupt enable**
	Voltage scaling completed Interrupt enable. EM0 only.			
24	EM23WAKEUP	0x0	RW	**EM23 Wake up Interrupt enable**
	EM23 wake up Interrupt enable			
23:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	IOVDD0BOD	0x0	RW	**VDDIO0 BOD Interrupt enable**
	IOVDD0 BOD Interrupt enable			
16	AVDDBOD	0x0	RW	**AVDD BOD Interrupt enable**
	AVDD BOD Interrupt enable			
15:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

11.5.9 EMU_EM4CTRL - EM4 Control

Offset	Bit Position																													
0x06C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0			0x0				0x0	
Access																								RW			RW				RW	
Name																								BOD3SENSEEM4WU			EM4IORETMODE				EM4ENTRY	

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	BOD3SENSEEM4WU	0x0	RW	**Set BOD3SENSE as EM4 wakeup** Enable BOD3SENSE as EM4 wakeup source
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	EM4IORETMODE	0x0	RW	**EM4 IO retention mode** Determine when IO retention will be applied and removed

Value	Mode	Description
0	DISABLE	No Retention: Pads enter reset state when entering EM4
1	EM4EXIT	Retention through EM4: Pads enter reset state when exiting EM4
2	SWUNLATCH	Retention through EM4 and Wakeup: software writes UNLATCH register to remove retention

3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	EM4ENTRY	0x0	RW	**EM4 entry request** This field is used to enter the Energy Mode 4 sequence. Writing the sequence 2,3,2,3,2,3,2,3,2 will enter the part into Energy Mode 4

11.5.10 EMU_CMD - EMU Command Register

Offset	Bit Position																														
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset														0x0	0x0						0x0	0x0						0x0			0x0	
Access														W(nB)	W(nB)						W(nB)	W(nB)						W(nB)			W(nB)	
Name														TAMPERRCCLR	RSTCAUSECLR						EM01VSCALE2	EM01VSCALE1						TEMPAVGREQ			EM4UNLATCH	

Bit	Name	Reset	Access	Description
31:19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18	TAMPERRCCLR	0x0	W(nB)	**Tamper Reset Cause Clear**
	Set this bit to clear the TAMPERRSTCAUSE register. Root access only			
17	RSTCAUSECLR	0x0	W(nB)	**Reset Cause Clear**
	Set this bit to clear the RMURSTCAUSE register			
16:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	EM01VSCALE2	0x0	W(nB)	**Scale voltage to Vscale2**
	EM01 Voltage Scale Command to scale to Voltage Scale Level 2			
10	EM01VSCALE1	0x0	W(nB)	**Scale voltage to Vscale1**
	EM01 Voltage Scale Command to scale to Voltage Scale Level 1			
9:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	TEMPAVGREQ	0x0	W(nB)	**Temperature Average Request**
	Request for Averaged Temperature Measurement			
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	EM4UNLATCH	0x0	W(nB)	**EM4 unlatch**
	GPIO unlatch request after EM4 wakeup. Only valid when EM4IORETMODE== SWUNLATCH			
0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

11.5.11 EMU_CTRL - EMU Control Register

Offset	Bit Position																														
0x074	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0													0x0							0x2						0x0			0x0
Access	RW	RW	RW													RW							RW						RW			RW
Name	EFPDRVDVDD	EFPDRVDECOUPLE	EFPDIRECTMODEEN													FLASHPWRUPONDEMAND							EM23VSCALE						TEMPAVGNUM			EM2DBGEN

Bit	Name	Reset	Access	Description
31	EFPDRVDVDD	0x0	RW	**EFP drives DVDD**
	EFP01 Drives DVDD. EFP IRQ is enabled on PC5. VREGVDD and DVDD pins should be shorted together on the PCB.			
30	EFPDRVDECOUPLE	0x0	RW	**EFP drives DECOUPLE**
	EFP Drives DECOUPLE. EMU voltage scaling is done through EFP			
29	EFPDIRECTMODEEN	0x0	RW	**EFP Direct Mode Enable**
	EFP01 Direct mode enable. EMU drive I2C lines to transition EFP01 between energy modes. Firmware must use I2C1 module with SDA routed to PC1 and SCL routed to SC2.			
28:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	FLASHPWRUPONDEMAND	0x0	RW	**Enable flash on demand wakeup**
	When set, during wake up, Flash will be in power down mode until either incoming Flash data fetch or when software issue powerup command to IMEM->MSC_CMD register			
15:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	EM23VSCALE	0x2	RW	**EM2/EM3 Vscale**
	Set VSCALE value for EM2/EM3 mode			

Value	Mode	Description
0	VSCALE0	VSCALE0. 0.9v
1	VSCALE1	VSCALE1. 1.0v
2	VSCALE2	VSCALE2. 1.1v

7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	TEMPAVGNUM	0x0	RW	**Averaged Temperature samples num**
	Number of samples taken for Averaged Temperature Measurement			

Value	Mode	Description

Bit	Name	Reset	Access	Description
	0	N16		16 measurements
	1	N64		64 measurements
2:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	EM2DBGEN	0x0	RW	**Enable debugging in EM2**
	Force debug power domain to stay on on EM2 entry. This allows debugger to remain connected in EM2.			

11.5.12 EMU_TEMPLIMITS - EMU Temperature Thresholds

Offset	Bit Position																													
0x078	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset												0x1FF																0x0				
Access												RW																RW				
Name												TEMPHIGH																TEMPLOW				

Bit	Name	Reset	Access	Description
31:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24:16	TEMPHIGH	0x1FF	RW	**Temp High limit**
	Temp threshold in degree Kelvin. The TEMPHIGH interrupt flag is set when a periodic temperature measurement is equal to or higher than this value.			
15:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	TEMPLOW	0x0	RW	**Temp Low limit**
	Temp threshold in degree Kelvin. The TEMPLOW interrupt flag is set when a periodic temperature measurement is equal to or lower than this value.			

11.5.13 EMU_STATUS - EMU Status Register

Offset	Bit Position																													
0x084	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																		0x0		0x0					0x2		0x0	0x0	0x0	0x0	0x0	0x0
Access																		R		R					R		R	R	R	R	R	R
Name																		EM2ENTERED		EM4IORET					VSCALE		VSCALEFAILED	VSCALEBUSY	TEMPAVGACTIVE	TEMPACTIVE	FIRSTTEMPDONE	LOCK

Bit	Name	Reset	Access	Description
31:15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14	EM2ENTERED	0x0	R	**EM2 entered**
	Confirm chip entered EM2 state. EM2 Entry request can be delayed or denied by peripherals.			
13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	EM4IORET	0x0	R	**EM4 IO retention status**
	The status of IO retention. Will be set upon EM4 entry based on EM4IORETMODE in EMU_EM4CTRL. Cleared by setting EM4UNLATCH in EMU_CMD			
11:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:6	VSCALE	0x2	R	**Vscale status**
	Current Voltage Scale Value			

Value	Mode	Description
0	VSCALE0	Voltage scaling set to 0.9v
1	VSCALE1	Voltage scaling set to 1.0v
2	VSCALE2	Voltage scaling set to 1.1v

5	VSCALEFAILED	0x0	R	**Vscale failed**
	Voltage scaling failed. (Time out)			
4	VSCALEBUSY	0x0	R	**Vscale busy**
	Voltage Scaling busy			
3	TEMPAVGACTIVE	0x0	R	**Temp Average active**
	Average Temperature Measurement active			
2	TEMPACTIVE	0x0	R	**Temp active**
	Temperature Measurement active			
1	FIRSTTEMPDONE	0x0	R	**First Temp done**
	First Temperatue mesaurement completed			

Bit	Name	Reset	Access	Description
0	LOCK	0x0	R	**Lock status**
	Indicates the current status of EMU Lock			

Value	Mode	Description
0	UNLOCKED	All EMU lockable registers are unlocked.
1	LOCKED	All EMU lockable registers are locked.

11.5.14 EMU_TEMP - Temperature

Offset	Bit Position																													
0x088	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0															0x0					0x0	
Access											R															R					R	
Name											TEMPAVG															TEMP					TEMPLSB	

Bit	Name	Reset	Access	Description
31:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26:16	TEMPAVG	0x0	R	**Averaged Temperature**
	Averaged Temperature Measurement. Temperature in Kelvin. 9 integer bits and 2 decimal bits (0.25 Degree resolution)			
15:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10:2	TEMP	0x0	R	**Temperature measured**
	Temperature in Kelvin. Value of last periodic temperature measurement. Value is asynchronously updated.			
1:0	TEMPLSB	0x0	R	**Temperature measured decimal part**
	Temperature decimal part			

11.5.15 EMU_RSTCTRL - Reset Management Control Register

Offset	Bit Position																												
0x090	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																						0x1			0x0	0x0			0x0	0x1		0x1
Access																						RW			RW	RW			RW	RW		RW
Name																						DECBODRMODE			IOVDD0BODRMODE	AVDDBODRMODE			LOCKUPRMODE	SYSRMODE		WDOG0RMODE

Bit	Name	Reset	Access	Description
31:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	DECBODRMODE	0x1	RW	**Enable DECBOD reset**

LVBOD Reset Mode. DECOUPLE monitoring. BOD must be trimmed before it is used as a reset source.

Value	Mode	Description
0	DISABLED	Reset request is blocked
1	ENABLED	The entire device is reset

Bit	Name	Reset	Access	Description
9:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	IOVDD0BODRMODE	0x0	RW	**Enable VDDIO0 BOD reset**

LEBOD2 Reset Mode. IOVDD0 monitoring. BOD must be trimmed before it is used as a reset source.

Value	Mode	Description
0	DISABLED	Reset request is blocked
1	ENABLED	The entire device is reset except some EMU registers

Bit	Name	Reset	Access	Description
6	AVDDBODRMODE	0x0	RW	**Enable AVDD BOD reset**

LEBOD1 Reset Mode. AVDD monitoring. BOD must be trimmed before it is used as a reset source.

Value	Mode	Description
0	DISABLED	Reset Request is block
1	ENABLED	The entire device is reset except some EMU registers

Bit	Name	Reset	Access	Description
5:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	LOCKUPRMODE	0x0	RW	**Enable M33 Lockup reset**

Core LOCKUP Reset Mode

Value	Mode	Description
0	DISABLED	Reset Request is Block

Bit	Name	Reset	Access	Description
	1	ENABLED		The entire device is reset except some EMU registers
2	SYSRMODE	0x1	RW	**Enable M33 System reset**
	Core Sysreset Reset Mode			
	Value	Mode		Description
	0	DISABLED		Reset request is blocked
	1	ENABLED		Device is reset except some EMU registers
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	WDOG0RMODE	0x1	RW	**Enable WDOG0 reset**
	WDOG0 Reset Mode			
	Value	Mode		Description
	0	DISABLED		Reset request is blocked
	1	ENABLED		The entire device is reset except some EMU registers

11.5.16 EMU_RSTCAUSE - Reset Cause

Offset	Bit Position																													
0x094	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																		0x0		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access	R																		R		R	R	R	R	R	R	R	R	R	R	R	R
Name	VREGIN																		SETAMPER		IOVDD0BOD	AVDDBOD	DECBOD	DVDDLEBOD	DVDDBOD	SYSREQ	LOCKUP	WDOG1	WDOG0	EM4	PIN	POR

Bit	Name	Reset	Access	Description
31	VREGIN	0x0	R	**DCDC VREGIN comparator**
	DCDC VREGIN comparator below threshold. For Information only, not a direct source for reset. Should be used to determine whether the previous reset was caused by DCDC input being too low to support current load. In this case it is advised to keep the chip in BYPASS mode and check battery level before re-enabling integrated DCDC			
30:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	SETAMPER	0x0	R	**SE Tamper event Reset**
	Last reset was a SE Tamper event reset			
12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	IOVDD0BOD	0x0	R	**LEBOD2 Reset**
	Brown Out Detector monitoring IOVDD0			
10	AVDDBOD	0x0	R	**LEBOD1 Reset**
	Brown Out Detector monitoring AVDD			
9	DECBOD	0x0	R	**LVBOD Reset**
	Brown Out Detector monitoring DECOUPLE			
8	DVDDLEBOD	0x0	R	**LEBOD Reset**
	Brown Out Detector monitoring DVDD in EM2/3			
7	DVDDBOD	0x0	R	**HVBOD Reset**
	Brown Out Detector monitoring DVDD in EM0/1			
6	SYSREQ	0x0	R	**M33 Core Sys Reset**
	Last Reset was as M33 Core System reset			
5	LOCKUP	0x0	R	**M33 Core Lockup Reset**
	Last Reset was as M33 Core Lockup reset			
4	WDOG1	0x0	R	**Watchdog 1 Reset**
	Last reset was a Watchdog 1 reset			
3	WDOG0	0x0	R	**Watchdog 0 Reset**
	Last reset was a Watchdog 0 reset			
2	EM4	0x0	R	**EM4 Wakeup Reset**

Bit	Name	Reset	Access	Description
	Last reset was a EM4 Wakeup			
1	PIN	0x0	R	**Pin Reset**
	Last reset was a Pin reset			
0	POR	0x0	R	**Power On Reset**
	Last reset was a Power On Reset			

11.5.17 EMU_TAMPERRSTCAUSE - Tamper Reset Cause

Offset	Bit Position																															
0x098	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	TAMPERRST																															

Bit	Name	Reset	Access	Description
31:0	TAMPERRST	0x0	R	**Tamper reset vector**
	Tamper reset vector. Reset cause indicator defining which tamper response index triggered the previous tamper reset. Cleared with TAMPERRCCLR			

11.5.18 EMU_DGIF - Interrupt Flags Debug

Offset	Bit Position																															
0x0A0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0					0x0																								
Access	RW	RW	RW					RW																								
Name	TEMPHIGHDGIF	TEMPLOWDGIF	TEMPDGIF					EM23WAKEUPDGIF																								

Bit	Name	Reset	Access	Description
31	TEMPHIGHDGIF	0x0	RW	**Temperature high Interrupt flag**
	Measured temperature above threshold			
30	TEMPLOWDGIF	0x0	RW	**Temperature low Interrupt flag**
	Measured temperature below threshold			
29	TEMPDGIF	0x0	RW	**Temperature Interrupt flag**
	Temperature Update			
28:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24	EM23WAKEUPDGIF	0x0	RW	**EM23 Wake up Interrupt flag**
	EM23 wake up			
23:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

11.5.19 EMU_DGIEN - Interrupt Enables Debug

Offset	Bit Position																													
0x0A4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0					0x0																								
Access	RW	RW	RW					RW																								
Name	TEMPHIGHDGIEN	TEMPLOWDGIEN	TEMPDGIEN					EM23WAKEUPDGIEN																								

Bit	Name	Reset	Access	Description
31	TEMPHIGHDGIEN	0x0	RW	**Temperature high Interrupt enable**
	Measured temperature above threshold			
30	TEMPLOWDGIEN	0x0	RW	**Temperature low Interrupt enable**
	Measured temperature below threshold			
29	TEMPDGIEN	0x0	RW	**Temperature Interrupt enable**
	Temperature Update			
28:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24	EM23WAKEUPDGIEN	0x0	RW	**EM23 Wake up Interrupt enable**
	EM23 wake up			
23:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

11.5.20 EMU_EFPIF - EFP Interrupt Register

Offset	Bit Position																													
0x100	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																EFPIF

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	EFPIF	0x0	RW	**EFP Interrupt Flag**
	EFP Interrupt			

11.5.21 EMU_EFPIEN - EFP Interrupt Enable Register

Offset	Bit Position																															
0x104	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																EFPIEN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	EFPIEN Enable EFP Interrupt	0x0	RW	**EFP Interrupt enable**

11.6 DCDC Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	DCDC_IPVERSION	R	IPVERSION
0x004	DCDC_CTRL	RW SYNC	Control
0x008	DCDC_EM01CTRL0	RW SYNC	EM01 Control
0x010	DCDC_EM23CTRL0	RW SYNC	EM23 Control
0x020	DCDC_PFMXCTRL	RW SYNC	PFMX Control Register
0x028	DCDC_IF	RWH INTFLAG	Interrupt Flags
0x02C	DCDC_IEN	RW	Interrupt Enable
0x030	DCDC_STATUS	RH	Status Register
0x034	DCDC_SYNCBUSY	RH	Syncbusy Status Register
0x040	DCDC_LOCK	W	Lock Register
0x044	DCDC_LOCKSTATUS	RH	Lock Status Register
0x1000	DCDC_IPVERSION_SET	R	IPVERSION
0x1004	DCDC_CTRL_SET	RW SYNC	Control
0x1008	DCDC_EM01CTRL0_SET	RW SYNC	EM01 Control
0x1010	DCDC_EM23CTRL0_SET	RW SYNC	EM23 Control
0x1020	DCDC_PFMXCTRL_SET	RW SYNC	PFMX Control Register
0x1028	DCDC_IF_SET	RWH INTFLAG	Interrupt Flags
0x102C	DCDC_IEN_SET	RW	Interrupt Enable
0x1030	DCDC_STATUS_SET	RH	Status Register
0x1034	DCDC_SYNCBUSY_SET	RH	Syncbusy Status Register
0x1040	DCDC_LOCK_SET	W	Lock Register
0x1044	DCDC_LOCKSTATUS_SET	RH	Lock Status Register
0x2000	DCDC_IPVERSION_CLR	R	IPVERSION
0x2004	DCDC_CTRL_CLR	RW SYNC	Control
0x2008	DCDC_EM01CTRL0_CLR	RW SYNC	EM01 Control
0x2010	DCDC_EM23CTRL0_CLR	RW SYNC	EM23 Control
0x2020	DCDC_PFMXCTRL_CLR	RW SYNC	PFMX Control Register
0x2028	DCDC_IF_CLR	RWH INTFLAG	Interrupt Flags
0x202C	DCDC_IEN_CLR	RW	Interrupt Enable
0x2030	DCDC_STATUS_CLR	RH	Status Register
0x2034	DCDC_SYNCBUSY_CLR	RH	Syncbusy Status Register
0x2040	DCDC_LOCK_CLR	W	Lock Register
0x2044	DCDC_LOCKSTATUS_CLR	RH	Lock Status Register
0x3000	DCDC_IPVERSION_TGL	R	IPVERSION
0x3004	DCDC_CTRL_TGL	RW SYNC	Control

Offset	Name	Type	Description
0x3008	DCDC_EM01CTRL0_TGL	RW SYNC	EM01 Control
0x3010	DCDC_EM23CTRL0_TGL	RW SYNC	EM23 Control
0x3020	DCDC_PFMXCTRL_TGL	RW SYNC	PFMX Control Register
0x3028	DCDC_IF_TGL	RWH INTFLAG	Interrupt Flags
0x302C	DCDC_IEN_TGL	RW	Interrupt Enable
0x3030	DCDC_STATUS_TGL	RH	Status Register
0x3034	DCDC_SYNCBUSY_TGL	RH	Syncbusy Status Register
0x3040	DCDC_LOCK_TGL	W	Lock Register
0x3044	DCDC_LOCKSTATUS_TGL	RH	Lock Status Register

11.7 DCDC Register Description

11.7.1 DCDC_IPVERSION - IPVERSION

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x4																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x4	R	**IPVERSION**
	IPVERSION number			

11.7.2 DCDC_CTRL - Control

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x10								0x0
Access																								RW								RW
Name																								IPKTMAXCTRL								MODE

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:4	IPKTMAXCTRL	0x10	RW	**Ton_max timeout control** Ton_max = (ipk_tmax_ctrl + 1)*0.07us; specifies the timeout duration when attempting to hit programmed peak current; TMAX interrupt flag gives user information whether timeout was hit before reaching peak current
3:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	MODE	0x0	RW	**DCDC/Bypass Mode Control** Used to switch between bypass and dcdc regulation, this triggers a sequence of controls. IF/STATUS registers can be used to check the true status of DCDC regulator/bypass switch

Value	Mode	Description
0	BYPASS	DCDC is OFF, bypass switch is enabled
1	DCDCREGULATION	Request DCDC regulation, bypass switch disabled

11.7.3 DCDC_EM01CTRL0 - EM01 Control

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x1						0x9		
Access																							RW						RW			
Name																							DRVSPEED						IPKVAL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	DRVSPEED	0x1	RW	**EM01 Drive Speed Setting**
	Used to configure drive speed for tradeoff between EMI and Efficiency			

Value	Mode	Description
1	DEFAULT_SETTING	Recommended for use for best efficiency and low EMI

Bit	Name	Reset	Access	Description
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	IPKVAL	0x9	RW	**EM01 Peak Current Setting**
	Used to configure for required peak/load current in EM01; Max load current is approximately 0.4*Ipk			

Value	Mode	Description
3	LOAD36MA	Ipeak = 90mA, Iload = 36mA
4	LOAD40MA	Ipeak = 100mA, Iload = 40mA
5	LOAD44MA	Ipeak = 110mA, Iload = 44mA
6	LOAD48MA	Ipeak = 120mA, Iload = 48mA
7	LOAD52MA	Ipeak = 130mA, Iload = 52mA
8	LOAD56MA	Ipeak = 140mA, Iload = 56mA
9	LOAD60MA	Ipeak = 150mA, Iload = 60mA

11.7.4 DCDC_EM23CTRL0 - EM23 Control

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x1						0x3		
Access																							RW						RW			
Name																							DRVSPEED						IPKVAL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	DRVSPEED	0x1	RW	**EM23 Drive Speed Setting**
	Used to configure drive speed for tradeoff between EMI and Efficiency			
	Value	Mode		Description
	1	DEFAULT_SETTING		Recommended for use for best efficiency and low EMI
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	IPKVAL	0x3	RW	**EM23 Peak Current Setting**
	Used to configure for required peak/load current in EM23			
	Value	Mode		Description
	3	LOAD5MA		Ipeak = 90mA, Iload = 36mA
	9	LOAD10MA		Ipeak = 150mA, Iload = 60mA

11.7.5 DCDC_PFMXCTRL - PFMX Control Register

Offset	Bit Position																															
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																				0xC									0xC			
Access																				RW									RW			
Name																				IPKTMAXCTRL									IPKVAL			

Bit	Name	Reset	Access	Description
31:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12:8	IPKTMAXCTRL	0xC	RW	**Ton_max timeout control**
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	IPKVAL	0xC	RW	**PFMX mode Peak Current Setting**

Used to configure for required peak/load current in PFMX mode

Value	Mode	Description
3	LOAD50MA	Ipeak = 90 mA, Iload = 50 mA
4	LOAD65MA	Ipeak = 100 mA, Iload = 65 mA
5	LOAD73MA	Ipeak = 110 mA, Iload = 73 mA
6	LOAD80MA	Ipeak = 120 mA, Iload = 80 mA
7	LOAD86MA	Ipeak = 130 mA, Iload = 86 mA
8	LOAD93MA	Ipeak = 140 mA, Iload = 93 mA
9	LOAD100MA	Ipeak = 150 mA, Iload = 100 mA
10	LOAD106MA	Ipeak = 160 mA, Iload = 106 mA
11	LOAD113MA	Ipeak = 170 mA, Iload = 113 mA
12	LOAD120MA	Ipeak = 180 mA, Iload = 120 mA

11.7.6 DCDC_IF - Interrupt Flags

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																							RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name																							PFMXMODE	PPMODE	EM4ERR	TMAX	REGULATION	VREGINHIGH	VREGINLOW	RUNNING	WARM	BYPSW

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	PFMXMODE	0x0	RW	**Entered PFMX mode**
	Entered PFMX mode			
8	PPMODE	0x0	RW	**Entered Pulse Pairing mode**
	Entered Pulse Pairing mode			
7	EM4ERR	0x0	RW	**EM4 Entry Request Error**
	EM4 entry error - software requesting EM4 entry when bypass switch is disabled			
6	TMAX	0x0	RW	**Ton_max Timeout Reached**
	Ton_max timeout was reached before peak current could be achieved			
5	REGULATION	0x0	RW	**DCDC in regulation**
	DCDC in regulation, output voltage is in range of target voltage			
4	VREGINHIGH	0x0	RW	**VREGIN above threshold**
	VREGIN/VBAT above threshold			
3	VREGINLOW	0x0	RW	**VREGIN below threshold**
	VREGIN/VBAT below threshold			
2	RUNNING	0x0	RW	**DCDC Running**
	biasen, vcmpen, buckmodeen=1, bypass switch has been turned off.. Note that DCDC might not be in regulation yet. ie output voltage may not be in range of target voltage			
1	WARM	0x0	RW	**DCDC Warmup Time Done**
	100us DCDC warmup time since biasen=1 and dcdcvcmpen=1 complete			
0	BYPSW	0x0	RW	**Bypass Switch Enabled**
	Bypass Switch Enabled			

11.7.7 DCDC_IEN - Interrupt Enable

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																							RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name																							PFMXMODE	PPMODE	EM4ERR	TMAX	REGULATION	VREGINHIGH	VREGINLOW	RUNNING	WARM	BYPSW

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	PFMXMODE	0x0	RW	**PFMX Mode Interrupt Enable**
	PFMX Mode Interrupt Enable			
8	PPMODE	0x0	RW	**Pulse Pairing Mode Interrupt Enable**
	Pulse Pairing Mode Interrupt Enable			
7	EM4ERR	0x0	RW	**EM4 Entry Req Interrupt Enable**
	EM4 Entry Request Error Interrupt Enable			
6	TMAX	0x0	RW	**Ton_max Timeout Interrupt Enable**
	Ton_max Timeout Interrupt Enable			
5	REGULATION	0x0	RW	**DCDC in Regulation Interrupt Enable**
	DCDC in Regulation Interrupt Enable			
4	VREGINHIGH	0x0	RW	**VREGIN above threshold Interrupt Enable**
	VREGIN above threshold Interrupt Enable			
3	VREGINLOW	0x0	RW	**VREGIN below threshold Interrupt Enable**
	VREGIN below threshold Interrupt Enable			
2	RUNNING	0x0	RW	**DCDC Running Interrupt Enable**
	DCDC Running Interrupt Enable			
1	WARM	0x0	RW	**DCDC Warmup Time Done Interrupt Enable**
	DCDC Warmup Time Done Interrupt Enable			
0	BYPSW	0x0	RW	**Bypass Switch Enabled Interrupt Enable**
	Bypass Switch Enabled Interrupt Enable			

11.7.8 DCDC_STATUS - Status Register

Offset	Bit Position																															
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0	0x0				0x0	0x0	0x0	0x0	0x0
Access																							R	R				R	R	R	R	R
Name																							PFMXMODE	PPMODE				BYPCMPOUT	VREGIN	RUNNING	WARM	BYPSW

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	PFMXMODE	0x0	R	**DCDC in PFMX mode**
	DCDC in pfmx mode			
8	PPMODE	0x0	R	**DCDC in pulse-pairing mode**
	DCDC in pulse pairing mode			
7:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	BYPCMPOUT	0x0	R	**Bypass Comparator Output**
	Bypass Comparator Output			
3	VREGIN	0x0	R	**VREGVDD comparator status**
	0: VREGVDD above threshold, 1: VREGVDD below threshold			
2	RUNNING	0x0	R	**DCDC is running**
	DCDC is running (buckmodeen=1, dcdcvcmpen=1, biasen=1, bypsw=0)			
1	WARM	0x0	R	**DCDC Warmup Done**
	100us DCDC warmup time since biasen=1 and dcdcvcmpen=1 complete			
0	BYPSW	0x0	R	**Bypass Switch is currently enabled**
	Bypass switch is currently enabled			

11.7.9 DCDC_SYNCBUSY - Syncbusy Status Register

Offset	Bit Position																													
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0				0x0	0x0	0x0	0x0
Access																									R				R	R	R	R
Name																									PFMXCTRL				EM23CTRL0	EM01CTRL1	EM01CTRL0	CTRL

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	PFMXCTRL	0x0	R	**PFMXCTRL Sync Busy Status**
	PFMXCTRL Sync Busy Status			
6:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	EM23CTRL0	0x0	R	**EM23CTRL0 Sync Busy Status**
	EM23CTRL0 Sync Busy Status			
2	EM01CTRL1	0x0	R	**EM01CTRL1 Sync Bust Status**
	EM01CTRL1 Sync Bust Status			
1	EM01CTRL0	0x0	R	**EM01CTRL0 Sync Busy Status**
	EM01CTRL0 Sync Busy Status			
0	CTRL	0x0	R	**CTRL Sync Busy Status**
	CTRL Sync Busy Status			

11.7.10 DCDC_LOCK - Lock Register

Offset	Bit Position																															
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W							
Name																									LOCKKEY							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x0	W	**Configuration Lock Key**
	Write any other value than the unlock code to lock all DCDC registers			

Value	Mode	Description
43981	UNLOCKKEY	

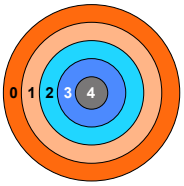
11.7.11 DCDC_LOCKSTATUS - Lock Status Register

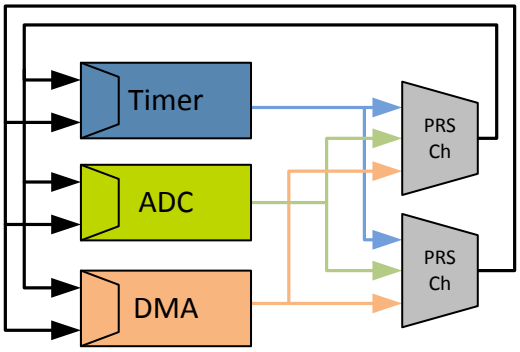
Offset	Bit Position																															
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																R
Name																																LOCK

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	LOCK	0x0	R	**Lock Status**
	Lock Status Read-Only Register			

Value	Mode	Description
0	UNLOCKED	Unlocked State
1	LOCKED	LOCKED STATE

12. PRS - Peripheral Reflex System





Quick Facts

What?

The PRS (Peripheral Reflex System) allows configurable, fast, and autonomous communication between peripherals.

Why?

Events and signals from one peripheral can be used as input signals to trigger actions in other peripherals. PRS reduces latency and ensures predictable timing by reducing software overhead and thus current consumption.

How?

Without CPU intervention the peripherals can send reflex signals to each other in single- or chained steps. The peripherals can be set up to perform actions based on the incoming reflex signals. This results in improved system performance and reduced energy consumption.

12.1 Introduction

The Peripheral Reflex System is a signal routing network allowing direct communication between different peripheral modules without involving the CPU. Peripheral modules which send out reflex signals to the PRS are called producers, and modules accepting reflex signals are called consumers. The PRS routes the reflex signals from producer to consumer peripherals, which perform actions depending on the reflex signals received.

12.2 Features

12 configurable asynchronous channels

- Each channel can be connected to any producer
- Consumers can be configured to listen to any asynchronous channel
- Can generate events to the CPU and the DMA
- Software controlled channel output using the SWPULSE and SWLEVEL registers
- Configurable logic to implement combinational functions between channels; multiple channels may be cascaded to produce more complex functions

4 configurable synchronous channels

- Special set of channels for high speed signalling between IADC and TIMER blocks

12.3 Functional Description

The PRS contains 12 asynchronous and 4 synchronous reflex channels. An overview of an asynchronous PRS reflex channel is shown in Figure 12.1 PRS Asynchronous Channel Overview on page 331. Synchronous channels are similar but do not include the configurable logic block or SWLEVEL / SWPULSE features. Asynchronous channels can be connected to any signal offered by the producers while the synchronous channels are restricted to special signals from the TIMER, IADC, and VDAC modules.

Similarly on the consumer side, all the peripherals can listen to asynchronous channels while only the TIMER, IADC, and VDAC modules can listen to synchronous channels. The consumers of a channel (synchronous or asynchronous) can choose which PRS channel to listen to and perform actions based on the reflex signals routed through that channel. Synchronous channels are only available in EM0 and EM1 while asynchronous channels are available in EM0, EM1, EM2 and EM3.

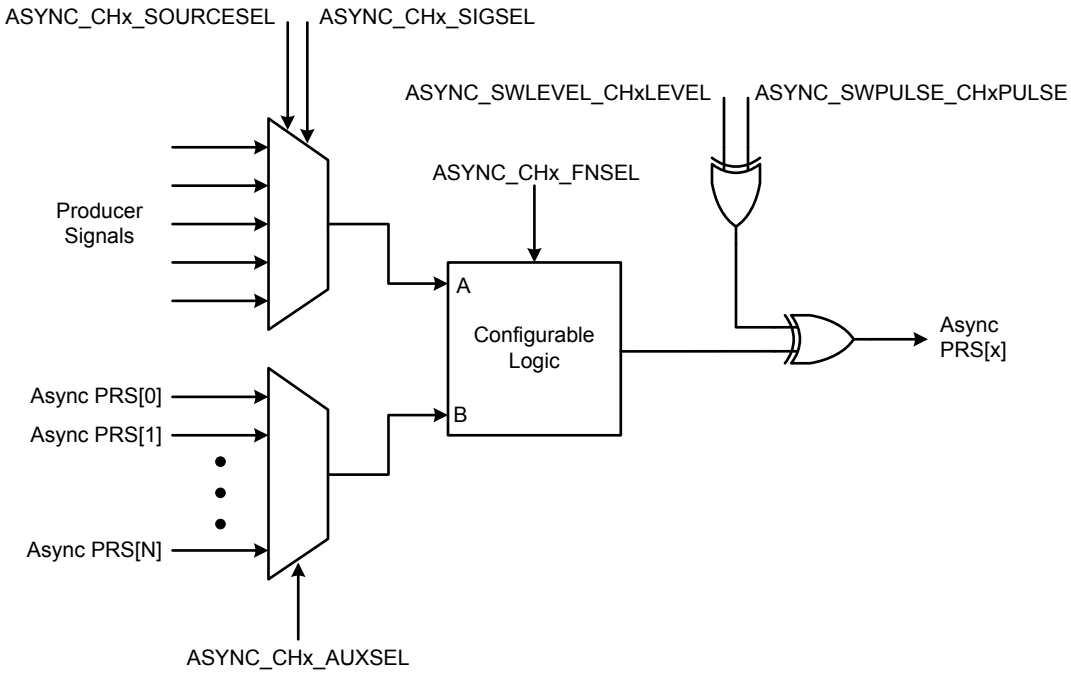


Figure 12.1. PRS Asynchronous Channel Overview

12.3.1 Asynchronous Channel Functions

Different functions can be applied to a reflex signal within the PRS. The asynchronous PRS channels can be manually triggered by writing to PRS_ASYNC_SWPULSE or PRS_ASYNC_SWLEVEL. SWLEVEL[n] is a programmable level for each asynchronous channel and holds the value it is programmed to. Setting SWPULSE[n] will cause the asynchronous channel to output a high pulse that is one EM01GRPACLK clock cycle wide. The SWLEVEL[n] and SWPULSE[n] signals are then XOR'ed with the output from the configurable logic block to form the output signal and is sent to the channel selection logic for every consumer signal. For example, when SWLEVEL[n] is set, if configurable logic produces a signal of 1, this will cause a channel output of 0.

12.3.2 Configurable Logic

The configurable logic feature enables a PRS channel to perform logic operations on the signal coming from the selected producer. Every asynchronous channel has a configurable logic block that can be programmed using the FNSEL field in the asynchronous channel control register. The configurable logic block for each channel has two inputs. Input A is the signal from the selected producer determined by SOURCESEL and SIGSEL of PRS_ASYNCHn_CTRL. Input B may be selected from the output of any other asynchronous PRS channel using the ASYNC_CHx_AUXSEL field. This allows for more complex logic functions to be created using multiple PRS channels.

Table 12.1. Configurable Logic Look up Table

A	B	FNSEL
0	0	FNSEL[0]
0	1	FNSEL[1]
1	0	FNSEL[2]
1	1	FNSEL[3]

The configurable logic feature is implemented as a 2 input look up table, with each bit of FNSEL representing the outcome for a specific input combination (see Table 12.1 Configurable Logic Look up Table on page 332). For example, if input A is 0 and input B is 1, then the PRS output will assume the value of bit 1 of FNSEL (FNSEL[1]).

To calculate the FNSEL field for an "A NAND B" function, the truth table can be filled out as:

Table 12.2. A NAND B Example

A	B	FNSEL = (A NAND B)
0	0	FNSEL[0] = 1
0	1	FNSEL[1] = 1
1	0	FNSEL[2] = 1
1	1	FNSEL[3] = 0

In this example, the value of FNSEL has been calculated to be 0111 (binary), or 0x7.

Using the FNSEL field, a total of 16 two-input logic functions can be implemented, as shown in Table 12.3 List of Logic Functions on page 332.

Table 12.3. List of Logic Functions

FNSEL value	Implemented Function
0x0	0
0x1	A NOR B
0x2	(NOT A) AND B
0x3	NOT A
0x4	A AND (NOT B)
0x5	(NOT B)
0x6	A XOR B
0x7	A NAND B
0x8	A AND B
0x9	A XNOR B

FNSEL value	Implemented Function
0xA	B
0xB	(NOT A) OR B
0xC	A
0xD	A OR (NOT B)
0xE	A OR B
0xF	1

The default value of FNSEL is 0xC, meaning that the input from the selected producer goes through unchanged. This feature can be used to combine multiple channels to get even more complex functions.

12.3.3 Producers

Through SOURCESEL in PRS_SYNCHx_CTRL or PRS_ASYNCHx_CTRL, each PRS channel (synchronous and asynchronous respectively) selects its signal producers. Each producer outputs one or more signals which can be selected by setting the SIGSEL field. Setting the SOURCESEL bits to 0 (Off) leads to a constant 0 output from the input mux regardless of SIGSEL.

The GPIO producer signals depend on settings in the GPIO module. They are selected using the edge interrupt configuration settings described in 23.3.10.1 Standard Interrupt Generation. PIN0 uses settings for the EXTI0 interrupt, PIN1 uses settings for EXTI1, and so on.

For example, to route PB00 as a producer for PRS channel 2, EXTI0, EXTI1, EXTI2, or EXTI3 should be configured to connect to PB00, and the corresponding GPIO PINx should be selected as the PRS channel 2 producer. If we choose EXTI1 via PRS producer "GPIO PIN1":

1. GPIO_EXTIPSELL_EXTIPSEL1 = PORTB, and GPIO_EXTIPINSELL_EXTIPINSEL1 = PIN0 connect PB00 through the EXTI1 signal.
2. PRS_ASYNC_CH2_CTRL_SOURCESEL = GPIO, and PRS_ASYNC_CH2_CTRL_SIGSEL = PIN1 connects the PIN1 (EXTI1) signal to asynchronous PRS channel 2 as a producer.

12.3.3.1 Producer Details

Table 12.4. Synchronous PRS Producers

Peripheral	SOURCESEL	Signal	SIGSEL
TIMER0	TIMER0 (0x01)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
TIMER1	TIMER1 (0x02)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
IADC0	IADC0 (0x03)	SCANENTRYDONE	0x0
		SCANTABLEDONE	0x1
		SINGLEDONE	0x2
TIMER2	TIMER2 (0x04)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
TIMER3	TIMER3 (0x05)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
TIMER4	TIMER4 (0x06)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
VDAC0	VDAC0 (0x07)	CH0DONESYNC	0x0
		CH1DONESYNC	0x1

Table 12.5. Asynchronous PRS Producers

Peripheral	SOURCESEL	Signal	SIGSEL
IADC0	IADC0 (0x01)	SCANENTRYDONE	0x0
		SCANTABLEDONE	0x1
		SINGLEDONE	0x2
LETIMER0	LETIMER0 (0x02)	CH0	0x0
		CH1	0x1
BURTC	BURTC (0x03)	COMP	0x0
		OVERFLOW	0x1
GPIO	GPIO (0x04)	PIN0	0x0
		PIN1	0x1
		PIN2	0x2
		PIN3	0x3
		PIN4	0x4
		PIN5	0x5
		PIN6	0x6
		PIN7	0x7
CMU	CMUL (0x05)	CLKOUT0	0x0
		CLKOUT1	0x1
		CLKOUT2	0x2
PRS	PRSL (0x08)	ASYNCH0	0x0
		ASYNCH1	0x1
		ASYNCH2	0x2
		ASYNCH3	0x3
		ASYNCH4	0x4
		ASYNCH5	0x5
		ASYNCH6	0x6
		ASYNCH7	0x7
	PRS (0x09)	ASYNCH8	0x0
		ASYNCH9	0x1
		ASYNCH10	0x2
		ASYNCH11	0x3
ACMP0	ACMP0 (0x0A)	OUT	0x0
ACMP1	ACMP1 (0x0B)	OUT	0x0

Peripheral	SOURCESEL	Signal	SIGSEL
VDAC0	VDAC0L (0x0C)	CH0WARM	0x0
		CH1WARM	0x1
		CH0DONEASYNC	0x2
		CH1DONEASYNC	0x3
		INTERNALTIMEROF	0x4
		REFRESHTIMEROF	0x5
PCNT0	PCNT0 (0x0E)	DIR	0x0
		UFOF	0x1
SYSRTC0	SYSRTC0 (0x0F)	GRP0OUT0	0x0
		GRP0OUT1	0x1
		GRP1OUT0	0x2
		GRP1OUT1	0x3
LESENSE	LESENSE (0x10)	DECOUT0	0x0
		DECOUT1	0x1
		DECOUT2	0x2
		DECCMP	0x3
HFXO0	HFXO0L (0x11)	STATUS	0x0
		STATUS1	0x1
EUSART0	EUSART0L (0x13)	CS	0x0
		IRDATX	0x1
		RTS	0x2
		RXDATAV	0x3
		TX	0x4
		TXC	0x5
		RXFL	0x6
		TXFL	0x7
USART0	USART0 (0x20)	CS	0x0
		IRTX	0x1
		RTS	0x2
		RXDATA	0x3
		TX	0x4
		TXC	0x5
TIMER0	TIMER0 (0x21)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4

Peripheral	SOURCESEL	Signal	SIGSEL
TIMER1	TIMER1 (0x22)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
TIMER2	TIMER2 (0x23)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
TIMER3	TIMER3 (0x24)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
CORE	CORE (0x25)	CTIOUT0	0x0
		CTIOUT1	0x1
		CTIOUT2	0x2
		CTIOUT3	0x3
TIMER4	TIMER4 (0x32)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
EUSART1	EUSART1L (0x33)	CS	0x0
		IRDATX	0x1
		RTS	0x2
		RXDATAV	0x3
		TX	0x4
		TXC	0x5
		RXFL	0x6
		TXFL	0x7

Peripheral	SOURCESEL	Signal	SIGSEL
EUSART2	EUSART2L (0x35)	CS	0x0
		IRDATX	0x1
		RTS	0x2
		RXDATAV	0x3
		TX	0x4
		TXC	0x5
		RXFL	0x6
		TXFL	0x7

12.3.4 Consumers

Consumer peripherals can be set to listen to a PRS channel and perform an action based on the signal received on that channel. This is done by programming the PRSSEL or SPRSSEL in the consumer registers. SPRSSEL is only present for signals with the ability to listen to synchronous channels. The consumer registers follow the naming convention PRS_CONSUMER_<peripheral_name>_<signal_name>. For example, the PRS_CONSUMER_TIMER0_CC0 register is used to select which PRS channel output is sent to the TIMER0 peripheral's CC0 signal. In turn, the target peripheral should be configured to use the associated PRS trigger as desired. This is described in the individual peripheral chapters.

Note: When configuring the synchronous PRS consumer registers, the target peripheral should be disabled or configured to not use the affected PRS signal. This will ensure that no false triggers occur at the consumer.

12.3.4.1 Event on PRS

The PRS can be used to send events to the MCU to wake the system. This is very useful in combination with the Wait For Event (WFE) instruction. Any asynchronous PRS channel can be selected for this using PRSSEL in PRS_CONSUMER_CORE_M33RXEV.

Using this feature, one can e.g. set up a timer to trigger an event to the MCU periodically, every time letting the MCU continue from a WFE instruction in its program. This can help in performance-critical sections where timing is known, and the goal is to wait for an event, execute some code, then wait for another event, execute some code, and so on.

12.3.4.2 DMA Request on PRS

Up to two independent DMA requests can be generated by the PRS. The PRS asynchronous channels triggering the DMA requests are selected with the PRSSEL fields in the PRS_CONSUMER_LDMAXBAR_DMAREQx registers.The requests are set whenever the selected asynchronous PRS outputs are high.

12.4 PRS Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	PRS_IPVERSION	R	PRS IPVERSION
0x008	PRS_ASYNC_SWPULSE	W	Software Pulse Register
0x00C	PRS_ASYNC_SWLEVEL	RW	Software Level Register
0x010	PRS_ASYNC_PEEK	RH	Async Channel Values
0x014	PRS_SYNC_PEEK	RH	Sync Channel Values
0x018	PRS_ASYNC_CHx_CTRL	RW	Async Channel Control Register
0x048	PRS_SYNC_CHx_CTRL	RW	Sync Channel Control Register
0x058	PRS_CONSUM-ER_CMU_CALDN	RW	CALDN Consumer Register
0x05C	PRS_CONSUMER_CMU_CAL-UP	RW	CALUP Consumer Register
0x060	PRS_CONSUMER_EU-SART0_CLK	RW	CLK Consumer Register
0x064	PRS_CONSUMER_EU-SART0_RX	RW	RX Consumer Register
0x068	PRS_CONSUMER_EU-SART0_TRIGGER	RW	TRIGGER Consumer Register
0x06C	PRS_CONSUMER_EU-SART1_CLK	RW	CLK Consumer Register
0x070	PRS_CONSUMER_EU-SART1_RX	RW	RX Consumer Register
0x074	PRS_CONSUMER_EU-SART1_TRIGGER	RW	TRIGGER Consumer Register
0x078	PRS_CONSUMER_EU-SART2_CLK	RW	CLK Consumer Register
0x07C	PRS_CONSUMER_EU-SART2_RX	RW	RX Consumer Register
0x080	PRS_CONSUMER_EU-SART2_TRIGGER	RW	TRIGGER Consumer Register
0x088	PRS_CONSUM-ER_IADC0_SCANTRIGGER	RW	SCAN Consumer Register
0x08C	PRS_CONSUMER_IADC0_SIN-GLETRIGGER	RW	SINGLE Consumer Register
0x090	PRS_CONSUMER_LDMAX-BAR_DMAREQ0	RW	DMAREQ0 Consumer Register
0x094	PRS_CONSUMER_LDMAX-BAR_DMAREQ1	RW	DMAREQ1 Consumer Register
0x0A8	PRS_CONSUMER_LE-SENSE_START	RW	START Consumer Register
0x0AC	PRS_CONSUMER_LETIM-ER0_CLEAR	RW	CLEAR Consumer Register

Offset	Name	Type	Description
0x0B0	PRS_CONSUMER_LETIMER0_START	RW	START Consumer Register
0x0B4	PRS_CONSUMER_LETIMER0_STOP	RW	STOP Consumer Register
0x0BC	PRS_CONSUMER_PCNT0_S0IN	RW	S0IN Consumer Register
0x0C0	PRS_CONSUMER_PCNT0_S1IN	RW	S1IN Consumer Register
0x114	PRS_CONSUMER_SETAMPER_TAMPERSRC25	RW	TAMPERSRC25 Consumer Register
0x118	PRS_CONSUMER_SETAMPER_TAMPERSRC26	RW	TAMPERSRC26 Consumer Register
0x11C	PRS_CONSUMER_SETAMPER_TAMPERSRC27	RW	TAMPERSRC27 Consumer Register
0x120	PRS_CONSUMER_SETAMPER_TAMPERSRC28	RW	TAMPERSRC28 Consumer Register
0x124	PRS_CONSUMER_SETAMPER_TAMPERSRC29	RW	TAMPERSRC29 Consumer Register
0x128	PRS_CONSUMER_SETAMPER_TAMPERSRC30	RW	TAMPERSRC30 Consumer Register
0x12C	PRS_CONSUMER_SETAMPER_TAMPERSRC31	RW	TAMPERSRC31 Consumer Register
0x130	PRS_CONSUMER_SYSRTC0_IN0	RW	IN0 Consumer Register
0x134	PRS_CONSUMER_SYSRTC0_IN1	RW	IN1 Consumer Register
0x138	PRS_CONSUMER_HFXO0_OSCREQ	RW	OSCREQ Consumer Register
0x13C	PRS_CONSUMER_HFXO0_TIMEOUT	RW	TIMEOUT Consumer Register
0x140	PRS_CONSUMER_CORE_CTIIN0	RW	CTI0 Consumer Selection
0x144	PRS_CONSUMER_CORE_CTIIN1	RW	CTI1 Consumer Selection
0x148	PRS_CONSUMER_CORE_CTIIN2	RW	CTI2 Consumer Selection
0x14C	PRS_CONSUMER_CORE_CTIIN3	RW	CTI3 Consumer Selection
0x150	PRS_CONSUMER_CORE_M33RXEV	RW	M33 Consumer Selection
0x154	PRS_CONSUMER_TIMER0_CC0	RW	CC0 Consumer Register
0x158	PRS_CONSUMER_TIMER0_CC1	RW	CC1 Consumer Register
0x15C	PRS_CONSUMER_TIMER0_CC2	RW	CC2 Consumer Register
0x160	PRS_CONSUMER_TIMER0_DTI	RW	DTI Consumer Register

Offset	Name	Type	Description
0x164	PRS_CONSUMER_TIMER0_DTIFS1	RW	DTI Consumer Register
0x168	PRS_CONSUMER_TIMER0_DTIFS2	RW	DTI Consumer Register
0x16C	PRS_CONSUMER_TIMER1_CC0	RW	CC0 Consumer Register
0x170	PRS_CONSUMER_TIMER1_CC1	RW	CC1 Consumer Register
0x174	PRS_CONSUMER_TIMER1_CC2	RW	CC2 Consumer Register
0x178	PRS_CONSUMER_TIMER1_DTI	RW	DTI Consumer Register
0x17C	PRS_CONSUMER_TIMER1_DTIFS1	RW	DTI Consumer Register
0x180	PRS_CONSUMER_TIMER1_DTIFS2	RW	DTI Consumer Register
0x184	PRS_CONSUMER_TIMER2_CC0	RW	CC0 Consumer Register
0x188	PRS_CONSUMER_TIMER2_CC1	RW	CC1 Consumer Register
0x18C	PRS_CONSUMER_TIMER2_CC2	RW	CC2 Consumer Register
0x190	PRS_CONSUMER_TIMER2_DTI	RW	DTI Consumer Register
0x194	PRS_CONSUMER_TIMER2_DTIFS1	RW	DTI Consumer Register
0x198	PRS_CONSUMER_TIMER2_DTIFS2	RW	DTI Consumer Register
0x19C	PRS_CONSUMER_TIMER3_CC0	RW	CC0 Consumer Register
0x1A0	PRS_CONSUMER_TIMER3_CC1	RW	CC1 Consumer Register
0x1A4	PRS_CONSUMER_TIMER3_CC2	RW	CC2 Consumer Register
0x1A8	PRS_CONSUMER_TIMER3_DTI	RW	DTI Consumer Register
0x1AC	PRS_CONSUMER_TIMER3_DTIFS1	RW	DTI Consumer Register
0x1B0	PRS_CONSUMER_TIMER3_DTIFS2	RW	DTI Consumer Register
0x1B4	PRS_CONSUMER_TIMER4_CC0	RW	CC0 Consumer Register
0x1B8	PRS_CONSUMER_TIMER4_CC1	RW	CC1 Consumer Register
0x1BC	PRS_CONSUMER_TIMER4_CC2	RW	CC2 Consumer Register
0x1C0	PRS_CONSUMER_TIMER4_DTI	RW	DTI Consumer Register
0x1C4	PRS_CONSUMER_TIMER4_DTIFS1	RW	DTI Consumer Register

Offset	Name	Type	Description
0x1C8	PRS_CONSUMER_TIMER4_DTIFS2	RW	DTI Consumer Register
0x1CC	PRS_CONSUMER_USART0_CLK	RW	CLK Consumer Register
0x1D0	PRS_CONSUMER_USART0_IR	RW	IR Consumer Register
0x1D4	PRS_CONSUMER_USART0_RX	RW	RX Consumer Register
0x1D8	PRS_CONSUMER_USART0_TRIGGER	RW	TRIGGER Consumer Register
0x1E8	PRS_CONSUMER_VDAC0_ASYNCTRIGCH0	RW	ASYNCTRIG Consumer Register
0x1EC	PRS_CONSUMER_VDAC0_ASYNCTRIGCH1	RW	ASYNCTRIG Consumer Register
0x1F0	PRS_CONSUMER_VDAC0_SYNCTRIGCH0	RW	SYNCTRIG Consumer Register
0x1F4	PRS_CONSUMER_VDAC0_SYNCTRIGCH1	RW	SYNCTRIG Consumer Register
0x1F8	PRS_CONSUMER_WDOG0_SRC0	RW	SRC0 Consumer Register
0x1FC	PRS_CONSUMER_WDOG0_SRC1	RW	SRC1 Consumer Register
0x200	PRS_CONSUMER_WDOG1_SRC0	RW	SRC0 Consumer Register
0x204	PRS_CONSUMER_WDOG1_SRC1	RW	SRC1 Consumer Register
0x1000	PRS_IPVERSION_SET	R	PRS IPVERSION
0x1008	PRS_ASYNC_SWPULSE_SET	W	Software Pulse Register
0x100C	PRS_ASYNC_SWLEVEL_SET	RW	Software Level Register
0x1010	PRS_ASYNC_PEEK_SET	RH	Async Channel Values
0x1014	PRS_SYNC_PEEK_SET	RH	Sync Channel Values
0x1018	PRS_ASYNC_CHx_CTRL_SET	RW	Async Channel Control Register
0x1048	PRS_SYNC_CHx_CTRL_SET	RW	Sync Channel Control Register
0x1058	PRS_CONSUMER_CMU_CALDN_SET	RW	CALDN Consumer Register
0x105C	PRS_CONSUMER_CMU_CALUP_SET	RW	CALUP Consumer Register
0x1060	PRS_CONSUMER_EUSART0_CLK_SET	RW	CLK Consumer Register
0x1064	PRS_CONSUMER_EUSART0_RX_SET	RW	RX Consumer Register
0x1068	PRS_CONSUMER_EUSART0_TRIGGER_SET	RW	TRIGGER Consumer Register
0x106C	PRS_CONSUMER_EUSART1_CLK_SET	RW	CLK Consumer Register

Offset	Name	Type	Description
0x1070	PRS_CONSUMER_EUSART1_RX_SET	RW	RX Consumer Register
0x1074	PRS_CONSUMER_EUSART1_TRIGGER_SET	RW	TRIGGER Consumer Register
0x1078	PRS_CONSUMER_EUSART2_CLK_SET	RW	CLK Consumer Register
0x107C	PRS_CONSUMER_EUSART2_RX_SET	RW	RX Consumer Register
0x1080	PRS_CONSUMER_EUSART2_TRIGGER_SET	RW	TRIGGER Consumer Register
0x1088	PRS_CONSUMER_IADC0_SCANTRIGGER_SET	RW	SCAN Consumer Register
0x108C	PRS_CONSUMER_IADC0_SINGLETRIGGER_SET	RW	SINGLE Consumer Register
0x1090	PRS_CONSUMER_LDMAXBAR_DMAREQ0_SET	RW	DMAREQ0 Consumer Register
0x1094	PRS_CONSUMER_LDMAXBAR_DMAREQ1_SET	RW	DMAREQ1 Consumer Register
0x10A8	PRS_CONSUMER_LESENSE_START_SET	RW	START Consumer Register
0x10AC	PRS_CONSUMER_LETIMER0_CLEAR_SET	RW	CLEAR Consumer Register
0x10B0	PRS_CONSUMER_LETIMER0_START_SET	RW	START Consumer Register
0x10B4	PRS_CONSUMER_LETIMER0_STOP_SET	RW	STOP Consumer Register
0x10BC	PRS_CONSUMER_PCNT0_S0IN_SET	RW	S0IN Consumer Register
0x10C0	PRS_CONSUMER_PCNT0_S1IN_SET	RW	S1IN Consumer Register
0x1114	PRS_CONSUMER_SETAMPER_TAMPERSRC25_SET	RW	TAMPERSRC25 Consumer Register
0x1118	PRS_CONSUMER_SETAMPER_TAMPERSRC26_SET	RW	TAMPERSRC26 Consumer Register
0x111C	PRS_CONSUMER_SETAMPER_TAMPERSRC27_SET	RW	TAMPERSRC27 Consumer Register
0x1120	PRS_CONSUMER_SETAMPER_TAMPERSRC28_SET	RW	TAMPERSRC28 Consumer Register
0x1124	PRS_CONSUMER_SETAMPER_TAMPERSRC29_SET	RW	TAMPERSRC29 Consumer Register
0x1128	PRS_CONSUMER_SETAMPER_TAMPERSRC30_SET	RW	TAMPERSRC30 Consumer Register
0x112C	PRS_CONSUMER_SETAMPER_TAMPERSRC31_SET	RW	TAMPERSRC31 Consumer Register
0x1130	PRS_CONSUMER_SYSRTC0_IN0_SET	RW	IN0 Consumer Register

Offset	Name	Type	Description
0x1134	PRS_CONSUMER_SYSRTC0_IN1_SET	RW	IN1 Consumer Register
0x1138	PRS_CONSUMER_HFXO0_OSCREQ_SET	RW	OSCREQ Consumer Register
0x113C	PRS_CONSUMER_HFXO0_TIMEOUT_SET	RW	TIMEOUT Consumer Register
0x1140	PRS_CONSUMER_CORE_CTIIN0_SET	RW	CTI0 Consumer Selection
0x1144	PRS_CONSUMER_CORE_CTIIN1_SET	RW	CTI1 Consumer Selection
0x1148	PRS_CONSUMER_CORE_CTIIN2_SET	RW	CTI2 Consumer Selection
0x114C	PRS_CONSUMER_CORE_CTIIN3_SET	RW	CTI3 Consumer Selection
0x1150	PRS_CONSUMER_CORE_M33RXEV_SET	RW	M33 Consumer Selection
0x1154	PRS_CONSUMER_TIMER0_CC0_SET	RW	CC0 Consumer Register
0x1158	PRS_CONSUMER_TIMER0_CC1_SET	RW	CC1 Consumer Register
0x115C	PRS_CONSUMER_TIMER0_CC2_SET	RW	CC2 Consumer Register
0x1160	PRS_CONSUMER_TIMER0_DTI_SET	RW	DTI Consumer Register
0x1164	PRS_CONSUMER_TIMER0_DTIFS1_SET	RW	DTI Consumer Register
0x1168	PRS_CONSUMER_TIMER0_DTIFS2_SET	RW	DTI Consumer Register
0x116C	PRS_CONSUMER_TIMER1_CC0_SET	RW	CC0 Consumer Register
0x1170	PRS_CONSUMER_TIMER1_CC1_SET	RW	CC1 Consumer Register
0x1174	PRS_CONSUMER_TIMER1_CC2_SET	RW	CC2 Consumer Register
0x1178	PRS_CONSUMER_TIMER1_DTI_SET	RW	DTI Consumer Register
0x117C	PRS_CONSUMER_TIMER1_DTIFS1_SET	RW	DTI Consumer Register
0x1180	PRS_CONSUMER_TIMER1_DTIFS2_SET	RW	DTI Consumer Register
0x1184	PRS_CONSUMER_TIMER2_CC0_SET	RW	CC0 Consumer Register
0x1188	PRS_CONSUMER_TIMER2_CC1_SET	RW	CC1 Consumer Register
0x118C	PRS_CONSUMER_TIMER2_CC2_SET	RW	CC2 Consumer Register

Offset	Name	Type	Description
0x1190	PRS_CONSUMER_TIMER2_DTI_SET	RW	DTI Consumer Register
0x1194	PRS_CONSUMER_TIMER2_DTIFS1_SET	RW	DTI Consumer Register
0x1198	PRS_CONSUMER_TIMER2_DTIFS2_SET	RW	DTI Consumer Register
0x119C	PRS_CONSUMER_TIMER3_CC0_SET	RW	CC0 Consumer Register
0x11A0	PRS_CONSUMER_TIMER3_CC1_SET	RW	CC1 Consumer Register
0x11A4	PRS_CONSUMER_TIMER3_CC2_SET	RW	CC2 Consumer Register
0x11A8	PRS_CONSUMER_TIMER3_DTI_SET	RW	DTI Consumer Register
0x11AC	PRS_CONSUMER_TIMER3_DTIFS1_SET	RW	DTI Consumer Register
0x11B0	PRS_CONSUMER_TIMER3_DTIFS2_SET	RW	DTI Consumer Register
0x11B4	PRS_CONSUMER_TIMER4_CC0_SET	RW	CC0 Consumer Register
0x11B8	PRS_CONSUMER_TIMER4_CC1_SET	RW	CC1 Consumer Register
0x11BC	PRS_CONSUMER_TIMER4_CC2_SET	RW	CC2 Consumer Register
0x11C0	PRS_CONSUMER_TIMER4_DTI_SET	RW	DTI Consumer Register
0x11C4	PRS_CONSUMER_TIMER4_DTIFS1_SET	RW	DTI Consumer Register
0x11C8	PRS_CONSUMER_TIMER4_DTIFS2_SET	RW	DTI Consumer Register
0x11CC	PRS_CONSUMER_USART0_CLK_SET	RW	CLK Consumer Register
0x11D0	PRS_CONSUMER_USART0_IR_SET	RW	IR Consumer Register
0x11D4	PRS_CONSUMER_USART0_RX_SET	RW	RX Consumer Register
0x11D8	PRS_CONSUMER_USART0_TRIGGER_SET	RW	TRIGGER Consumer Register
0x11E8	PRS_CONSUMER_VDAC0_ASYNCTRIGCH0_SET	RW	ASYNCTRIG Consumer Register
0x11EC	PRS_CONSUMER_VDAC0_ASYNCTRIGCH1_SET	RW	ASYNCTRIG Consumer Register
0x11F0	PRS_CONSUMER_VDAC0_SYNCTRIGCH0_SET	RW	SYNCTRIG Consumer Register

Offset	Name	Type	Description
0x11F4	PRS_CONSUMER_VDAC0_SYNCTRIGCH1_SET	RW	SYNCTRIG Consumer Register
0x11F8	PRS_CONSUMER_WDOG0_SRC0_SET	RW	SRC0 Consumer Register
0x11FC	PRS_CONSUMER_WDOG0_SRC1_SET	RW	SRC1 Consumer Register
0x1200	PRS_CONSUMER_WDOG1_SRC0_SET	RW	SRC0 Consumer Register
0x1204	PRS_CONSUMER_WDOG1_SRC1_SET	RW	SRC1 Consumer Register
0x2000	PRS_IPVERSION_CLR	R	PRS IPVERSION
0x2008	PRS_ASYNC_SWPULSE_CLR	W	Software Pulse Register
0x200C	PRS_ASYNC_SWLEVEL_CLR	RW	Software Level Register
0x2010	PRS_ASYNC_PEEK_CLR	RH	Async Channel Values
0x2014	PRS_SYNC_PEEK_CLR	RH	Sync Channel Values
0x2018	PRS_ASYNC_CHx_CTRL_CLR	RW	Async Channel Control Register
0x2048	PRS_SYNC_CHx_CTRL_CLR	RW	Sync Channel Control Register
0x2058	PRS_CONSUMER_CMU_CALDN_CLR	RW	CALDN Consumer Register
0x205C	PRS_CONSUMER_CMU_CALUP_CLR	RW	CALUP Consumer Register
0x2060	PRS_CONSUMER_EUSART0_CLK_CLR	RW	CLK Consumer Register
0x2064	PRS_CONSUMER_EUSART0_RX_CLR	RW	RX Consumer Register
0x2068	PRS_CONSUMER_EUSART0_TRIGGER_CLR	RW	TRIGGER Consumer Register
0x206C	PRS_CONSUMER_EUSART1_CLK_CLR	RW	CLK Consumer Register
0x2070	PRS_CONSUMER_EUSART1_RX_CLR	RW	RX Consumer Register
0x2074	PRS_CONSUMER_EUSART1_TRIGGER_CLR	RW	TRIGGER Consumer Register
0x2078	PRS_CONSUMER_EUSART2_CLK_CLR	RW	CLK Consumer Register
0x207C	PRS_CONSUMER_EUSART2_RX_CLR	RW	RX Consumer Register
0x2080	PRS_CONSUMER_EUSART2_TRIGGER_CLR	RW	TRIGGER Consumer Register
0x2088	PRS_CONSUMER_IADC0_SCANTRIGGER_CLR	RW	SCAN Consumer Register
0x208C	PRS_CONSUMER_IADC0_SINGLETRIGGER_CLR	RW	SINGLE Consumer Register

Offset	Name	Type	Description
0x2090	PRS_CONSUMER_LDMAX-BAR_DMAREQ0_CLR	RW	DMAREQ0 Consumer Register
0x2094	PRS_CONSUMER_LDMAX-BAR_DMAREQ1_CLR	RW	DMAREQ1 Consumer Register
0x20A8	PRS_CONSUMER_LE-SENSE_START_CLR	RW	START Consumer Register
0x20AC	PRS_CONSUMER_LETIM-ER0_CLEAR_CLR	RW	CLEAR Consumer Register
0x20B0	PRS_CONSUMER_LETIM-ER0_START_CLR	RW	START Consumer Register
0x20B4	PRS_CONSUMER_LETIM-ER0_STOP_CLR	RW	STOP Consumer Register
0x20BC	PRS_CONSUM-ER_PCNT0_S0IN_CLR	RW	S0IN Consumer Register
0x20C0	PRS_CONSUM-ER_PCNT0_S1IN_CLR	RW	S1IN Consumer Register
0x2114	PRS_CONSUMER_SETAM-PER_TAMPERSRC25_CLR	RW	TAMPERSRC25 Consumer Register
0x2118	PRS_CONSUMER_SETAM-PER_TAMPERSRC26_CLR	RW	TAMPERSRC26 Consumer Register
0x211C	PRS_CONSUMER_SETAM-PER_TAMPERSRC27_CLR	RW	TAMPERSRC27 Consumer Register
0x2120	PRS_CONSUMER_SETAM-PER_TAMPERSRC28_CLR	RW	TAMPERSRC28 Consumer Register
0x2124	PRS_CONSUMER_SETAM-PER_TAMPERSRC29_CLR	RW	TAMPERSRC29 Consumer Register
0x2128	PRS_CONSUMER_SETAM-PER_TAMPERSRC30_CLR	RW	TAMPERSRC30 Consumer Register
0x212C	PRS_CONSUMER_SETAM-PER_TAMPERSRC31_CLR	RW	TAMPERSRC31 Consumer Register
0x2130	PRS_CONSUM-ER_SYSRTC0_IN0_CLR	RW	IN0 Consumer Register
0x2134	PRS_CONSUM-ER_SYSRTC0_IN1_CLR	RW	IN1 Consumer Register
0x2138	PRS_CONSUMER_HFXO0_OS-CREQ_CLR	RW	OSCREQ Consumer Register
0x213C	PRS_CONSUM-ER_HFXO0_TIMEOUT_CLR	RW	TIMEOUT Consumer Register
0x2140	PRS_CONSUM-ER_CORE_CTIIN0_CLR	RW	CTI0 Consumer Selection
0x2144	PRS_CONSUM-ER_CORE_CTIIN1_CLR	RW	CTI1 Consumer Selection
0x2148	PRS_CONSUM-ER_CORE_CTIIN2_CLR	RW	CTI2 Consumer Selection
0x214C	PRS_CONSUM-ER_CORE_CTIIN3_CLR	RW	CTI3 Consumer Selection

Offset	Name	Type	Description
0x2150	PRS_CONSUMER_CORE_M33RXEV_CLR	RW	M33 Consumer Selection
0x2154	PRS_CONSUMER_TIMER0_CC0_CLR	RW	CC0 Consumer Register
0x2158	PRS_CONSUMER_TIMER0_CC1_CLR	RW	CC1 Consumer Register
0x215C	PRS_CONSUMER_TIMER0_CC2_CLR	RW	CC2 Consumer Register
0x2160	PRS_CONSUMER_TIMER0_DTI_CLR	RW	DTI Consumer Register
0x2164	PRS_CONSUMER_TIMER0_DTIFS1_CLR	RW	DTI Consumer Register
0x2168	PRS_CONSUMER_TIMER0_DTIFS2_CLR	RW	DTI Consumer Register
0x216C	PRS_CONSUMER_TIMER1_CC0_CLR	RW	CC0 Consumer Register
0x2170	PRS_CONSUMER_TIMER1_CC1_CLR	RW	CC1 Consumer Register
0x2174	PRS_CONSUMER_TIMER1_CC2_CLR	RW	CC2 Consumer Register
0x2178	PRS_CONSUMER_TIMER1_DTI_CLR	RW	DTI Consumer Register
0x217C	PRS_CONSUMER_TIMER1_DTIFS1_CLR	RW	DTI Consumer Register
0x2180	PRS_CONSUMER_TIMER1_DTIFS2_CLR	RW	DTI Consumer Register
0x2184	PRS_CONSUMER_TIMER2_CC0_CLR	RW	CC0 Consumer Register
0x2188	PRS_CONSUMER_TIMER2_CC1_CLR	RW	CC1 Consumer Register
0x218C	PRS_CONSUMER_TIMER2_CC2_CLR	RW	CC2 Consumer Register
0x2190	PRS_CONSUMER_TIMER2_DTI_CLR	RW	DTI Consumer Register
0x2194	PRS_CONSUMER_TIMER2_DTIFS1_CLR	RW	DTI Consumer Register
0x2198	PRS_CONSUMER_TIMER2_DTIFS2_CLR	RW	DTI Consumer Register
0x219C	PRS_CONSUMER_TIMER3_CC0_CLR	RW	CC0 Consumer Register
0x21A0	PRS_CONSUMER_TIMER3_CC1_CLR	RW	CC1 Consumer Register
0x21A4	PRS_CONSUMER_TIMER3_CC2_CLR	RW	CC2 Consumer Register
0x21A8	PRS_CONSUMER_TIMER3_DTI_CLR	RW	DTI Consumer Register

Offset	Name	Type	Description
0x21AC	PRS_CONSUMER_TIMER3_DTIFS1_CLR	RW	DTI Consumer Register
0x21B0	PRS_CONSUMER_TIMER3_DTIFS2_CLR	RW	DTI Consumer Register
0x21B4	PRS_CONSUMER_TIMER4_CC0_CLR	RW	CC0 Consumer Register
0x21B8	PRS_CONSUMER_TIMER4_CC1_CLR	RW	CC1 Consumer Register
0x21BC	PRS_CONSUMER_TIMER4_CC2_CLR	RW	CC2 Consumer Register
0x21C0	PRS_CONSUMER_TIMER4_DTI_CLR	RW	DTI Consumer Register
0x21C4	PRS_CONSUMER_TIMER4_DTIFS1_CLR	RW	DTI Consumer Register
0x21C8	PRS_CONSUMER_TIMER4_DTIFS2_CLR	RW	DTI Consumer Register
0x21CC	PRS_CONSUMER_USART0_CLK_CLR	RW	CLK Consumer Register
0x21D0	PRS_CONSUMER_USART0_IR_CLR	RW	IR Consumer Register
0x21D4	PRS_CONSUMER_USART0_RX_CLR	RW	RX Consumer Register
0x21D8	PRS_CONSUMER_USART0_TRIGGER_CLR	RW	TRIGGER Consumer Register
0x21E8	PRS_CONSUMER_VDAC0_ASYNCTRIGCH0_CLR	RW	ASYNCTRIG Consumer Register
0x21EC	PRS_CONSUMER_VDAC0_ASYNCTRIGCH1_CLR	RW	ASYNCTRIG Consumer Register
0x21F0	PRS_CONSUMER_VDAC0_SYNCTRIGCH0_CLR	RW	SYNCTRIG Consumer Register
0x21F4	PRS_CONSUMER_VDAC0_SYNCTRIGCH1_CLR	RW	SYNCTRIG Consumer Register
0x21F8	PRS_CONSUMER_WDOG0_SRC0_CLR	RW	SRC0 Consumer Register
0x21FC	PRS_CONSUMER_WDOG0_SRC1_CLR	RW	SRC1 Consumer Register
0x2200	PRS_CONSUMER_WDOG1_SRC0_CLR	RW	SRC0 Consumer Register
0x2204	PRS_CONSUMER_WDOG1_SRC1_CLR	RW	SRC1 Consumer Register
0x3000	PRS_IPVERSION_TGL	R	PRS IPVERSION
0x3008	PRS_ASYNC_SWPULSE_TGL	W	Software Pulse Register
0x300C	PRS_ASYNC_SWLEVEL_TGL	RW	Software Level Register

Offset	Name	Type	Description
0x3010	PRS_ASYNC_PEEK_TGL	RH	Async Channel Values
0x3014	PRS_SYNC_PEEK_TGL	RH	Sync Channel Values
0x3018	PRS_ASYNC_CHx_CTRL_TGL	RW	Async Channel Control Register
0x3048	PRS_SYNC_CHx_CTRL_TGL	RW	Sync Channel Control Register
0x3058	PRS_CONSUMER_CMU_CALDN_TGL	RW	CALDN Consumer Register
0x305C	PRS_CONSUMER_CMU_CALUP_TGL	RW	CALUP Consumer Register
0x3060	PRS_CONSUMER_EUSART0_CLK_TGL	RW	CLK Consumer Register
0x3064	PRS_CONSUMER_EUSART0_RX_TGL	RW	RX Consumer Register
0x3068	PRS_CONSUMER_EUSART0_TRIGGER_TGL	RW	TRIGGER Consumer Register
0x306C	PRS_CONSUMER_EUSART1_CLK_TGL	RW	CLK Consumer Register
0x3070	PRS_CONSUMER_EUSART1_RX_TGL	RW	RX Consumer Register
0x3074	PRS_CONSUMER_EUSART1_TRIGGER_TGL	RW	TRIGGER Consumer Register
0x3078	PRS_CONSUMER_EUSART2_CLK_TGL	RW	CLK Consumer Register
0x307C	PRS_CONSUMER_EUSART2_RX_TGL	RW	RX Consumer Register
0x3080	PRS_CONSUMER_EUSART2_TRIGGER_TGL	RW	TRIGGER Consumer Register
0x3088	PRS_CONSUMER_IADC0_SCANTRIGGER_TGL	RW	SCAN Consumer Register
0x308C	PRS_CONSUMER_IADC0_SINGLETRIGGER_TGL	RW	SINGLE Consumer Register
0x3090	PRS_CONSUMER_LDMAXBAR_DMAREQ0_TGL	RW	DMAREQ0 Consumer Register
0x3094	PRS_CONSUMER_LDMAXBAR_DMAREQ1_TGL	RW	DMAREQ1 Consumer Register
0x30A8	PRS_CONSUMER_LESENSE_START_TGL	RW	START Consumer Register
0x30AC	PRS_CONSUMER_LETIMER0_CLEAR_TGL	RW	CLEAR Consumer Register
0x30B0	PRS_CONSUMER_LETIMER0_START_TGL	RW	START Consumer Register
0x30B4	PRS_CONSUMER_LETIMER0_STOP_TGL	RW	STOP Consumer Register
0x30BC	PRS_CONSUMER_PCNT0_S0IN_TGL	RW	S0IN Consumer Register

Offset	Name	Type	Description
0x30C0	PRS_CONSUMER_PCNT0_S1IN_TGL	RW	S1IN Consumer Register
0x3114	PRS_CONSUMER_SETAMPER_TAMPERSRC25_TGL	RW	TAMPERSRC25 Consumer Register
0x3118	PRS_CONSUMER_SETAMPER_TAMPERSRC26_TGL	RW	TAMPERSRC26 Consumer Register
0x311C	PRS_CONSUMER_SETAMPER_TAMPERSRC27_TGL	RW	TAMPERSRC27 Consumer Register
0x3120	PRS_CONSUMER_SETAMPER_TAMPERSRC28_TGL	RW	TAMPERSRC28 Consumer Register
0x3124	PRS_CONSUMER_SETAMPER_TAMPERSRC29_TGL	RW	TAMPERSRC29 Consumer Register
0x3128	PRS_CONSUMER_SETAMPER_TAMPERSRC30_TGL	RW	TAMPERSRC30 Consumer Register
0x312C	PRS_CONSUMER_SETAMPER_TAMPERSRC31_TGL	RW	TAMPERSRC31 Consumer Register
0x3130	PRS_CONSUMER_SYSRTC0_IN0_TGL	RW	IN0 Consumer Register
0x3134	PRS_CONSUMER_SYSRTC0_IN1_TGL	RW	IN1 Consumer Register
0x3138	PRS_CONSUMER_HFXO0_OSCREQ_TGL	RW	OSCREQ Consumer Register
0x313C	PRS_CONSUMER_HFXO0_TIMEOUT_TGL	RW	TIMEOUT Consumer Register
0x3140	PRS_CONSUMER_CORE_CTIIN0_TGL	RW	CTI0 Consumer Selection
0x3144	PRS_CONSUMER_CORE_CTIIN1_TGL	RW	CTI1 Consumer Selection
0x3148	PRS_CONSUMER_CORE_CTIIN2_TGL	RW	CTI2 Consumer Selection
0x314C	PRS_CONSUMER_CORE_CTIIN3_TGL	RW	CTI3 Consumer Selection
0x3150	PRS_CONSUMER_CORE_M33RXEV_TGL	RW	M33 Consumer Selection
0x3154	PRS_CONSUMER_TIMER0_CC0_TGL	RW	CC0 Consumer Register
0x3158	PRS_CONSUMER_TIMER0_CC1_TGL	RW	CC1 Consumer Register
0x315C	PRS_CONSUMER_TIMER0_CC2_TGL	RW	CC2 Consumer Register
0x3160	PRS_CONSUMER_TIMER0_DTI_TGL	RW	DTI Consumer Register
0x3164	PRS_CONSUMER_TIMER0_DTIFS1_TGL	RW	DTI Consumer Register
0x3168	PRS_CONSUMER_TIMER0_DTIFS2_TGL	RW	DTI Consumer Register

Offset	Name	Type	Description
0x316C	PRS_CONSUMER_TIMER1_CC0_TGL	RW	CC0 Consumer Register
0x3170	PRS_CONSUMER_TIMER1_CC1_TGL	RW	CC1 Consumer Register
0x3174	PRS_CONSUMER_TIMER1_CC2_TGL	RW	CC2 Consumer Register
0x3178	PRS_CONSUMER_TIMER1_DTI_TGL	RW	DTI Consumer Register
0x317C	PRS_CONSUMER_TIMER1_DTIFS1_TGL	RW	DTI Consumer Register
0x3180	PRS_CONSUMER_TIMER1_DTIFS2_TGL	RW	DTI Consumer Register
0x3184	PRS_CONSUMER_TIMER2_CC0_TGL	RW	CC0 Consumer Register
0x3188	PRS_CONSUMER_TIMER2_CC1_TGL	RW	CC1 Consumer Register
0x318C	PRS_CONSUMER_TIMER2_CC2_TGL	RW	CC2 Consumer Register
0x3190	PRS_CONSUMER_TIMER2_DTI_TGL	RW	DTI Consumer Register
0x3194	PRS_CONSUMER_TIMER2_DTIFS1_TGL	RW	DTI Consumer Register
0x3198	PRS_CONSUMER_TIMER2_DTIFS2_TGL	RW	DTI Consumer Register
0x319C	PRS_CONSUMER_TIMER3_CC0_TGL	RW	CC0 Consumer Register
0x31A0	PRS_CONSUMER_TIMER3_CC1_TGL	RW	CC1 Consumer Register
0x31A4	PRS_CONSUMER_TIMER3_CC2_TGL	RW	CC2 Consumer Register
0x31A8	PRS_CONSUMER_TIMER3_DTI_TGL	RW	DTI Consumer Register
0x31AC	PRS_CONSUMER_TIMER3_DTIFS1_TGL	RW	DTI Consumer Register
0x31B0	PRS_CONSUMER_TIMER3_DTIFS2_TGL	RW	DTI Consumer Register
0x31B4	PRS_CONSUMER_TIMER4_CC0_TGL	RW	CC0 Consumer Register
0x31B8	PRS_CONSUMER_TIMER4_CC1_TGL	RW	CC1 Consumer Register
0x31BC	PRS_CONSUMER_TIMER4_CC2_TGL	RW	CC2 Consumer Register
0x31C0	PRS_CONSUMER_TIMER4_DTI_TGL	RW	DTI Consumer Register
0x31C4	PRS_CONSUMER_TIMER4_DTIFS1_TGL	RW	DTI Consumer Register

Offset	Name	Type	Description
0x31C8	PRS_CONSUMER_TIMER4_DTIFS2_TGL	RW	DTI Consumer Register
0x31CC	PRS_CONSUMER_USART0_CLK_TGL	RW	CLK Consumer Register
0x31D0	PRS_CONSUMER_USART0_IR_TGL	RW	IR Consumer Register
0x31D4	PRS_CONSUMER_USART0_RX_TGL	RW	RX Consumer Register
0x31D8	PRS_CONSUMER_USART0_TRIGGER_TGL	RW	TRIGGER Consumer Register
0x31E8	PRS_CONSUMER_VDAC0_ASYNCTRIGCH0_TGL	RW	ASYNCTRIG Consumer Register
0x31EC	PRS_CONSUMER_VDAC0_ASYNCTRIGCH1_TGL	RW	ASYNCTRIG Consumer Register
0x31F0	PRS_CONSUMER_VDAC0_SYNCTRIGCH0_TGL	RW	SYNCTRIG Consumer Register
0x31F4	PRS_CONSUMER_VDAC0_SYNCTRIGCH1_TGL	RW	SYNCTRIG Consumer Register
0x31F8	PRS_CONSUMER_WDOG0_SRC0_TGL	RW	SRC0 Consumer Register
0x31FC	PRS_CONSUMER_WDOG0_SRC1_TGL	RW	SRC1 Consumer Register
0x3200	PRS_CONSUMER_WDOG1_SRC0_TGL	RW	SRC0 Consumer Register
0x3204	PRS_CONSUMER_WDOG1_SRC1_TGL	RW	SRC1 Consumer Register

12.5 PRS Register Description

12.5.1 PRS_IPVERSION - PRS IPVERSION

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x2																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x2	R	**New BitField**
	The read only IPVERSION field goves the version for this module. There may be minor software changes required for modules with different values of IPVERSION			

12.5.2 PRS_ASYNC_SWPULSE - Software Pulse Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																					W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)
Name																					CH11PULSE	CH10PULSE	CH9PULSE	CH8PULSE	CH7PULSE	CH6PULSE	CH5PULSE	CH4PULSE	CH3PULSE	CH2PULSE	CH1PULSE	CH0PULSE

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	CH11PULSE	0x0	W(nB)	**Channel pulse**
	Channel 11 pulse generation			
10	CH10PULSE	0x0	W(nB)	**Channel pulse**
	Channel 10 pulse generation			
9	CH9PULSE	0x0	W(nB)	**Channel pulse**
	Channel 9 pulse generation			
8	CH8PULSE	0x0	W(nB)	**Channel pulse**
	Channel 8 pulse generation			
7	CH7PULSE	0x0	W(nB)	**Channel pulse**
	Channel 7 pulse generation			
6	CH6PULSE	0x0	W(nB)	**Channel pulse**
	Channel 6 pulse generation			
5	CH5PULSE	0x0	W(nB)	**Channel pulse**
	Channel 5 pulse generation			
4	CH4PULSE	0x0	W(nB)	**Channel pulse**
	Channel 4 pulse generation			
3	CH3PULSE	0x0	W(nB)	**Channel pulse**
	Channel 3 pulse generation			
2	CH2PULSE	0x0	W(nB)	**Channel pulse**
	Channel 2 pulse generation			
1	CH1PULSE	0x0	W(nB)	**Channel pulse**
	Channel 1 pulse generation			
0	CH0PULSE	0x0	W(nB)	**Channel pulse**
	Channel 0 pulse generation			

12.5.3 PRS_ASYNC_SWLEVEL - Software Level Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name																					CH11LEVEL	CH10LEVEL	CH9LEVEL	CH8LEVEL	CH7LEVEL	CH6LEVEL	CH5LEVEL	CH4LEVEL	CH3LEVEL	CH2LEVEL	CH1LEVEL	CH0LEVEL

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	CH11LEVEL	0x0	RW	**Channel Level**
	Channel 11 Software Level			
10	CH10LEVEL	0x0	RW	**Channel Level**
	Channel 10 Software Level			
9	CH9LEVEL	0x0	RW	**Channel Level**
	Channel 9 Software Level			
8	CH8LEVEL	0x0	RW	**Channel Level**
	Channel 8 Software Level			
7	CH7LEVEL	0x0	RW	**Channel Level**
	Channel 7 Software Level			
6	CH6LEVEL	0x0	RW	**Channel Level**
	Channel 6 Software Level			
5	CH5LEVEL	0x0	RW	**Channel Level**
	Channel 5 Software Level			
4	CH4LEVEL	0x0	RW	**Channel Level**
	Channel 4 Software Level			
3	CH3LEVEL	0x0	RW	**Channel Level**
	Channel 3 Software Level			
2	CH2LEVEL	0x0	RW	**Channel Level**
	Channel 2 Software Level			
1	CH1LEVEL	0x0	RW	**Channel Level**
	Channel 1 Software Level			
0	CH0LEVEL	0x0	RW	**Channel Level**
	Channel 0 Software Level			

12.5.4 PRS_ASYNC_PEEK - Async Channel Values

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																					R	R	R	R	R	R	R	R	R	R	R	R
Name																					CH11VAL	CH10VAL	CH9VAL	CH8VAL	CH7VAL	CH6VAL	CH5VAL	CH4VAL	CH3VAL	CH2VAL	CH1VAL	CH0VAL

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	CH11VAL	0x0	R	**Channel 11 Current Value**
	See bit 0.			
10	CH10VAL	0x0	R	**Channel 10 Current Value**
	See bit 0.			
9	CH9VAL	0x0	R	**Channel 9 Current Value**
	See bit 0.			
8	CH8VAL	0x0	R	**Channel 8 Current Value**
	See bit 0.			
7	CH7VAL	0x0	R	**Channel 7 Current Value**
	See bit 0.			
6	CH6VAL	0x0	R	**Channel 6 Current Value**
	See bit 0.			
5	CH5VAL	0x0	R	**Channel 5 Current Value**
	See bit 0.			
4	CH4VAL	0x0	R	**Channel 4 Current Value**
	See bit 0.			
3	CH3VAL	0x0	R	**Channel 3 Current Value**
	See bit 0.			
2	CH2VAL	0x0	R	**Channel 2 Current Value**
	See bit 0.			
1	CH1VAL	0x0	R	**Channel 1 Current Value**
	See bit 0.			
0	CH0VAL	0x0	R	**Channel 0 Current Value**
	Sample the current output value of channel 0. This value may be one or two clock delayed			

12.5.5 PRS_SYNC_PEEK - Sync Channel Values

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													R	R	R	R
Name																													CH3VAL	CH2VAL	CH1VAL	CH0VAL

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	CH3VAL Channel 3 current value	0x0	R	**Channel Value**
2	CH2VAL Channel 2 current value	0x0	R	**Channel Value**
1	CH1VAL Channel 1 current value	0x0	R	**Channel Value**
0	CH0VAL Channel 0 current value	0x0	R	**Channel Value**

12.5.6 PRS_ASYNC_CHx_CTRL - Async Channel Control Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0								0xC					0x0												0x0		
Access					RW								RW					RW												RW		
Name					AUXSEL								FNSEL					SOURCESEL												SIGSEL		

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:24	AUXSEL	0x0	RW	**Aux Select**
	Select Asynchronous PRS channel as input B of LUT function. Async PRS[n] is selected with AUXSEL = n.			
23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	FNSEL	0xC	RW	**Function Select**
	Look up table function select. Signal A is the selected producer input. Signal B is the output of the previous PRS channel.			

Value	Mode	Description
0	LOGICAL_ZERO	Logical 0
1	A_NOR_B	A NOR B
2	NOT_A_AND_B	(!A) AND B
3	NOT_A	!A
4	A_AND_NOT_B	A AND (!B)
5	NOT_B	!B
6	A_XOR_B	A XOR B
7	A_NAND_B	A NAND B
8	A_AND_B	A AND B
9	A_XNOR_B	A XNOR B
10	B	B
11	NOT_A_OR_B	(!A) OR B
12	A	A
13	A_OR_NOT_B	A OR (!B)
14	A_OR_B	A OR B
15	LOGICAL_ONE	Logical 1

15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14:8	SOURCESEL	0x0	RW	**Source Select**

Bit	Name	Reset	Access	Description
	Select input source for asynchronous PRS channel. See Asynchronous Producers table for details.			
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	SIGSEL	0x0	RW	**Signal Select**
	Select signal input for asynchronous PRS channel. See Asynchronous Producers table for details.			

Value	Mode	Description
0	NONE	

12.5.7 PRS_SYNC_CHx_CTRL - Sync Channel Control Register

Offset	Bit Position																													
0x048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0										0x0	
Access																					RW										RW	
Name																					SOURCESEL										SIGSEL	

Bit	Name	Reset	Access	Description
31:15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14:8	SOURCESEL	0x0	RW	**Source Select**
	Select input source to sync PRS channel.			
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	SIGSEL	0x0	RW	**Signal Select**
	Select signal input to sync PRS channel.			

Value	Mode	Description
0	NONE	

12.5.8 PRS_CONSUMER_CMU_CALDN - CALDN Consumer Register

Offset	Bit Position																															
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CALDN async channel select**
	CALDN async channel select			

12.5.9 PRS_CONSUMER_CMU_CALUP - CALUP Consumer Register

Offset	Bit Position																															
0x05C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CALUP async channel select**
	CALUP async channel select			

12.5.10 PRS_CONSUMER_EUSART0_CLK - CLK Consumer Register

Offset	Bit Position																															
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CLK async channel select**
	CLK async channel select			

12.5.11 PRS_CONSUMER_EUSART0_RX - RX Consumer Register

Offset	Bit Position																															
0x064	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**RX async channel select**
	RX async channel select			

12.5.12 PRS_CONSUMER_EUSART0_TRIGGER - TRIGGER Consumer Register

Offset	Bit Position																															
0x068	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													RW		
Name																													PRSSEL		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TRIGGER async channel select**
	TRIGGER async channel select			

12.5.13 PRS_CONSUMER_EUSART1_CLK - CLK Consumer Register

Offset	Bit Position																															
0x06C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													RW		
Name																													PRSSEL		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CLK async channel select**
	CLK async channel select			

12.5.14 PRS_CONSUMER_EUSART1_RX - RX Consumer Register

Offset	Bit Position																															
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**RX async channel select**
	RX async channel select			

12.5.15 PRS_CONSUMER_EUSART1_TRIGGER - TRIGGER Consumer Register

Offset	Bit Position																															
0x074	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TRIGGER async channel select**
	TRIGGER async channel select			

12.5.16 PRS_CONSUMER_EUSART2_CLK - CLK Consumer Register

Offset	Bit Position																															
0x078	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CLK async channel select**
	CLK async channel select			

12.5.17 PRS_CONSUMER_EUSART2_RX - RX Consumer Register

Offset	Bit Position																															
0x07C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**RX async channel select**
	RX async channel select			

12.5.18 PRS_CONSUMER_EUSART2_TRIGGER - TRIGGER Consumer Register

Offset	Bit Position																													
0x080	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													RW		
Name																													PRSSEL		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TRIGGER async channel select**
	TRIGGER async channel select			

12.5.19 PRS_CONSUMER_IADC0_SCANTRIGGER - SCAN Consumer Register

Offset	Bit Position																													
0x088	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0		
Access																							RW						RW		
Name																							SPRSSEL						PRSSEL		

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**SCAN sync channel select**
	SCAN sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**SCAN async channel select**
	SCAN async channel select			

12.5.20 PRS_CONSUMER_IADC0_SINGLETRIGGER - SINGLE Consumer Register

Offset	Bit Position																													
0x08C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**SINGLE sync channel select**
	SINGLE sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**SINGLE async channel select**
	SINGLE async channel select			

12.5.21 PRS_CONSUMER_LDMAXBAR_DMAREQ0 - DMAREQ0 Consumer Register

Offset	Bit Position																													
0x090	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													RW		
Name																													PRSSEL		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DMAREQ0 async channel select**
	DMAREQ0 async channel select			

12.5.22 PRS_CONSUMER_LDMAXBAR_DMAREQ1 - DMAREQ1 Consumer Register

Offset	Bit Position																															
0x094	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DMAREQ1 async channel select**
	DMAREQ1 async channel select			

12.5.23 PRS_CONSUMER_LESENSE_START - START Consumer Register

Offset	Bit Position																															
0x0A8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**START async channel select**
	START async channel select			

12.5.24 PRS_CONSUMER_LETIMER0_CLEAR - CLEAR Consumer Register

Offset	Bit Position																															
0x0AC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CLEAR async channel select**
	CLEAR async channel select			

12.5.25 PRS_CONSUMER_LETIMER0_START - START Consumer Register

Offset	Bit Position																															
0x0B0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**START async channel select**
	START async channel select			

12.5.26 PRS_CONSUMER_LETIMER0_STOP - STOP Consumer Register

Offset	Bit Position																															
0x0B4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**STOP async channel select**
	STOP async channel select			

12.5.27 PRS_CONSUMER_PCNT0_S0IN - S0IN Consumer Register

Offset	Bit Position																															
0x0BC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**S0IN async channel select**
	S0IN async channel select			

12.5.28 PRS_CONSUMER_PCNT0_S1IN - S1IN Consumer Register

Offset	Bit Position																															
0x0C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**S1IN async channel select**
	S1IN async channel select			

12.5.29 PRS_CONSUMER_SETAMPER_TAMPERSRC25 - TAMPERSRC25 Consumer Register

Offset	Bit Position																															
0x114	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TAMPERSRC25 async channel select**
	TAMPERSRC25 async channel select			

12.5.30 PRS_CONSUMER_SETAMPER_TAMPERSRC26 - TAMPERSRC26 Consumer Register

Offset	Bit Position																															
0x118	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TAMPERSRC26 async channel select**
	TAMPERSRC26 async channel select			

12.5.31 PRS_CONSUMER_SETAMPER_TAMPERSRC27 - TAMPERSRC27 Consumer Register

Offset	Bit Position																															
0x11C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TAMPERSRC27 async channel select**
	TAMPERSRC27 async channel select			

12.5.32 PRS_CONSUMER_SETAMPER_TAMPERSRC28 - TAMPERSRC28 Consumer Register

Offset	Bit Position																															
0x120	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TAMPERSRC28 async channel select**
	TAMPERSRC28 async channel select			

12.5.33 PRS_CONSUMER_SETAMPER_TAMPERSRC29 - TAMPERSRC29 Consumer Register

Offset	Bit Position																															
0x124	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TAMPERSRC29 async channel select**
	TAMPERSRC29 async channel select			

12.5.34 PRS_CONSUMER_SETAMPER_TAMPERSRC30 - TAMPERSRC30 Consumer Register

Offset	Bit Position																															
0x128	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TAMPERSRC30 async channel select**
	TAMPERSRC30 async channel select			

12.5.35 PRS_CONSUMER_SETAMPER_TAMPERSRC31 - TAMPERSRC31 Consumer Register

Offset	Bit Position																															
0x12C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TAMPERSRC31 async channel select**
	TAMPERSRC31 async channel select			

12.5.36 PRS_CONSUMER_SYSRTC0_IN0 - IN0 Consumer Register

Offset	Bit Position																															
0x130	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**IN0 async channel select**
	IN0 async channel select			

12.5.37 PRS_CONSUMER_SYSRTC0_IN1 - IN1 Consumer Register

Offset	Bit Position																															
0x134	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**IN1 async channel select**
	IN1 async channel select			

12.5.38 PRS_CONSUMER_HFXO0_OSCREQ - OSCREQ Consumer Register

Offset	Bit Position																															
0x138	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**OSC async channel select**
	OSC async channel select			

12.5.39 PRS_CONSUMER_HFXO0_TIMEOUT - TIMEOUT Consumer Register

Offset	Bit Position																															
0x13C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TIMEOUT async channel select**
	TIMEOUT async channel select			

12.5.40 PRS_CONSUMER_CORE_CTIIN0 - CTI0 Consumer Selection

Offset	Bit Position																															
0x140	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CTI async channel select**
	CTI async channel select			

12.5.41 PRS_CONSUMER_CORE_CTIIN1 - CTI1 Consumer Selection

Offset	Bit Position																															
0x144	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CTI async channel select**
	CTI async channel select			

12.5.42 PRS_CONSUMER_CORE_CTIIN2 - CTI2 Consumer Selection

Offset	Bit Position																															
0x148	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CTI async channel select**
	CTI async channel select			

12.5.43 PRS_CONSUMER_CORE_CTIIN3 - CTI3 Consumer Selection

Offset	Bit Position																															
0x14C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CTI async channel select**
	CTI async channel select			

12.5.44 PRS_CONSUMER_CORE_M33RXEV - M33 Consumer Selection

Offset	Bit Position																													
0x150	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													RW		
Name																													PRSSEL		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**M33 async channel select**
	M33 async channel select			

12.5.45 PRS_CONSUMER_TIMER0_CC0 - CC0 Consumer Register

Offset	Bit Position																													
0x154	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0		
Access																							RW						RW		
Name																							SPRSSEL						PRSSEL		

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC0 sync channel select**
	CC0 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC0 async channel select**
	CC0 async channel select			

12.5.46 PRS_CONSUMER_TIMER0_CC1 - CC1 Consumer Register

Offset	Bit Position																													
0x158	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC1 sync channel select**
	CC1 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC1 async channel select**
	CC1 async channel select			

12.5.47 PRS_CONSUMER_TIMER0_CC2 - CC2 Consumer Register

Offset	Bit Position																													
0x15C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC2 sync channel select**
	CC2 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC2 async channel select**
	CC2 async channel select			

12.5.48 PRS_CONSUMER_TIMER0_DTI - DTI Consumer Register

Offset	Bit Position																															
0x160	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

12.5.49 PRS_CONSUMER_TIMER0_DTIFS1 - DTI Consumer Register

Offset	Bit Position																															
0x164	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

12.5.50 PRS_CONSUMER_TIMER0_DTIFS2 - DTI Consumer Register

Offset	Bit Position																															
0x168	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

12.5.51 PRS_CONSUMER_TIMER1_CC0 - CC0 Consumer Register

Offset	Bit Position																															
0x16C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC0 sync channel select**
	CC0 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC0 async channel select**
	CC0 async channel select			

12.5.52 PRS_CONSUMER_TIMER1_CC1 - CC1 Consumer Register

Offset	Bit Position																													
0x170	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC1 sync channel select**
	CC1 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC1 async channel select**
	CC1 async channel select			

12.5.53 PRS_CONSUMER_TIMER1_CC2 - CC2 Consumer Register

Offset	Bit Position																													
0x174	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC2 sync channel select**
	CC2 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC2 async channel select**
	CC2 async channel select			

12.5.54 PRS_CONSUMER_TIMER1_DTI - DTI Consumer Register

Offset	Bit Position																															
0x178	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

12.5.55 PRS_CONSUMER_TIMER1_DTIFS1 - DTI Consumer Register

Offset	Bit Position																															
0x17C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

12.5.56 PRS_CONSUMER_TIMER1_DTIFS2 - DTI Consumer Register

Offset	Bit Position																															
0x180	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

12.5.57 PRS_CONSUMER_TIMER2_CC0 - CC0 Consumer Register

Offset	Bit Position																															
0x184	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC0 sync channel select**
	CC0 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC0 async channel select**
	CC0 async channel select			

12.5.58 PRS_CONSUMER_TIMER2_CC1 - CC1 Consumer Register

Offset	Bit Position																													
0x188	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC1 sync channel select**
	CC1 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC1 async channel select**
	CC1 async channel select			

12.5.59 PRS_CONSUMER_TIMER2_CC2 - CC2 Consumer Register

Offset	Bit Position																													
0x18C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC2 sync channel select**
	CC2 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC2 async channel select**
	CC2 async channel select			

12.5.60 PRS_CONSUMER_TIMER2_DTI - DTI Consumer Register

Offset	Bit Position																															
0x190	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

12.5.61 PRS_CONSUMER_TIMER2_DTIFS1 - DTI Consumer Register

Offset	Bit Position																															
0x194	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

12.5.62 PRS_CONSUMER_TIMER2_DTIFS2 - DTI Consumer Register

Offset	Bit Position																															
0x198	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

12.5.63 PRS_CONSUMER_TIMER3_CC0 - CC0 Consumer Register

Offset	Bit Position																															
0x19C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC0 sync channel select**
	CC0 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC0 async channel select**
	CC0 async channel select			

12.5.64 PRS_CONSUMER_TIMER3_CC1 - CC1 Consumer Register

Offset	Bit Position																													
0x1A0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC1 sync channel select**
	CC1 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC1 async channel select**
	CC1 async channel select			

12.5.65 PRS_CONSUMER_TIMER3_CC2 - CC2 Consumer Register

Offset	Bit Position																													
0x1A4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC2 sync channel select**
	CC2 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC2 async channel select**
	CC2 async channel select			

12.5.66 PRS_CONSUMER_TIMER3_DTI - DTI Consumer Register

Offset	Bit Position																															
0x1A8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													RW		
Name																													PRSSEL		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

12.5.67 PRS_CONSUMER_TIMER3_DTIFS1 - DTI Consumer Register

Offset	Bit Position																															
0x1AC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													RW		
Name																													PRSSEL		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

12.5.68 PRS_CONSUMER_TIMER3_DTIFS2 - DTI Consumer Register

Offset	Bit Position																															
0x1B0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

12.5.69 PRS_CONSUMER_TIMER4_CC0 - CC0 Consumer Register

Offset	Bit Position																															
0x1B4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC0 sync channel select**
	CC0 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC0 async channel select**
	CC0 async channel select			

12.5.70 PRS_CONSUMER_TIMER4_CC1 - CC1 Consumer Register

Offset	Bit Position																													
0x1B8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC1 sync channel select**
	CC1 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC1 async channel select**
	CC1 async channel select			

12.5.71 PRS_CONSUMER_TIMER4_CC2 - CC2 Consumer Register

Offset	Bit Position																													
0x1BC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC2 sync channel select**
	CC2 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC2 async channel select**
	CC2 async channel select			

12.5.72 PRS_CONSUMER_TIMER4_DTI - DTI Consumer Register

Offset	Bit Position																															
0x1C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

12.5.73 PRS_CONSUMER_TIMER4_DTIFS1 - DTI Consumer Register

Offset	Bit Position																															
0x1C4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

12.5.74 PRS_CONSUMER_TIMER4_DTIFS2 - DTI Consumer Register

Offset	Bit Position																															
0x1C8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

12.5.75 PRS_CONSUMER_USART0_CLK - CLK Consumer Register

Offset	Bit Position																															
0x1CC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CLK async channel select**
	CLK async channel select			

12.5.76 PRS_CONSUMER_USART0_IR - IR Consumer Register

Offset	Bit Position																															
0x1D0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**IR async channel select**
	IR async channel select			

12.5.77 PRS_CONSUMER_USART0_RX - RX Consumer Register

Offset	Bit Position																															
0x1D4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**RX async channel select**
	RX async channel select			

12.5.78 PRS_CONSUMER_USART0_TRIGGER - TRIGGER Consumer Register

Offset	Bit Position																															
0x1D8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													RW		
Name																													PRSSEL		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TRIGGER async channel select**
	TRIGGER async channel select			

12.5.79 PRS_CONSUMER_VDAC0_ASYNCTRIGCH0 - ASYNCTRIG Consumer Register

Offset	Bit Position																															
0x1E8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													RW		
Name																													PRSSEL		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**ASYNCTRIG async channel select**
	ASYNCTRIG async channel select			

12.5.80 PRS_CONSUMER_VDAC0_ASYNCTRIGCH1 - ASYNCTRIG Consumer Register

Offset	Bit Position																													
0x1EC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													RW		
Name																													PRSSEL		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**ASYNCTRIG async channel select**
	ASYNCTRIG async channel select			

12.5.81 PRS_CONSUMER_VDAC0_SYNCTRIGCH0 - SYNCTRIG Consumer Register

Offset	Bit Position																													
0x1F0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0									
Access																							RW									
Name																							SPRSSEL									

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**SYNCTRIG sync channel select**
	SYNCTRIG sync channel select			
7:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

12.5.82 PRS_CONSUMER_VDAC0_SYNCTRIGCH1 - SYNCTRIG Consumer Register

Offset	Bit Position																													
0x1F4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0								
Access																							RW								
Name																							SPRSSEL								

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**SYNCTRIG sync channel select**
	SYNCTRIG sync channel select			
7:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

12.5.83 PRS_CONSUMER_WDOG0_SRC0 - SRC0 Consumer Register

Offset	Bit Position																													
0x1F8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													RW		
Name																													PRSSEL		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**SRC0 async channel select**
	SRC0 async channel select			

12.5.84 PRS_CONSUMER_WDOG0_SRC1 - SRC1 Consumer Register

Offset	Bit Position																															
0x1FC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													RW		
Name																													PRSSEL		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**SRC1 async channel select**
	SRC1 async channel select			

12.5.85 PRS_CONSUMER_WDOG1_SRC0 - SRC0 Consumer Register

Offset	Bit Position																															
0x200	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													RW		
Name																													PRSSEL		

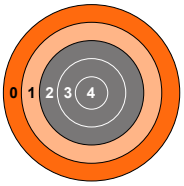
Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**SRC0 async channel select**
	SRC0 async channel select			

12.5.86 PRS_CONSUMER_WDOG1_SRC1 - SRC1 Consumer Register

Offset	Bit Position																															
0x204	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**SRC1 async channel select**
	SRC1 async channel select			

13. GPCRC - General Purpose Cyclic Redundancy Check



Quick Facts

What?

The GPCRC is an error-detecting module commonly used in digital networks and storage systems to detect accidental changes to data.

Why?

The GPCRC module can detect errors in data, giving a higher system reliability and robustness.

How?

Blocks of data entering GPCRC module can have a short checksum, based on the remainder of a polynomial division of their contents; on retrieval the calculation is repeated, and corrective action can be taken against presumed data corruption if the check values do not match.

13.1 Introduction

The GPCRC module implements a Cyclic Redundancy Check (CRC) function. It supports both 32-bit and 16-bit polynomials. The supported 32-bit polynomial is 0x04C11DB7(IEEE 802.3), while the 16-bit polynomial can be programmed to any value, depending on the needs of the application. Common 16-bit polynomials are 0x1021 (CCITT-16), 0x3D65 (IEC16-MBus), and 0x8005 (zigbee, 802.15.4, and USB).

13.2 Features

- Programmable 16-bit polynomial, fixed 32-bit polynomial
- Byte-level bit reversal for the CRC input
- Byte-order reorientation for the CRC input
- Word or half-word bit reversal of the CRC result
- Ability to configure and seed an operation in a single register write
- Single-cycle CRC computation for 32-, 16-, or 8-bit blocks
- DMA operation

13.3 Functional Description

An overview of the GPCRC module is shown in Figure 13.1 GPCRC Overview on page 402.

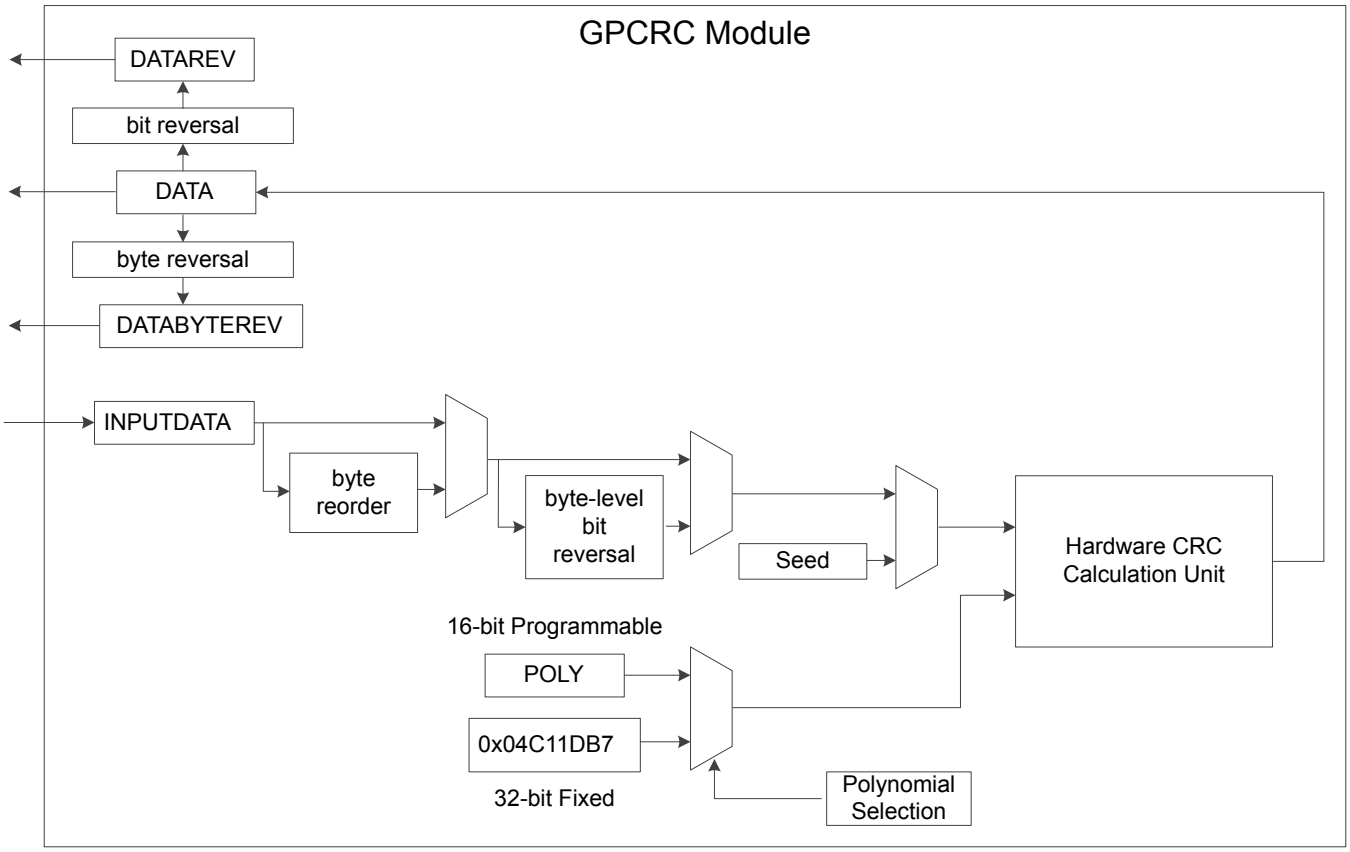


Figure 13.1. GPCRC Overview

13.3.1 Polynomial Specification

POLYSEL in GPCRC_CTRL selects between 32-bit and 16-bit polynomial functions. When a 32-bit polynomial is selected, the fixed IEEE 802.3 polynomial(0x04C11DB7) is used. When a 16-bit polynomial is selected, any valid polynomial can be defined by the user in GPCRC_POLY.

A valid 16-bit CRC polynomial must have an x^16 term and an x^0 term. Theoretically, a 16-bit polynomial has 17 terms total. The convention used is to omit the x^16 term. The polynomial should be written in **reversed** (little endian) bit order. The most significant bit corresponds to the lowest order term. Thus, the most significant bit in CRC_POLY represents the x^0 term, and the least significant bit in CRC_POLY represents the x^15 term. The highest significant bit of CRC_POLY should always set to 1.The polynomial representation for the CRC-16-CCIT polynomial x^16 + x^12 + x^5 + 1, or 0x8408 in reversed order, is shown in Figure 13.2 Polynomial Representation on page 403.

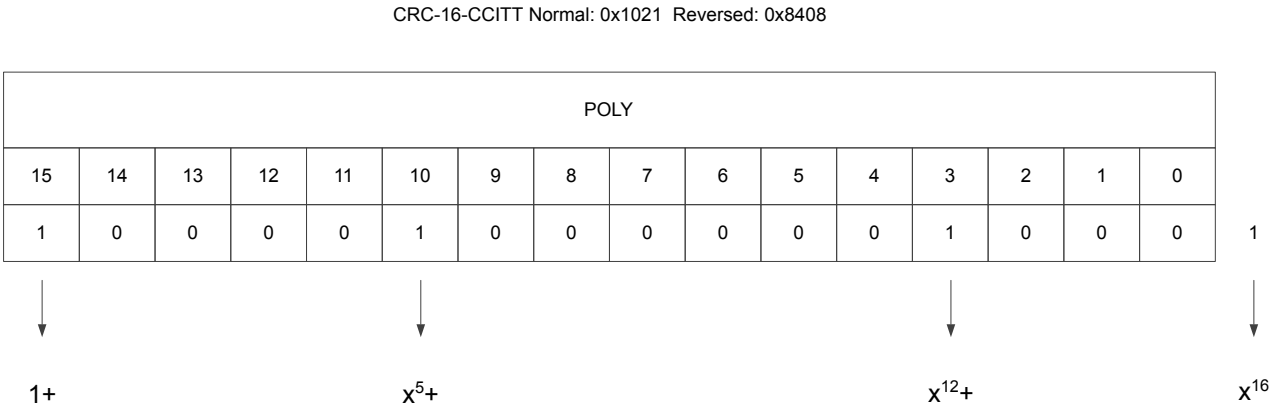


Figure 13.2. Polynomial Representation

13.3.2 Input and Output Specification

The CRC input data can be written to the GPCRC_INPUTDATA, GPCRC_INPUTDATAHWORD or GPCRC_INPUTDATABYTE register via the APB bus based on different data size. If BYTEMODE in GPCRC_CTRL is set, only the least significant byte of the data word will be used for the CRC calculation no matter which input register is written. There are also three output registers for different ordering. Reading from GPCRC_DATA will get the result based on the polynomial in reversed order, while reading from GPCRC_DATAREV will get the result based on the polynomial in normal order. The CRC calculation completes in one clock cycle. Reads from the GPCRC_DATA, GPCRC_DATAREV or GPCRC_DATABYTEREV registers and writes to the GPCRC_CMD register are halted while the calculation is in progress.

.

13.3.3 Initialization

The CRC can be pre-loaded or re-initialized by first writing a 32-bit programmable init value to INIT in GPCRC_INIT and then setting INIT in GPCRC_CMD. It can also be re-initialized automatically when read from DATA, DATAREV or DATABYTEREV provided that AUTOINIT in GPCRC_CTRL is set, the CRC would be re-initialized with the stored init value.

13.3.4 DMA Usage

A DMA channel may be used to transfer data into the CRC engine. All bytes and half-word writes must be word-aligned. The recommended DMA usage model is to use the DMA to transfer all available words of data and use software writes to capture any remaining bytes.

13.3.5 Byte-Level Bit Reversal and Byte Reordering

The byte-level bit reversal and byte reordering operations occur before the data is used in the CRC calculation. Byte reordering can occur on words or half words. The hardware ignores the BYTEREVERSE field with any byte writes or operations with byte mode enabled (BYTEMODE = 1), but the bit reversal settings (BITREVERSE) are still applied to the byte. 32-bit little endian MSB-first data can be treated like 32-bit little endian LSB-first data, as shown in Figure 13.3 Data Ordering Example - 32-bit MSB -first to LSB-first on page 404. In this example, 32-bit data is written to GPCRC_INPUTDATA, BYTEREVERSE is set for byte ordering, and BITREVERSE is set for byte-level bit reversal.

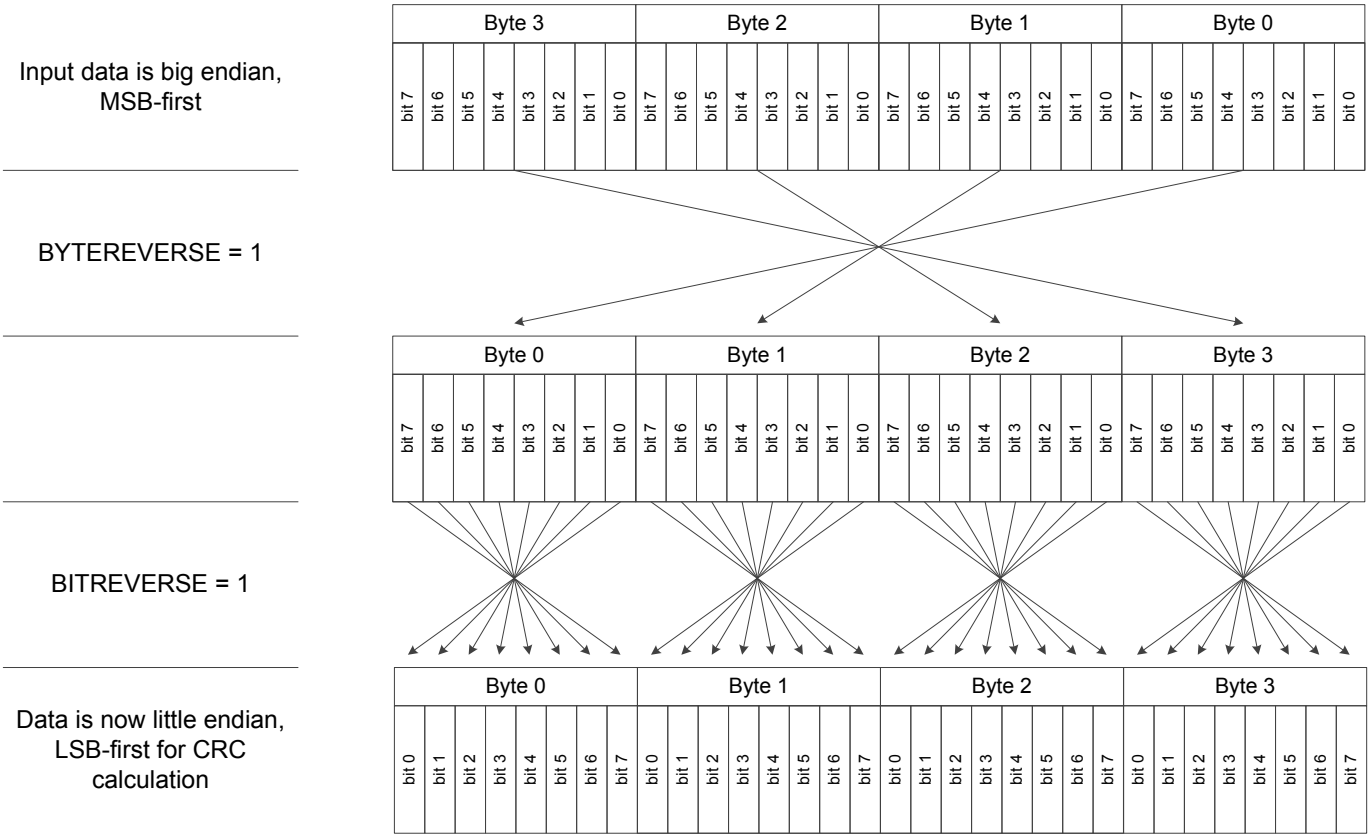


Figure 13.3. Data Ordering Example - 32-bit MSB -first to LSB-first

When handling 16-bit data, the byte reordering function only swap the two lowest bytes and clear the two highest bytes, as shown in Figure 13.4 Data Ordering Example - 16-bit MSB -first to LSB-first on page 405. In this example, 16-bit data is written to GPCRC_INPUTDATAHWORD, BYTEREVERSE is set for byte ordering, and BITREVERSE is set for byte-level bit reversal.

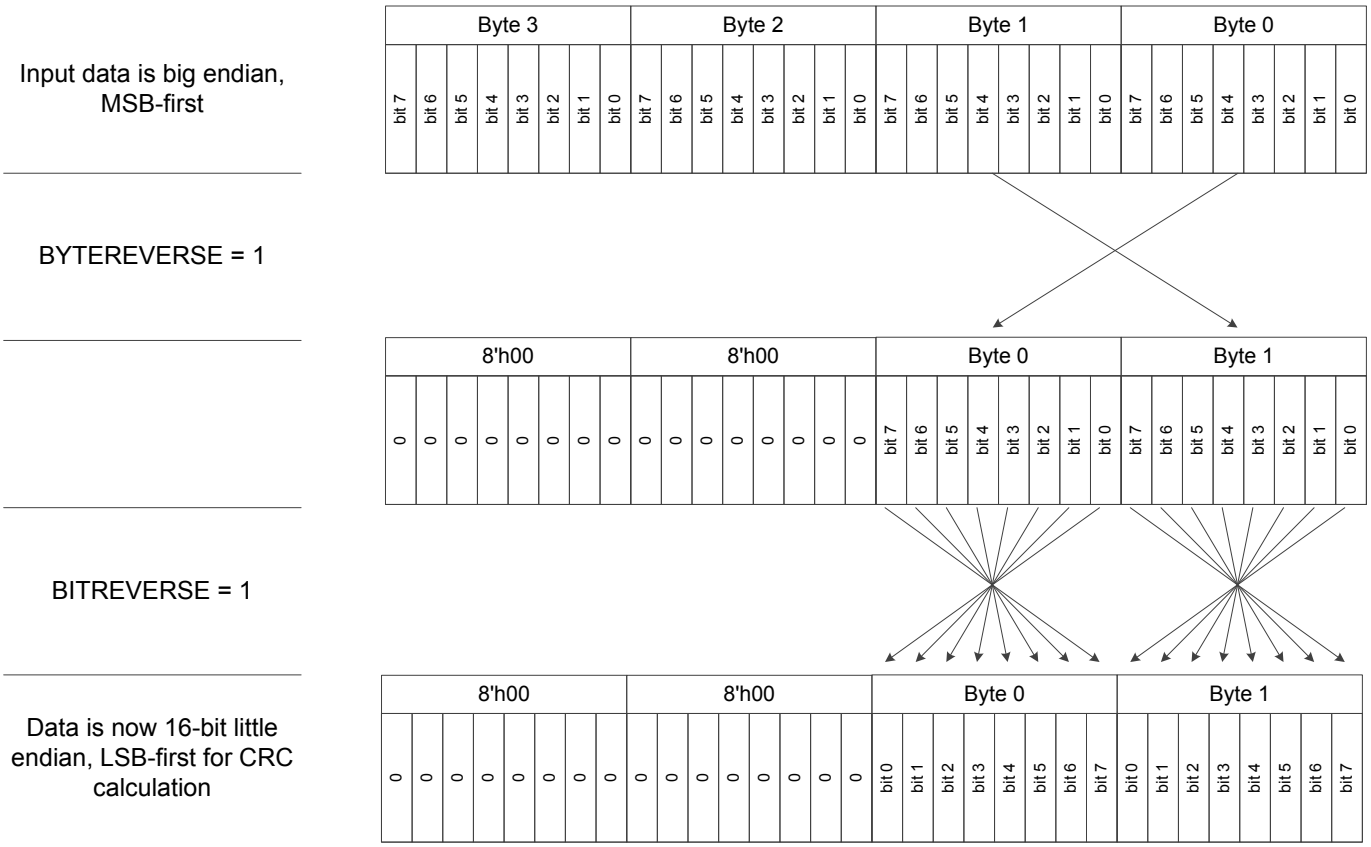


Figure 13.4. Data Ordering Example - 16-bit MSB -first to LSB-first

Assuming a word input byte order of B3 B2 B1 B0, the values used in the CRC calculation for the various settings of the byte-level bit reversal and byte reordering are shown in Table 13.1 Byte-Level Bit Reversal and Byte Reordering Results (B3 B2 B1 B0 Input Order) on page 405.

Table 13.1. Byte-Level Bit Reversal and Byte Reordering Results (B3 B2 B1 B0 Input Order)

Input Width(bits)	BYTEREVERSE Setting	BITREVERSE Setting	Input to CRC Calculation
32	0	0	B3 B2 B1 B0
32	1	1	'B0 'B1 'B2 'B3
32	1	0	B0 B1 B2 B3
32	0	1	'B3 'B2 'B1 'B0
16	0	0	XX XX B1 B0
16	1	1	XX XX 'B0 'B1
16	1	0	XX XX B0 B1
16	0	1	XX XX 'B1 'B0
8	-	0	XX XX XX XX B0
8	-	1	XX XX XX XX 'B0

Input Width(bits)	BYTEREVERSE Setting	BITREVERSE Setting	Input to CRC Calculation

Notes:

1. X indicates a "don't care".
2. Bn is the byte field within the word.
3. 'Bn is the bit-reversed byte field within the word.

13.4 GPCRC Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	GPCRC_IPVERSION	R	IP Version ID
0x004	GPCRC_EN	RW	CRC Enable
0x008	GPCRC_CTRL	RW	Control Register
0x00C	GPCRC_CMD	W	Command Register
0x010	GPCRC_INIT	RWH	CRC Init Value
0x014	GPCRC_POLY	RW	CRC Polynomial Value
0x018	GPCRC_INPUTDATA	W	Input 32-Bit Data Register
0x01C	GPCRC_INPUTDATAHWORD	W	Input 16-Bit Data Register
0x020	GPCRC_INPUTDATABYTE	W	Input 8-Bit Data Register
0x024	GPCRC_DATA	RH(r)	CRC Data Register
0x028	GPCRC_DATAREV	RH(r)	CRC Data Reverse Register
0x02C	GPCRC_DATABYTEREV	RH(r)	CRC Data Byte Reverse Register
0x1000	GPCRC_IPVERSION_SET	R	IP Version ID
0x1004	GPCRC_EN_SET	RW	CRC Enable
0x1008	GPCRC_CTRL_SET	RW	Control Register
0x100C	GPCRC_CMD_SET	W	Command Register
0x1010	GPCRC_INIT_SET	RWH	CRC Init Value
0x1014	GPCRC_POLY_SET	RW	CRC Polynomial Value
0x1018	GPCRC_INPUTDATA_SET	W	Input 32-Bit Data Register
0x101C	GPCRC_INPUTDATAH-WORD_SET	W	Input 16-Bit Data Register
0x1020	GPCRC_INPUTDATA-BYTE_SET	W	Input 8-Bit Data Register
0x1024	GPCRC_DATA_SET	RH(r)	CRC Data Register
0x1028	GPCRC_DATAREV_SET	RH(r)	CRC Data Reverse Register
0x102C	GPCRC_DATABYTEREV_SET	RH(r)	CRC Data Byte Reverse Register
0x2000	GPCRC_IPVERSION_CLR	R	IP Version ID
0x2004	GPCRC_EN_CLR	RW	CRC Enable
0x2008	GPCRC_CTRL_CLR	RW	Control Register
0x200C	GPCRC_CMD_CLR	W	Command Register
0x2010	GPCRC_INIT_CLR	RWH	CRC Init Value
0x2014	GPCRC_POLY_CLR	RW	CRC Polynomial Value
0x2018	GPCRC_INPUTDATA_CLR	W	Input 32-Bit Data Register
0x201C	GPCRC_INPUTDATAH-WORD_CLR	W	Input 16-Bit Data Register
0x2020	GPCRC_INPUTDATA-BYTE_CLR	W	Input 8-Bit Data Register

Offset	Name	Type	Description
0x2024	GPCRC_DATA_CLR	RH(r)	CRC Data Register
0x2028	GPCRC_DATAREV_CLR	RH(r)	CRC Data Reverse Register
0x202C	GPCRC_DATABYTEREV_CLR	RH(r)	CRC Data Byte Reverse Register
0x3000	GPCRC_IPVERSION_TGL	R	IP Version ID
0x3004	GPCRC_EN_TGL	RW	CRC Enable
0x3008	GPCRC_CTRL_TGL	RW	Control Register
0x300C	GPCRC_CMD_TGL	W	Command Register
0x3010	GPCRC_INIT_TGL	RWH	CRC Init Value
0x3014	GPCRC_POLY_TGL	RW	CRC Polynomial Value
0x3018	GPCRC_INPUTDATA_TGL	W	Input 32-Bit Data Register
0x301C	GPCRC_INPUTDATAH-WORD_TGL	W	Input 16-Bit Data Register
0x3020	GPCRC_INPUTDATA-BYTE_TGL	W	Input 8-Bit Data Register
0x3024	GPCRC_DATA_TGL	RH(r)	CRC Data Register
0x3028	GPCRC_DATAREV_TGL	RH(r)	CRC Data Reverse Register
0x302C	GPCRC_DATABYTEREV_TGL	RH(r)	CRC Data Byte Reverse Register

13.5 GPCRC Register Description

13.5.1 GPCRC_IPVERSION - IP Version ID

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x0	R	**IP Version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

13.5.2 GPCRC_EN - CRC Enable

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																EN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	EN	0x0	RW	**CRC Enable**

The ENABLE bit enables the module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit.

Value	Mode	Description
0	DISABLE	Disable CRC function. Reordering functions are still available. Only BITREVERSE and BYTEREVERSE bits are configurable in this mode.
1	ENABLE	Writes to INPUTDATA registers will result in CRC operations.

13.5.3 GPCRC_CTRL - Control Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																			0x0			0x0	0x0	0x0				0x0				
Access																			RW			RW	RW	RW				RW				
Name																			AUTOINIT			BYTEREVERSE	BITREVERSE	BYTEMODE				POLYSEL				

Bit	Name	Reset	Access	Description
31:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	AUTOINIT	0x0	RW	**Auto Init Enable**
	Enables auto init by re-seeding the CRC result based on the value in INIT after reading of DATA, DATAREV or DATABYTEREV.			
12:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	BYTEREVERSE	0x0	RW	**Byte Reverse Mode**
	Allows byte level reverse of bytes B3, B2, B1, B0 within the 32-bit data word			

Value	Mode	Description
0	NORMAL	No reverse: B3, B2, B1, B0
1	REVERSED	Reverse byte order. For 32-bit: B0, B1, B2, B3; For 16-bit: 0, 0, B0, B1

Bit	Name	Reset	Access	Description
9	BITREVERSE	0x0	RW	**Byte-level Bit Reverse Enable**
	Reverses bits within each byte of the 32-bit data word			

Value	Mode	Description
0	NORMAL	No reverse
1	REVERSED	Reverse bit order in each byte

Bit	Name	Reset	Access	Description
8	BYTEMODE	0x0	RW	**Byte Mode Enable**
	Treats all writes as bytes. Only the least significant byte of the data-word will be used for CRC calculation for all writes			
7:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	POLYSEL	0x0	RW	**Polynomial Select**
	Selects 16-bit CRC programmable polynomial or 32-bit CRC fixed polynomial			

Value	Mode	Description
0	CRC32	CRC-32 (0x04C11DB7) polynomial selected
1	CRC16	16-bit CRC programmable polynomial selected

Bit	Name	Reset	Access	Description
3:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

13.5.4 GPCRC_CMD - Command Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																W
Name																																INIT

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	INIT	0x0	W	**Initialization Enable** Writing 1 to this bit initialize the CRC by writing the INIT value in CRC_INIT to CRC_DATA.

13.5.5 GPCRC_INIT - CRC Init Value

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x0																
Access																RW																
Name																INIT																

Bit	Name	Reset	Access	Description
31:0	INIT	0x0	RW	**CRC Initialization Value** This value is loaded into CRC_DATA upon issuing the INIT command in CRC_CMD

13.5.6 GPCRC_POLY - CRC Polynomial Value

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	RW															
Name																	POLY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	POLY	0x0	RW	**CRC Polynomial Value**
	This value defines 16-bit POLY, which is used as the polynomial during the 16-bit CRC calculation. The polynomial is defined in reversed representation, meaning that the lowest degree term is in the highest bit position of POLY. Additionally, the highest degree term in the polynomial is implicit. Further examples of the CRC configuration can be found in the documentation.			

13.5.7 GPCRC_INPUTDATA - Input 32-Bit Data Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x0																
Access																W																
Name																INPUTDATA																

Bit	Name	Reset	Access	Description
31:0	INPUTDATA	0x0	W	**Input Data for 32-bit**
	CRC Input 32-bit Data can be written to this register. Each time this register is written, the CRC value is updated.			

13.5.8 GPCRC_INPUTDATAHWORD - Input 16-Bit Data Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	W															
Name																	INPUTDATAHWORD															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	INPUTDATAHWORD	0x0	W	**Input Data for 16-bit**
	CRC Input 16-bit Data can be written to this register. Each time this register is written, the CRC value is updated.			

13.5.9 GPCRC_INPUTDATABYTE - Input 8-Bit Data Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W							
Name																									INPUTDATABYTE							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	INPUTDATABYTE	0x0	W	**Input Data for 8-bit**
	CRC Input 8-bit Data can be written to this register. Each time this register is written, the CRC value is updated.			

13.5.10 GPCRC_DATA - CRC Data Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R(r)																															
Name	DATA																															

Bit	Name	Reset	Access	Description
31:0	DATA	0x0	R(r)	**CRC Data Register**
	CRC Data Register, read only. The CRC data register may still be indirectly written from software, by writing the INIT register and then issue an INITIALIZE command.			

13.5.11 GPCRC_DATAREV - CRC Data Reverse Register

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R(r)																															
Name	DATAREV																															

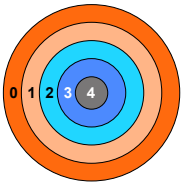
Bit	Name	Reset	Access	Description
31:0	DATAREV	0x0	R(r)	**Data Reverse Value**
	Bit reversed version of CRC Data register. When a 32-bit CRC polynomial is selected, the reversal occurs on the entire 32-bit word. When a 16-bit CRC polynomial is selected, the bits [15:0] are reversed.			

13.5.12 GPCRC_DATABYTEREV - CRC Data Byte Reverse Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R(r)																															
Name	DATABYTEREV																															

Bit	Name	Reset	Access	Description
31:0	DATABYTEREV	0x0	R(r)	**Data Byte Reverse Value**
	Byte reversed version of CRC Data register. When a 32-bit CRC polynomial is selected, the bytes are swizzled to {B0, B1, B2, B3}. When a 16-bit CRC polynomial is selected, the bytes are swizzled to {0, 0, B0, B1}.			

14. SYSRTC - System RTC



Quick Facts

What?

The System Real-Time Counter (SYSRTC) is a 32-bit Real Time Clock ensuring timekeeping in low energy modes.

Why?

Timekeeping over long time periods while using as little power as possible is required in many low power applications.

How?

The SYSRTC operates a central counter off a low-frequency oscillator source. It has configurable compare and capture channels which can be used to trigger wake-up, generate PRS signalling, or capture system events. The SYSRTC allows the system to stay in low energy modes for long periods of time and maintain reliable timekeeping.

14.1 Introduction

The SYSRTC (System Real Time Counter), is a 32-bit counter kept running down to energy mode EM3. It can be used as a sleep timer / wakeup source as well as a timekeeping counter during low energy modes. Multiple groups of capture / compare registers are available to different cores in the system, allowing the peripheral and time base to be shared across cores and save energy. Only group 0 registers are directly accessible to the main processor core, but compare / capture signals and interrupts from group 1 are also available to the main processor.

Capture compare channels can be used to trigger interrupts, generate PRS signals, capture PRS events, and to wake the device up from EM1, EM2, or EM3.

Note: Critical portions of the EFR32/EFM32 software stack related to system timing and power management make use of SYSRTC in such a way that it is effectively unavailable to user software. Please refer to relevant software documentation for additional information.

14.2 Features

- 32-bit counter
- Debug mode
- 32.768 kHz LFXO or LFRCO / 1 kHz ULFRCO
- Low energy wake-up source
- Separate groups of capture / compare registers and signals
  - 2 compare channels per group
  - 1 capture channel per group
  - Capture / compare available in PRS

14.3 Functional Description

14.3.1 Interrupts and Wake Events

Each group has a dedicated, independent interrupt line, and has the ability to wake the system. The Group 0 interrupt line (SYSRTC_APP) is used and fully controlled by user code running on the main processor. The Group 1 interrupt line (SYSRTC_SEQ) is also available to the main processor, but Group 1 registers and configuration are not directly available to the user application.

14.3.2 Counter

A single counter value (CNT) is used across all groups. The SYSRTC module is enabled by setting its EN bit field. The counter value is asserted to 0x00000000 on reset. The counter can be started/stopped by writing to the START/STOP bit fields in the CMD register. The RUNNING field in the STATUS register indicates that the counter is running when it is set.

Once started, the counter increments by 1 on each cycle (typically 32.768 kHz). The counter value can be programmed directly by writing to the CNT register. Once the programmed value is applied and the counter is running, the counter will increment on every clock starting from the newly programmed value.

When the counter reaches its maximum value of 0xFFFFFFFF, an overflow event is generated, followed by a counter wrap-around to its reset value (from which counting continues) and the OVF interrupt flag (OVFIF) on the next cycle. The overflow event is common for all the groups, i.e. OVFIF flags in all groups get set.

The normal operation of SYSRTC is to configure it, enable it, start it, and then leave it running. This should be done by a single core so that other cores only access the registers for their designated group as needed. If SYSRTC needs to be disabled, it is recommended to stop it first using the STOP command.

14.3.3 Compare Events

A compare event for channel "x" of group "n" is generated whenever the counter is RUNNING, the CMPxEN bit is set / enabled in the GRPn_CTRL register, and the CNT value is equal to the GRPn_CMPxVALUE register setting. This event is followed by GRPn_IF.CMPxIF being set on the next counter clock cycle.

Compare events can be routed as PRS producers on the GRPnOUTx signals. There are several options for the match action, selected by CMPxCMOA in GRPn_CTRL. Note that when using the PULSE option, the PRS output should already be cleared for the pulse to get set and the PULSE option should remain configured until the pulse is cleared (otherwise if the PULSE option is reprogrammed to the SET option, the "pulse" remains set). A possible use case when using the CMPIF option is to signal early events prior to the following wake-up. After wakeup, the compare flag should be processed and cleared. To avoid a race condition on the PRS output, the compare flag should be cleared away from the next possible compare event.

Note that when setting the compare value to the current counter value, a compare event may not get generated until the counter overflows and reaches the current value again. To generate a compare event quickly, it is recommended to program the compare value to the current counter value + 1. Compare events are group-specific.

14.3.4 Capture Events

SYSRTC groups support counter value capture triggered by PRS consumer signals. For group "n" the SYSRTC0 "INn" PRS consumer is used to trigger captures. Capture can be triggered on RISING, FALLING, or BOTH edges, according to the setting programmed in CAP0EDGE of the GRPn_CTRL register. A capture event for group "n" is generated whenever the counter is RUNNING, the CAP0EN bit is set / enabled in the GRPn_CTRL register, and the desired event occurs on the PRS output.

A capture event is followed by GRPn_IF_CAP0IF being set after up to 3 counter clock cycles. At the same time the flag is set, the GRPn_CAP0VALUE register captures the current counter value. Note that PRS input edges should not occur more frequently than once every three counter cycles. If the counter is being started/stopped or GRPn_CTRL.CAP0EN / GRPn_CTRL.CAP0EDGE is changed close to the PRS input edge, a race condition may occur. Capture events are group-specific.

14.3.5 SYSRTC Behavior on SWRST/Disablement/STOP

On SWRST / Disablement, the counter is reset to 0x00000000, PRS outputs are cleared, compare/capture events are disabled, compare/capture flags are reset, and the CAP0VALUE register is reset.

When the counter is stopped using the STOP command in the CMD register, all other settings remain unchanged, except that the RUNNING status will return to 0, which blocks any compare/capture events until the counter is started again.

14.3.6 Debug Functionality

By default, the counter value is frozen when the main processor is halted during debugging. The RUNNING status bit is not affected by debug halt, and will continue to indicate that the counter is active. If DEBUGRUN in the CFG register is set, the counter will not halt when the main processor is halted, and SYSRTC will continue to count clocks.

Note that the main processor runs on a much higher frequency than the counter and that the halt condition needs to last long enough (more than 3 counter cycles) for the counter to reach the frozen state.

14.4 SYSRTC Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	SYSRTC_IPVERSION	R	IP VERSION
0x004	SYSRTC_EN	RW ENABLE	Module Enable Register
0x008	SYSRTC_SWRST	RW SWRST	Software Reset Register
0x00C	SYSRTC_CFG	RW CONFIG	Configuration Register
0x010	SYSRTC_CMD	W LFSYNC	Command Register
0x014	SYSRTC_STATUS	RH	Status Register
0x018	SYSRTC_CNT	RWH LFSYNC	Counter Value Register
0x01C	SYSRTC_SYNCBUSY	RH	Synchronization Busy Register
0x020	SYSRTC_LOCK	W	Configuration Lock Register
0x040	SYSRTC_GRP0_IF	RWH INTFLAG	Group Interrupt Flags
0x044	SYSRTC_GRP0_IEN	RW	Group Interrupt Enables
0x048	SYSRTC_GRP0_CTRL	RW	Group Control Register
0x04C	SYSRTC_GRP0_CMP0VALUE	RW	Compare 0 Value Register
0x050	SYSRTC_GRP0_CMP1VALUE	RW	Compare 1 Value Register
0x054	SYSRTC_GRP0_CAP0VALUE	RH	Capture 0 Value Register
0x058	SYSRTC_GRP0_SYNCBUSY	RH	Synchronization Busy Register
0x1000	SYSRTC_IPVERSION_SET	R	IP VERSION
0x1004	SYSRTC_EN_SET	RW ENABLE	Module Enable Register
0x1008	SYSRTC_SWRST_SET	RW SWRST	Software Reset Register
0x100C	SYSRTC_CFG_SET	RW CONFIG	Configuration Register
0x1010	SYSRTC_CMD_SET	W LFSYNC	Command Register
0x1014	SYSRTC_STATUS_SET	RH	Status Register
0x1018	SYSRTC_CNT_SET	RWH LFSYNC	Counter Value Register
0x101C	SYSRTC_SYNCBUSY_SET	RH	Synchronization Busy Register
0x1020	SYSRTC_LOCK_SET	W	Configuration Lock Register
0x1040	SYSRTC_GRP0_IF_SET	RWH INTFLAG	Group Interrupt Flags
0x1044	SYSRTC_GRP0_IEN_SET	RW	Group Interrupt Enables
0x1048	SYSRTC_GRP0_CTRL_SET	RW	Group Control Register
0x104C	SYSRTC_GRP0_CMP0VAL-UE_SET	RW	Compare 0 Value Register
0x1050	SYSRTC_GRP0_CMP1VAL-UE_SET	RW	Compare 1 Value Register
0x1054	SYSRTC_GRP0_CAP0VAL-UE_SET	RH	Capture 0 Value Register
0x1058	SYSRTC_GRP0_SYN-CBUSY_SET	RH	Synchronization Busy Register
0x2000	SYSRTC_IPVERSION_CLR	R	IP VERSION

Offset	Name	Type	Description
0x2004	SYSRTC_EN_CLR	RW ENABLE	Module Enable Register
0x2008	SYSRTC_SWRST_CLR	RW SWRST	Software Reset Register
0x200C	SYSRTC_CFG_CLR	RW CONFIG	Configuration Register
0x2010	SYSRTC_CMD_CLR	W LFSYNC	Command Register
0x2014	SYSRTC_STATUS_CLR	RH	Status Register
0x2018	SYSRTC_CNT_CLR	RWH LFSYNC	Counter Value Register
0x201C	SYSRTC_SYNCBUSY_CLR	RH	Synchronization Busy Register
0x2020	SYSRTC_LOCK_CLR	W	Configuration Lock Register
0x2040	SYSRTC_GRP0_IF_CLR	RWH INTFLAG	Group Interrupt Flags
0x2044	SYSRTC_GRP0_IEN_CLR	RW	Group Interrupt Enables
0x2048	SYSRTC_GRP0_CTRL_CLR	RW	Group Control Register
0x204C	SYSRTC_GRP0_CMP0VAL-UE_CLR	RW	Compare 0 Value Register
0x2050	SYSRTC_GRP0_CMP1VAL-UE_CLR	RW	Compare 1 Value Register
0x2054	SYSRTC_GRP0_CAP0VAL-UE_CLR	RH	Capture 0 Value Register
0x2058	SYSRTC_GRP0_SYN-CBUSY_CLR	RH	Synchronization Busy Register
0x3000	SYSRTC_IPVERSION_TGL	R	IP VERSION
0x3004	SYSRTC_EN_TGL	RW ENABLE	Module Enable Register
0x3008	SYSRTC_SWRST_TGL	RW SWRST	Software Reset Register
0x300C	SYSRTC_CFG_TGL	RW CONFIG	Configuration Register
0x3010	SYSRTC_CMD_TGL	W LFSYNC	Command Register
0x3014	SYSRTC_STATUS_TGL	RH	Status Register
0x3018	SYSRTC_CNT_TGL	RWH LFSYNC	Counter Value Register
0x301C	SYSRTC_SYNCBUSY_TGL	RH	Synchronization Busy Register
0x3020	SYSRTC_LOCK_TGL	W	Configuration Lock Register
0x3040	SYSRTC_GRP0_IF_TGL	RWH INTFLAG	Group Interrupt Flags
0x3044	SYSRTC_GRP0_IEN_TGL	RW	Group Interrupt Enables
0x3048	SYSRTC_GRP0_CTRL_TGL	RW	Group Control Register
0x304C	SYSRTC_GRP0_CMP0VAL-UE_TGL	RW	Compare 0 Value Register
0x3050	SYSRTC_GRP0_CMP1VAL-UE_TGL	RW	Compare 1 Value Register
0x3054	SYSRTC_GRP0_CAP0VAL-UE_TGL	RH	Capture 0 Value Register
0x3058	SYSRTC_GRP0_SYN-CBUSY_TGL	RH	Synchronization Busy Register

14.5 SYSRTC Register Description

14.5.1 SYSRTC_IPVERSION - IP VERSION

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP VERSION**
	Gives access to the IP VERSION of SYSRTC			

14.5.2 SYSRTC_EN - Module Enable Register

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status**
	Set when EN cleared and cleared when the peripheal core reset is finished			
0	EN	0x0	RW	**SYSRTC Enable**
	Enable the SYSRTC by requesting Clk from CMU			

14.5.3 SYSRTC_SWRST - Software Reset Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	W
Name																															RESETTING	SWRST

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESETTING	0x0	R	**Software reset busy status** When SWRST command is issued, resetting logic sets this status immediately and it is later cleared when the reset process is finished
0	SWRST	0x0	W	**Software reset command** A software reset command field resets the module back to the initial condition, similar to the power-on reset condition

14.5.4 SYSRTC_CFG - Configuration Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																DEBUGRUN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	DEBUGRUN	0x0	RW	**Debug Mode Run Enable** Set this bit to keep the SYSRTC running during a debug halt.

Value	Mode	Description
0	DISABLE	SYSRTC is frozen in debug mode
1	ENABLE	SYSRTC is running in debug mode

14.5.5 SYSRTC_CMD - Command Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															W(nB)	W(nB)
Name																															STOP	START

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	STOP	0x0	W(nB)	**Stop SYSRTC**
	Write a 1 to stop the SYSRTC			
0	START	0x0	W(nB)	**Start SYSRTC**
	Write a 1 to start the SYSRTC			

14.5.6 SYSRTC_STATUS - Status Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	R
Name																															LOCKSTATUS	RUNNING

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	LOCKSTATUS	0x0	R	**Lock Status**
	Indicates the current status of SYSRTC Lock			

Value	Mode	Description
0	UNLOCKED	SYSRTC registers are unlocked
1	LOCKED	SYSRTC registers are locked

Bit	Name	Reset	Access	Description
0	RUNNING	0x0	R	**SYSRTC running status**
	Indicates the current status of SYSRTC running			

14.5.7 SYSRTC_CNT - Counter Value Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	CNT																															

Bit	Name	Reset	Access	Description
31:0	CNT	0x0	RW	**Counter Value**
	Gives access to the counter value of the SYSRTC.			

14.5.8 SYSRTC_SYNCBUSY - Synchronization Busy Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														R	R	R
Name																														CNT	STOP	START

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	CNT	0x0	R	**Sync busy for CNT bitfield**
	Last writing of CNT is synchronizing to LF clock			
1	STOP	0x0	R	**Sync busy for STOP bitfield**
	Last writing of STOP is synchronizing to LF clock			
0	START	0x0	R	**Sync busy for START bitfield**
	Last writing of START is synchronizing to LF clock			

14.5.9 SYSRTC_LOCK - Configuration Lock Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W							
Name																									LOCKKEY							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x0	W	**Configuration Lock Key**
	Write any other value than the unlock code to lock EN, SWRST, CFG, CMD, CNT registers from editing. Write the unlock code to unlock.			

Value	Mode	Description
18294	UNLOCK	Write to unlock SYSRTC lockable registers

14.5.10 SYSRTC_GRP0_IF - Group Interrupt Flags

Offset	Bit Position																													
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													RW	RW	RW	RW
Name																													CAP0	CMP1	CMP0	OVF

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	CAP0	0x0	RW	**Capture 0 Interrupt Flag**
	This bit is set when CAP0VALUE is updated			
2	CMP1	0x0	RW	**Compare 1 Interrupt Flag**
	This bit is set when counter matches COMP1VALUE			
1	CMP0	0x0	RW	**Compare 0 Interrupt Flag**
	This bit is set when counter matches COMP0VALUE			
0	OVF	0x0	RW	**Overflow Interrupt Flag**
	This bit is set when counter overflows			

14.5.11 SYSRTC_GRP0_IEN - Group Interrupt Enables

Offset	Bit Position																													
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													RW	RW	RW	RW
Name																													CAP0	CMP1	CMP0	OVF

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	CAP0 Enable CAP0 interrupt	0x0	RW	**Capture 0 Interrupt Enable**
2	CMP1 Enable CMP1 interrupt	0x0	RW	**Compare 1 Interrupt Enable**
1	CMP0 Enable CMP0 interrupt	0x0	RW	**Compare 0 Interrupt Enable**
0	OVF Enable OVF interrupt	0x0	RW	**Overflow Interrupt Enable**

14.5.12 SYSRTC_GRP0_CTRL - Group Control Register

Offset	Bit Position																															
0x048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																						0x0		0x0			0x0			0x0	0x0	0x0
Access																						RW		RW			RW			RW	RW	RW
Name																						CAP0EDGE		CMP1CMOA			CMP0CMOA			CAP0EN	CMP1EN	CMP0EN

Bit	Name	Reset	Access	Description
31:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10:9	CAP0EDGE	0x0	RW	**Capture 0 Edge Select**
	Select edge detection for Capture 0			
	Value	Mode		Description
	0	RISING		Rising edges detected
	1	FALLING		Falling edges detected
	2	BOTH		Both edges detected
8:6	CMP1CMOA	0x0	RW	**Compare 1 Compare Match Output Action**
	Select PRS output action on Compare 1 match			
	Value	Mode		Description
	0	CLEAR		Cleared on the next cycle
	1	SET		Set on the next cycle
	2	PULSE		Set on the next cycle, cleared on the cycle after
	3	TOGGLE		Inverted on the next cycle
	4	CMPIF		Export this channel's CMP IF
5:3	CMP0CMOA	0x0	RW	**Compare 0 Compare Match Output Action**
	Select PRS output action on Compare 0 match			
	Value	Mode		Description
	0	CLEAR		Cleared on the next cycle
	1	SET		Set on the next cycle
	2	PULSE		Set on the next cycle, cleared on the cycle after
	3	TOGGLE		Inverted on the next cycle
	4	CMPIF		Export this channel's CMP IF
2	CAP0EN	0x0	RW	**Capture 0 Enable**
	Set this bit to enable Capture 0			

Bit	Name	Reset	Access	Description
1	CMP1EN	0x0	RW	**Compare 1 Enable**
	Set this bit to enable Compare 1			
0	CMP0EN	0x0	RW	**Compare 0 Enable**
	Set this bit to enable Compare 0			

14.5.13 SYSRTC_GRP0_CMP0VALUE - Compare 0 Value Register

Offset	Bit Position																													
0x04C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	CMP0VALUE																															

Bit	Name	Reset	Access	Description
31:0	CMP0VALUE	0x0	RW	**Compare 0 Value**
	Compare 0 match value			

14.5.14 SYSRTC_GRP0_CMP1VALUE - Compare 1 Value Register

Offset	Bit Position																													
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	CMP1VALUE																															

Bit	Name	Reset	Access	Description
31:0	CMP1VALUE	0x0	RW	**Compare 1 Value**
	Compare 1 match value			

14.5.15 SYSRTC_GRP0_CAP0VALUE - Capture 0 Value Register

Offset	Bit Position																															
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	CAP0VALUE																															

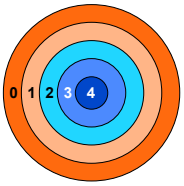
Bit	Name	Reset	Access	Description
31:0	CAP0VALUE	0x0	R	**Capture 0 Value**
	Capture 0 captured value			

14.5.16 SYSRTC_GRP0_SYNCBUSY - Synchronization Busy Register

Offset	Bit Position																															
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														R	R	R
Name																														CMP1VALUE	CMP0VALUE	CTRL

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	CMP1VALUE	0x0	R	**Sync busy for CMP1VALUE register**
	Last writing of CMP1VALUE is synchronizing to LF clock			
1	CMP0VALUE	0x0	R	**Sync busy for CMP0VALUE register**
	Last writing of CMP0VALUE is synchronizing to LF clock			
0	CTRL	0x0	R	**Sync busy for CTRL register**
	Last writing of CTRL is synchronizing to LF clock			

15. BURTC - Back-Up Real Time Counter



Quick Facts

What?

The BURTC is a 32 bit counter which operates on a low frequency oscillator, and is capable of running in all Energy Modes.

Why?

It can provide periodic Wakeup events and PRS signals which can be used to wake up peripherals from any energy mode.

The availability of the BURTC in EM4, where most of the device is powered down, makes it ideal for keeping track of time in EM4.

How?

The BURTC provides a very wide range of periods for the interrupts facilitating flexible ultra-low energy operation.

15.1 Introduction

The Back-Up Real Time Counter (BURTC) is a 32-bit counter which operates on a low frequency oscillator, and is capable of running in all Energy Modes. It can provide periodic Wakeup events and PRS signals which can be used to wake up peripherals from any energy mode. The BURTC provides a very wide range of periods for the interrupts facilitating flexible ultra-low energy operation. The availability of the BURTC in EM4, where most of the device is powered down, makes it ideal for keeping track of time in EM4. A single compare channel is available which can be used to trigger an interrupt and/or wake the device up from a low energy mode.

15.2 Features

A low frequency oscillator is used as clock signal and the BURTC with one compare channel which can trigger wake-up, generate PRS signalling, or capture system events. 32-bit resolution and selectable prescaling allows the system to stay in low energy modes for long periods of time and still maintain reliable timekeeping.

- 32-bit Real Time Counter
- 15-bit pre-counter for flexible frequency scaling of main counter
- EM2/3/4 operation and wakeup
- Reset only by External Pin and Power-On Resets
- Interrupt/wake up event after deterministic intervals
- PRS Outputs
- Debug mode
  - Configurable to either run or stop when processor is stopped (break)

15.3 Functional Description

An overview of the BURTC module is shown in Figure 15.1 BURTC Overview on page 430.

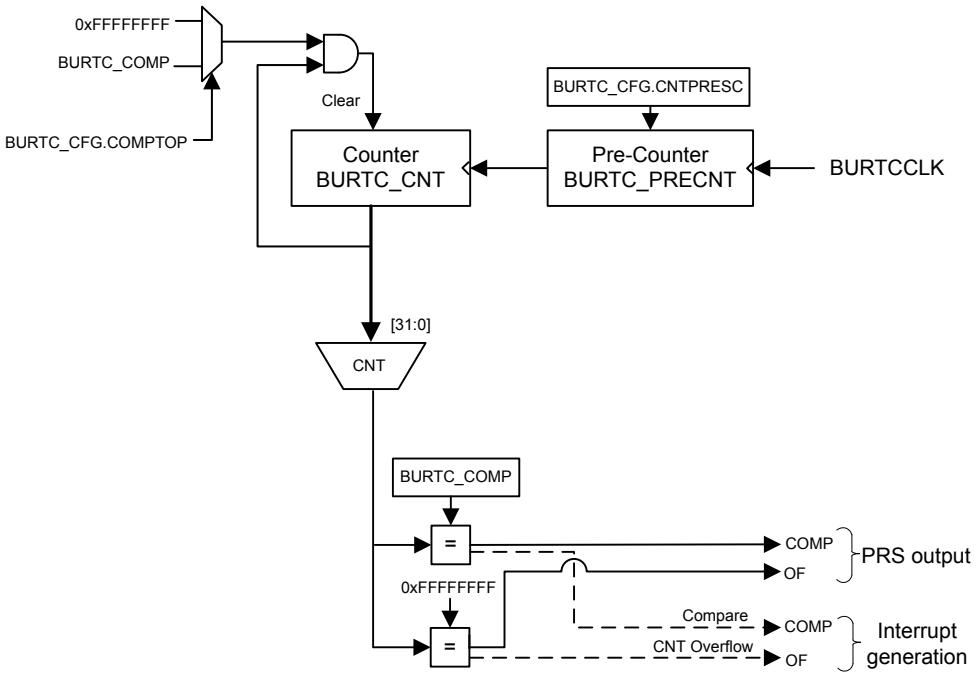


Figure 15.1. BURTC Overview

15.3.1 Clock Selection

The BURTC source clock (BURTCCLK) can be selected to be the LFXO, LFRCO, or ULFRCO by configuring the CMU_EM4GRPACLKCTRL.CLKSEL bitfield. Note that in EM3, only ULFRCO is a valid source clock.

15.3.2 Configuration

To configure and use the BURTC properly, the following programming sequence must be followed:

1. Configure any desired options in the BURTC_CFG register. Note that the BURTC_CFG register can only be written when BURTC_EN.EN = 0 - a bus fault will occur if writing BURTC_CFG register while BURTC_EN.EN = 1.
2. Set BURTC_EN.EN = 1.
3. Set BURTC_CMD.START = 1 to start the BURTC counter.

   Note: All low frequency synchronization registers can only be programmed after EN is set to 1. The BURTC counter will only start to count once START command is issued. For HV Sync registers (e.g., BURTC_CMD), the first bitfield write will occur without issue. However, on subsequent bitfield writes to HV Sync registers, the firmware needs to poll the corresponding bit in BURTC_SYNCBUSY before programming the same bitfield once again.

To stop the BURTC, set BURTC_CMD.STOP = 1

15.3.3 Debug Features and Description

By default, the BURTC is halted when code execution is halted from the debugger. By setting the DEBUGRUN bit in the BURTC_CFG register, the BURTC will continue to run even when the debugger has halted the system.

15.3.4 Counter

The BURTC consists of two counters: the 32-bit main counter, BURTC_CNT, and a 15-bit pre-counter, BURTC_PRECNT. The pre-counter is a free running counter clocked by low frequency clock, used to generate a specific frequency for the main counter. The pre-counter will be counting only when the BURTC_CFG.CNTPRESC value is set greater than 0.

The BURTC peripheral clock is requested by setting the EN bit in BURTC_EN. Then the BURTC counters can be started by setting the command register START in BURTC_CMD. When BURTC_CMD.START has been initiated and BURTC_CFG.CNTPRESC > 0, the pre-counter (BURTC_PRECNT) increments upon each positive clock edge of the BURTCCLK, wrapping around to zero when it overflows.

The main counter can be accessed in BURTC_CNT register, and counts at frequency determined by the CNTPRESC bitfiled in BURTC_CFG. Setting CNTPRESC to 0 gives the maximum resolution, with the main counter clocked at the same frequency as the BURTCCLK. When CNTPRESC > 0, the main counter increments upon each tick given from the pre-counter, allowing the main counter ticks to be power-of-2 divisions of the BURTCCLK.

The Table 15.1 BURTC Resolution vs Overflow, $F_{BURTCCLK}$ = 32768 Hz on page 431 table below shows the BURTC Resolution vs Overflow Time when using a 32768 Hz oscillator as the source clock of BURTC.

Table 15.1. BURTC Resolution vs Overflow, $F_{BURTCCLK}$ = 32768 Hz

BURTC_CFG.CNTPRESC	Main counter period, T_{CNT}	Overflow Time
DIV1	30.5 µs	36.4 hours
DIV2	61 µs	72.8 hours
DIV4	122 µs	145.6 hours
DIV8	244 µs	12 days
DIV16	488 µs	24 days
DIV32	977 µs	48 days
DIV64	1.95 ms	97 days
DIV128	3.91 ms	194 days
DIV256	7.81 ms	388 days
DIV512	15.6 ms	776 days
DIV1024	31.25 ms	4.2 years
DIV2048	62.5 ms	8.5 years
DIV4096	0.125 s	17 years
DIV8192	0.25 s	34 years
DIV16384	0.5 s	68 years
DIV32768	1 s	136 years

By default, the counter will keep counting until it reaches the top value, 0xFFFFFFFF, and then it wrap around and continue counting from zero. If COMPTOP in BURTC_CFG is set, the main counter will wrap to 0 on a Compare value match (i.e., BURTC_CNT = BURTC_COMP). If using the Compare value match, make sure to set COMPTOP prior to or at the same time the BURTC is enabled. Setting COMPTOP after enabling the BURTC will result in a bus fault error.

The counters of the BURTC, BURTC_CNT and BURTC_PRECNT, can at any time be written by software, as long as the registers are not locked using BURTC_LOCKKEY. All BURTC control registers with Sync Type HV uses the 2 FF synchronization scheme.

Note: Writing to the BURTC_PRECNT register may alter the frequency of the ticks for the BURTC_CNT register.

15.3.5 Compare Channel

A single compare channel is available in the BURTC. The compare value is set in BURTC_COMP register. If BURTC_CFG.COMPTOP is set, the main counter will clear to 0 when it matches the value set in BURTC_COMP.

15.3.6 Interrupts

The BURTC has two interrupts: one for counter overflow and another for the compare match event. Individual interrupts are enabled by BURTC_IEN register bits, and the respective bits can be used as EM2 wakeup. BURTC_EM4WUEN enables the wakeup enable from EM4 for those events.

15.3.7 Register Lock

To prevent accidental writes to the BURTC registers, the BURTC_LOCK register can be written to any other value than the unlock value. To unlock the register, write the unlock value to BURTC_LOCKKEY. Registers affected by this lock are:

- BURTC_CFG
- BURTC_EN
- BURTC_CMD
- BURTC_PRECNT
- BURTC_CNT
- BURTC_COMP
- BURTC_IEN

15.4 BURTC Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	BURTC_IPVERSION	R	IP Version ID
0x004	BURTC_EN	RW ENABLE	Module Enable Register
0x008	BURTC_CFG	RW CONFIG	Configuration Register
0x00C	BURTC_CMD	W LFSYNC	Command Register
0x010	BURTC_STATUS	RH	Status Register
0x014	BURTC_IF	RWH INTFLAG	Interrupt Flag Register
0x018	BURTC_IEN	RW	Interrupt Enable Register
0x01C	BURTC_PRECNT	RWH LFSYNC	Pre-Counter Value Register
0x020	BURTC_CNT	RWH LFSYNC	Counter Value Register
0x024	BURTC_EM4WUEN	RW	EM4 Wakeup Request Enable Register
0x028	BURTC_SYNCBUSY	RH	Synchronization Busy Register
0x02C	BURTC_LOCK	W	Configuration Lock Register
0x030	BURTC_COMP	RW LFSYNC	Compare Value Register
0x1000	BURTC_IPVERSION_SET	R	IP Version ID
0x1004	BURTC_EN_SET	RW ENABLE	Module Enable Register
0x1008	BURTC_CFG_SET	RW CONFIG	Configuration Register
0x100C	BURTC_CMD_SET	W LFSYNC	Command Register
0x1010	BURTC_STATUS_SET	RH	Status Register
0x1014	BURTC_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1018	BURTC_IEN_SET	RW	Interrupt Enable Register
0x101C	BURTC_PRECNT_SET	RWH LFSYNC	Pre-Counter Value Register
0x1020	BURTC_CNT_SET	RWH LFSYNC	Counter Value Register
0x1024	BURTC_EM4WUEN_SET	RW	EM4 Wakeup Request Enable Register
0x1028	BURTC_SYNCBUSY_SET	RH	Synchronization Busy Register
0x102C	BURTC_LOCK_SET	W	Configuration Lock Register
0x1030	BURTC_COMP_SET	RW LFSYNC	Compare Value Register
0x2000	BURTC_IPVERSION_CLR	R	IP Version ID
0x2004	BURTC_EN_CLR	RW ENABLE	Module Enable Register
0x2008	BURTC_CFG_CLR	RW CONFIG	Configuration Register
0x200C	BURTC_CMD_CLR	W LFSYNC	Command Register
0x2010	BURTC_STATUS_CLR	RH	Status Register
0x2014	BURTC_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2018	BURTC_IEN_CLR	RW	Interrupt Enable Register
0x201C	BURTC_PRECNT_CLR	RWH LFSYNC	Pre-Counter Value Register
0x2020	BURTC_CNT_CLR	RWH LFSYNC	Counter Value Register

Offset	Name	Type	Description
0x2024	BURTC_EM4WUEN_CLR	RW	EM4 Wakeup Request Enable Register
0x2028	BURTC_SYNCBUSY_CLR	RH	Synchronization Busy Register
0x202C	BURTC_LOCK_CLR	W	Configuration Lock Register
0x2030	BURTC_COMP_CLR	RW LFSYNC	Compare Value Register
0x3000	BURTC_IPVERSION_TGL	R	IP Version ID
0x3004	BURTC_EN_TGL	RW ENABLE	Module Enable Register
0x3008	BURTC_CFG_TGL	RW CONFIG	Configuration Register
0x300C	BURTC_CMD_TGL	W LFSYNC	Command Register
0x3010	BURTC_STATUS_TGL	RH	Status Register
0x3014	BURTC_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3018	BURTC_IEN_TGL	RW	Interrupt Enable Register
0x301C	BURTC_PRECNT_TGL	RWH LFSYNC	Pre-Counter Value Register
0x3020	BURTC_CNT_TGL	RWH LFSYNC	Counter Value Register
0x3024	BURTC_EM4WUEN_TGL	RW	EM4 Wakeup Request Enable Register
0x3028	BURTC_SYNCBUSY_TGL	RH	Synchronization Busy Register
0x302C	BURTC_LOCK_TGL	W	Configuration Lock Register
0x3030	BURTC_COMP_TGL	RW LFSYNC	Compare Value Register

15.5 BURTC Register Description

15.5.1 BURTC_IPVERSION - IP Version ID

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP Version ID**

15.5.2 BURTC_EN - Module Enable Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status** When EN is cleared, DISABLING is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs and FIFOs. The CNT and PRECNT count registers are not reset during disablement.
0	EN	0x0	RW	**BURTC Enable** Enable the BURTC to make the peripheral clock available to the module

15.5.3 BURTC_CFG - Configuration Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0						0x0	0x0
Access																									RW						RW	RW
Name																									CNTPRESC						COMPTOP	DEBUGRUN

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:4	CNTPRESC	0x0	RW	**Counter prescaler value.** Configure counting frequency of the CNT register

Value	Mode	Description
0	DIV1	CLK_CNT = (BURTC LF CLK)/1
1	DIV2	CLK_CNT = (BURTC LF CLK)/2
2	DIV4	CLK_CNT = (BURTC LF CLK)/4
3	DIV8	CLK_CNT = (BURTC LF CLK)/8
4	DIV16	CLK_CNT = (BURTC LF CLK)/16
5	DIV32	CLK_CNT = (BURTC LF CLK)/32
6	DIV64	CLK_CNT = (BURTC LF CLK)/64
7	DIV128	CLK_CNT = (BURTC LF CLK)/128
8	DIV256	CLK_CNT = (BURTC LF CLK)/256
9	DIV512	CLK_CNT = (BURTC LF CLK)/512
10	DIV1024	CLK_CNT = (BURTC LF CLK)/1024
11	DIV2048	CLK_CNT = (BURTC LF CLK)/2048
12	DIV4096	CLK_CNT = (BURTC LF CLK)/4096
13	DIV8192	CLK_CNT = (BURTC LF CLK)/8192
14	DIV16384	CLK_CNT = (BURTC LF CLK)/16384
15	DIV32768	CLK_CNT = (BURTC LF CLK)/32768

3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	COMPTOP	0x0	RW	**Compare Channel is Top Value** When set, the counter is cleared in the clock cycle after a compare match with compare channel

Value	Mode	Description
0	DISABLE	The top value of the BURTC is 4294967295 (0xFFFFFFFF)

Bit	Name	Reset	Access	Description
	1	ENABLE		The top value of the BURTC is given by COMP
0	DEBUGRUN	0x0	RW	**Debug Mode Run Enable**
	Set this bit to enable the BURTC to keep running in debug			

Value	Mode	Description
0	X0	BURTC is frozen in debug mode
1	X1	BURTC is running in debug mode

15.5.4 BURTC_CMD - Command Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															W(nB)	W(nB)
Name																															STOP	START

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	STOP	0x0	W(nB)	**Stop BURTC counter**
	Write a 1 to stop the BURTC counter.			
0	START	0x0	W(nB)	**Start BURTC counter**
	Write a 1 to start the BURTC counter.			

15.5.5 BURTC_STATUS - Status Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	R
Name																															LOCK	RUNNING

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	LOCK	0x0	R	**Configuration Lock Status**
	Indicates the current status of BURTC Lock			
	Value	Mode		Description
	0	UNLOCKED		All BURTC lockable registers are unlocked.
	1	LOCKED		All BURTC lockable registers are locked.
0	RUNNING	0x0	R	**BURTC running status**
	Indicates the current status of BURTC running			

15.5.6 BURTC_IF - Interrupt Flag Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															COMP	OF

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	COMP	0x0	RW	**Compare Match Interrupt Flag**
	Set on a compare match between CNT and COMP.			
0	OF	0x0	RW	**Overflow Interrupt Flag**
	Set on a CNT value overflow.			

15.5.7 BURTC_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															COMP	OF

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	COMP	0x0	RW	**Compare Match Interrupt Flag**
	Set to enable the COMPIF Interrupt			
0	OF	0x0	RW	**Overflow Interrupt Flag**
	Set to enable the OFIF Interrupt			

15.5.8 BURTC_PRECNT - Pre-Counter Value Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																		0x0														
Access																		RW														
Name																		PRECNT														

Bit	Name	Reset	Access	Description
31:15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14:0	PRECNT	0x0	RW	**Pre-Counter Value**
	Gives access to the Pre-counter value of the BURTC.			

15.5.9 BURTC_CNT - Counter Value Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	CNT																															

Bit	Name	Reset	Access	Description
31:0	CNT	0x0	RW	**Counter Value**
	Gives access to the counter value of the BURTC.			

15.5.10 BURTC_EM4WUEN - EM4 Wakeup Request Enable Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															COMPEM4WUEN	OFEM4WUEN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	COMPEM4WUEN	0x0	RW	**Compare Match EM4 Wakeup Enable**
	Compare Match EM4 wakeup requests. No Synchronization done into peripheral clock domain.			
0	OFEM4WUEN	0x0	RW	**Overflow EM4 Wakeup Enable**
	Overflow EM4 Wakeup request. No Synchronization done into peripheral clock domain.			

15.5.11 BURTC_SYNCBUSY - Synchronization Busy Register

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												R	R	R	R	R
Name																												COMP	CNT	PRECNT	STOP	START

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	COMP	0x0	R	**Sync busy for COMP**
	Last writing of COMP is synchronizing to LF clock			
3	CNT	0x0	R	**Sync busy for CNT**
	Last writing of CNT is synchronizing to LF clock			
2	PRECNT	0x0	R	**Sync busy for PRECNT**
	Last writing of PRECNT is synchronizing to LF clock			
1	STOP	0x0	R	**Sync busy for STOP**
	Last writing of STOP is synchronizing to LF clock			
0	START	0x0	R	**Sync busy for START**
	Last writing of START is synchronizing to LF clock			

15.5.12 BURTC_LOCK - Configuration Lock Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0xAEE8								
Access																								W								
Name																								LOCKKEY								

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0xAEE8	W	**Configuration Lock Key**

Write any other value than the unlock code to lock BURTC_EN, BURTC_CFG, BURTC_CMD, BURTC_PRECNT, BURTC_CNT and BURTC_COMP registers from editing. Write the unlock code to unlock.

Value	Mode	Description
44776	UNLOCK	Write to unlock all BURTC lockable registers

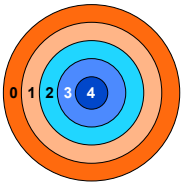
15.5.13 BURTC_COMP - Compare Value Register

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x0																
Access																RW																
Name																COMP																

Bit	Name	Reset	Access	Description
31:0	COMP	0x0	RW	**Compare Value**

A compare match event occurs when CNT is equal to this value. This event sets the COMP interrupt flag. It is also available as a PRS signal.

16. BURAM - Backup RAM



Quick Facts

What?

The BURAM is a dedicated 128-byte low-power RAM that is retained in EM4.

Why?

Most of the system, including the RAM, is powered off at EM4 entry to minimize current draw. The purpose of the BURAM is to retain critical data for use when the system wakes up.

How?

Because it is separate from the main system RAM, the BURAM has a dedicated power supply that is not shutdown when the system enters EM4.

16.1 Introduction

The Back-Up RAM (BURAM) is a dedicated 128-byte RAM that remains powered when the system enters EM4. Upon exit from EM4, the data retained in the BURAM can be accessed by the application software.

16.2 Functional Description

The BURAM consists of 32 x 32-bit registers, which are retained in all energy modes, including EM4. Each word in the BURAM is accessible through the corresponding 32 RETx_REG register. Note that each RETx_REG register has an undefined state out of reset.

16.3 BURAM Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	BURAM_RETx_REG	RW	Retention Register
0x1000	BURAM_RETx_REG_SET	RW	Retention Register
0x2000	BURAM_RETx_REG_CLR	RW	Retention Register
0x3000	BURAM_RETx_REG_TGL	RW	Retention Register

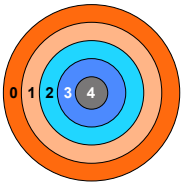
16.4 BURAM Register Description

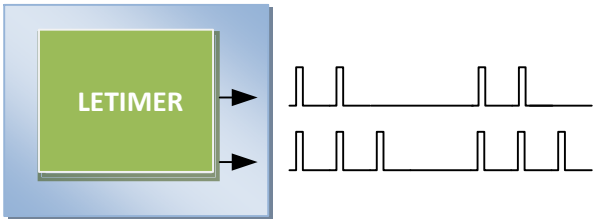
16.4.1 BURAM_RETx_REG - Retention Register

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	RETREG																															

Bit	Name	Reset	Access	Description
31:0	RETREG	0x0	RW	**Latch based Retention register** The RETREG registers are undefined out of reset. Any written RETREG values will be retained through any event other than a brownout or power-on reset.

17. LETIMER - Low Energy Timer





Quick Facts

What?

The LETIMER is a down-counter that can keep track of time and output configurable waveforms. Running on a 32768 Hz clock, the LETIMER is available in EM0 Active, EM1 Sleep, EM2 DeepSleep, and EM3 Stop.

Why?

The LETIMER can be used to provide repeatable waveforms to external components while remaining in EM2 DeepSleep. It is well suited for applications such as metering systems or to provide more compare values than available in the SYSRTC.

How?

With buffered repeat and top value registers, the LETIMER can provide glitch-free waveforms at frequencies up to 16 kHz. It can be coupled with SYSRTC using PRS, allowing advanced time-keeping and wake-up functions in EM2 DeepSleep and EM3 Stop

17.1 Introduction

The LETIMER is a down-counter that can keep track of time and output configurable waveforms with minimal software intervention. Running on a Low Frequency clock, the LETIMER is available in Energy Mode0, Energy Mode 1 and optionally available in Energy Mode 2 and Energy Mode 3. Because of this, it can be used for timing and output generation when most of the device is powered down, allowing simple tasks to be performed while the power consumption of the system is kept at an absolute minimum. It is well suited for applications such as metering systems or to provide more compare values than available in the SYSRTC. With buffered repeat and top value registers, the LETIMER can provide glitch-free waveforms at frequencies up to 16 kHz. It can be coupled with other peripherals using PRS, allowing advanced time-keeping and wake-up functions

17.2 Features

High-level features

- 24-bit Down counter
- 8-bit prescalar
- 2 Compare match registers
- TOP register can be Timer top value
- TOP register can be double buffered using TOPBUFF register
- Double buffered 8-bit Repeat Register
- Timer Start/Stop/Clear trigger can be from PRS or Software
- Configurable 2 Output pins - Toggle/Pulse/PWM
- Interrupt - Compare match/Timer underflow/Repeat done
- Optionally runs during debug
- 2 output pins can optionally be configured to provide different waveforms on timer underflow:
  - Toggle output pin
  - Pulse output with width of One Prescaled clock period
  - PWM
- 2 PRS Output

17.3 Functional Description

An overview of the LETIMER module is shown in Figure 17.1 LETIMER Overview on page 446. The LETIMER is a 24-bit down-counter with two compare registers, LETIMERn_COMP0 and LETIMERn_COMP1. The LETIMERn_TOP register can optionally act as a top value for the counter. The repeat counter LETIMERn_REP0 allows the timer to count a specified number of times before it stops. Both the LETIMERn_TOP and LETIMERn_REP0 registers can be double buffered by the LETIMERn_TOPBUFF and LETIMERn_REP1 registers to allow continuous operation. The timer can generate a single pin output, or two linked outputs.

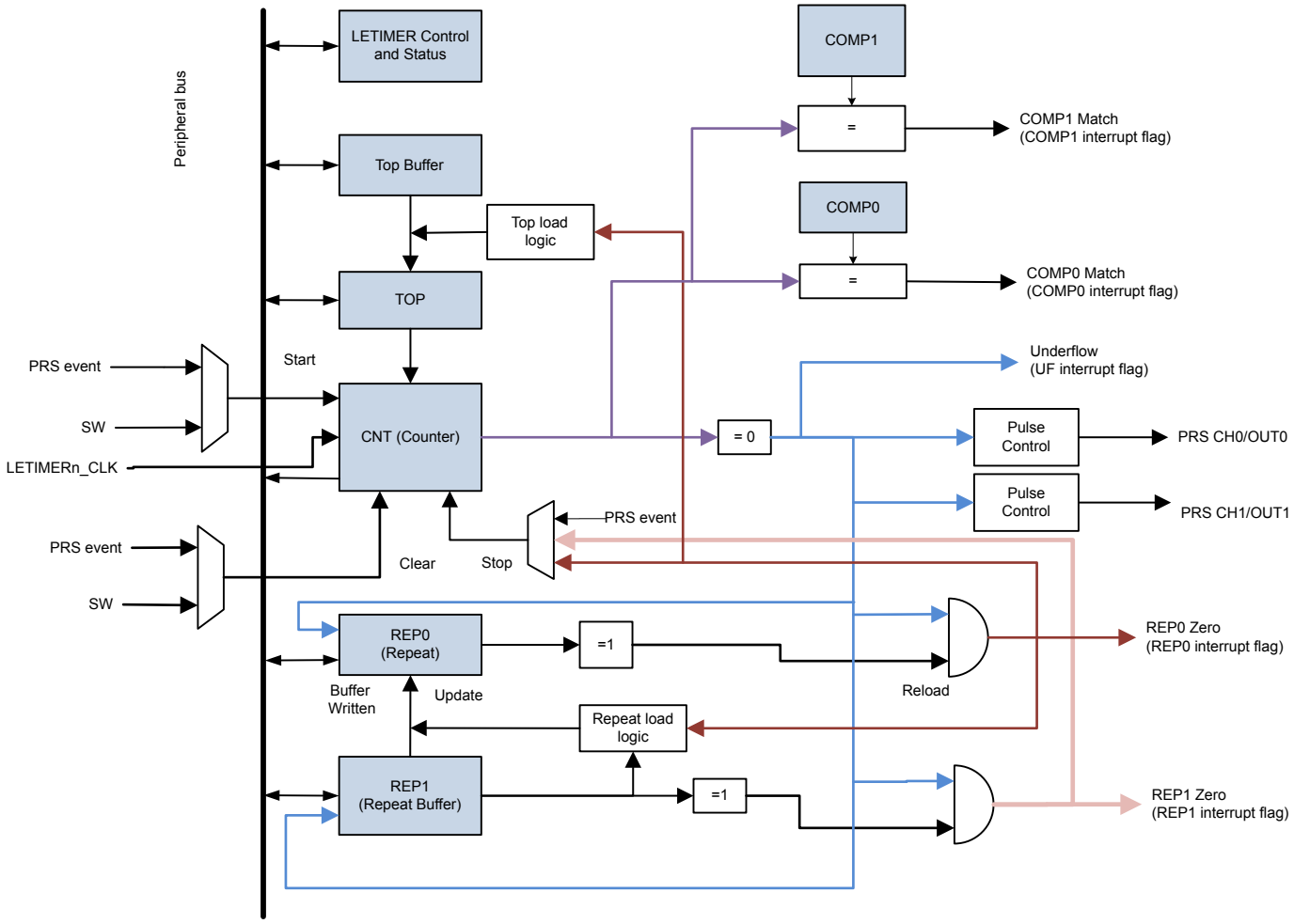


Figure 17.1. LETIMER Overview

17.3.1 Internal Overview

Timer

The timer value can be read using the LETIMERn_CNT register. The value can be written, and it can also be cleared by setting the CLEAR command bit in LETIMERn_CMD. If the CLEAR and START commands are issued at the same time, the timer will be cleared, then start counting at the top value.

Compare Registers

- The LETIMER has two compare match registers, LETIMERn_COMP0 and LETIMERn_COMP1. Each of these compare registers are capable of generating an interrupt when the counter value LETIMERn_CNT is equal to their value. When LETIMERn_CNT is equal to the value of LETIMERn_COMP0, the interrupt flag COMP0 in LETIMERn_IF is set, and when LETIMERn_CNT is equal to the value of LETIMERn_COMP1, the interrupt flag COMP1 in LETIMERn_IF is set.
- **Top Value**

  If CNTTOPEN in LETIMERn_CTRL is set, the value of LETIMERn_TOP acts as the top value of the timer, and LETIMERn_TOP is loaded into LETIMERn_CNT on timer underflow. If CNTTOPEN is cleared to 0, the timer wraps around to 0xFFFFFF. The underflow interrupt flag UF in LETIMERn_IF is set when the timer reaches zero.
- **Repeat Modes**

  By default, the timer wraps around to the top value or 0xFFFFFF on each underflow, and continues counting. The repeat counters can be used to get more control of the operation of the timer, including defining the number of times the counter should wrap around. Four different repeat modes are available, see Table 17.1 LETIMER Repeat Modes on page 447.

Table 17.1. LETIMER Repeat Modes

REPMODE	Mode	Description
0b00	Free-running	The timer runs until it is stopped.
0b01	One-shot	The timer runs as long as LETIMERn_REP0 != 0. LETIMERn_REP0 is decremented at each timer underflow.
0b10	Buffered	The timer runs as long as LETIMERn_REP0 != 0. LETIMERn_REP0 is decremented on each timer underflow. If LETIMERn_REP1 has been written with Non zero value, then it is loaded into LETIMERn_REP0 when LETIMERn_REP0 is about to be decremented to 0 and Timer countinue counting with new LETIMERn_REP0.
0b11	Double	The timer runs as long as LETIMERn_REP0 != 0 or LETIMERn_REP1 != 0. Both LETIMERn_REP0 and LETIMERn_REP1 are decremented at each timer underflow.

  The interrupt flags REP0 and REP1 in LETIMERn_IF are set whenever LETIMERn_REP0 or LETIMERn_REP1 are decremented to 0 respectively. REP0 is also set when the value of LETIMERn_REP1 is loaded into LETIMERn_REP0 in buffered mode.

  Write operations to LETIMERn_REP0 have priority over buffer loads from LETIMERn_REP1.
- **Buffered Top Value**

  In Buffered Mode, If BUFTOP in LETIMERn_CTRL is set, the value of LETIMERn_TOP is buffered by LETIMERn_TOPBUFF. In this mode, the value of LETIMERn_TOPBUFF is loaded into LETIMERn_TOP every time LETIMERn_REP0 is about to decrement to 0. This can be used to generate continually changing output waveforms.

  Write operations to LETIMERn_TOP have priority over buffer loads from LETIMERn__TOPBUFF.

17.3.2 Free Running Mode

In free-running mode, the LETIMER acts as a regular timer and the repeat operation is disabled. When started, the timer runs until it is stopped using the STOP command bit in LETIMERn_CMD/PRS. A state machine for this mode is shown in Figure 17.2 LETIMER State Machine for Free-running Mode on page 448 .

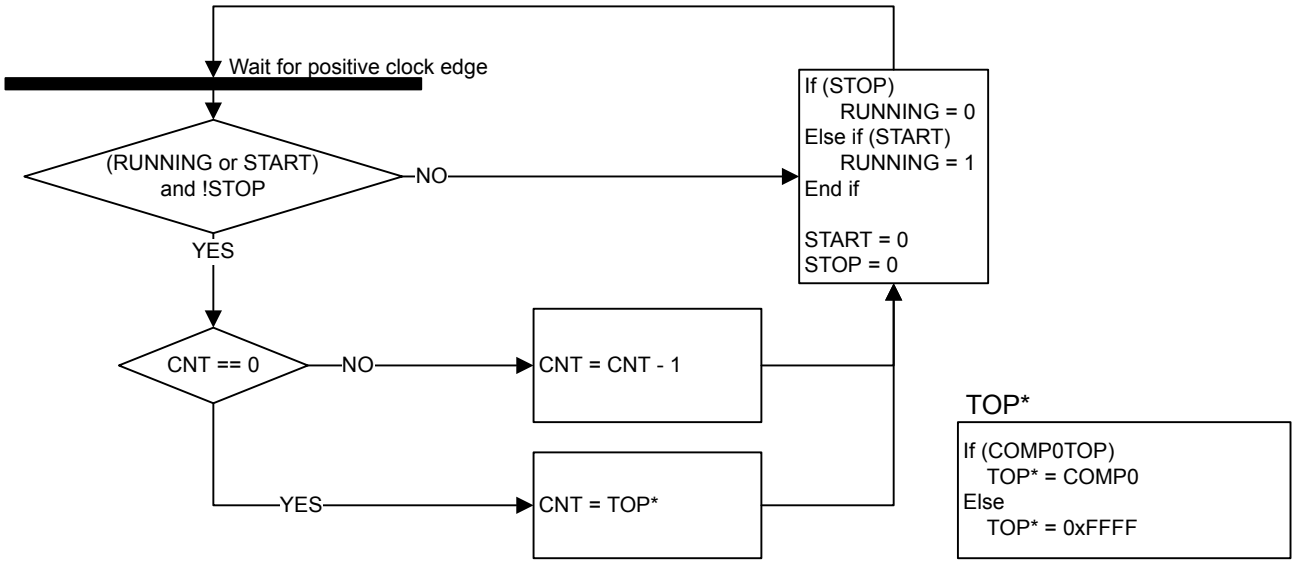


Figure 17.2. LETIMER State Machine for Free-running Mode

Note that the CLEAR command bit in LETIMERn_CMD always has priority over Decrement and Load TOP to LETIMERn_CNT. When the clear command is used, LETIMERn_CNT is set to 0 and an underflow event will not be generated when LETIMERn_CNT wraps around to the top value or 0xFFFFFF. Since no underflow event is generated, no output action is performed. LETIMERn_REP0, LETIMERn_REP1, LETIMERn_COMP0 and LETIMERn_COMP1 are also left untouched.

17.3.3 One-shot Mode

The one-shot repeat mode is the most basic repeat mode. In this mode, the repeat register LETIMERn_REP0 is decremented every time the timer underflows, and the timer stops when LETIMERn_REP0 goes from 1 to 0. In this mode, the timer counts down LETIMERn_REP0 times, i.e. the timer underflows LETIMERn_REP0 times.

Note: Note that write operations to LETIMERn_REP0 have priority over the timer decrement event. If LETIMERn_REP0 is assigned a new value in the same cycle as a timer decrement event occurs, the timer decrement will not occur and the new value is assigned.

LETIMERn_REP0 can be written while the timer is running to allow the timer to run for longer periods at a time without stopping. Write to LETIMERn_REP0 should be done after checking SYNC busy statusFigure 17.3 LETIMER One-shot Repeat State Machine on page 449 .

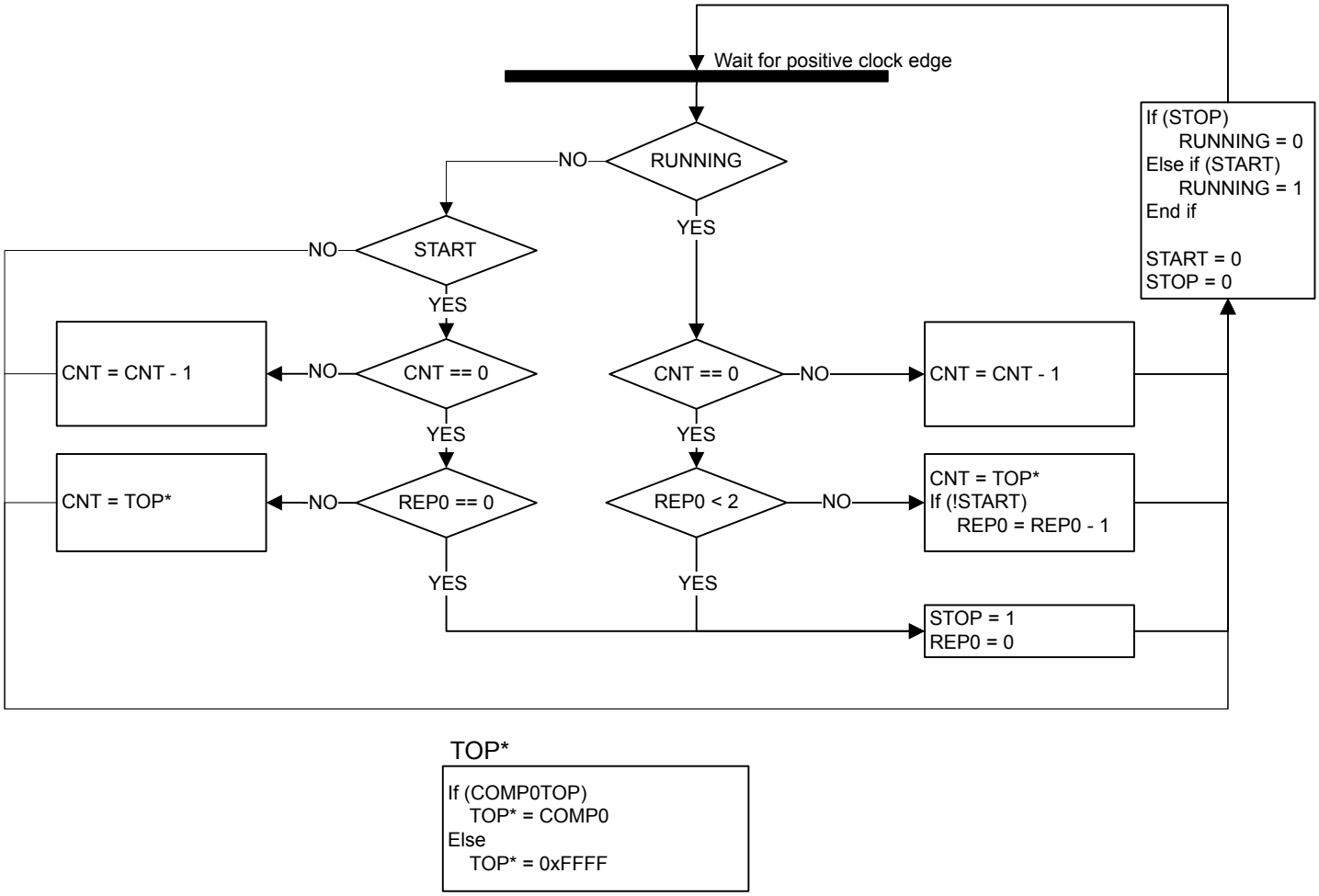


Figure 17.3. LETIMER One-shot Repeat State Machine

17.3.4 Buffered Mode

The Buffered repeat mode allows buffered timer operation. When started, the timer runs LETIMERn_REP0 number of times. If LETIMERn_REP1 has been written since the last time it was used and if it is nonzero, LETIMERn_REP1 is then loaded into LETIMERn_REP0, and counting continues the new number of times. The timer keeps going as long as LETIMERn_REP1 is updated with a nonzero value before LETIMERn_REP0 is finished counting down. The timer top value (LETIMERn_TOP) may also optionally be buffered using Top buff value (LETIMERn_TOPBUFF) by setting BUFTOP in LETIMERn_CTRL.

If the timer is started when both LETIMERn_CNT and LETIMERn_REP0 are zero but LETIMERn_REP1 is non-zero, LETIMERn_REP1 is loaded into LETIMERn_REP0, and the counter counts the loaded number of times.

Used in conjunction with a buffered top value, both the top and repeat values of the timer may be buffered, and the timer can for instance be set to run 4 times with period 7 (top value 6), 6 times with period 200, then 3 times with period 50.

A state machine for the buffered repeat mode is shown in Figure 17.4 LETIMER Buffered Repeat State Machine on page 450. $REP1_{USED}$ shown in the state machine is an internal variable that keeps track of whether the value in LETIMERn_REP1 has been loaded into LETIMERn_REP0 or not. The purpose of this is that a value written to LETIMERn_REP1 should only be counted once. $REP1_{USED}$ is cleared whenever LETIMERn_REP1 is used.

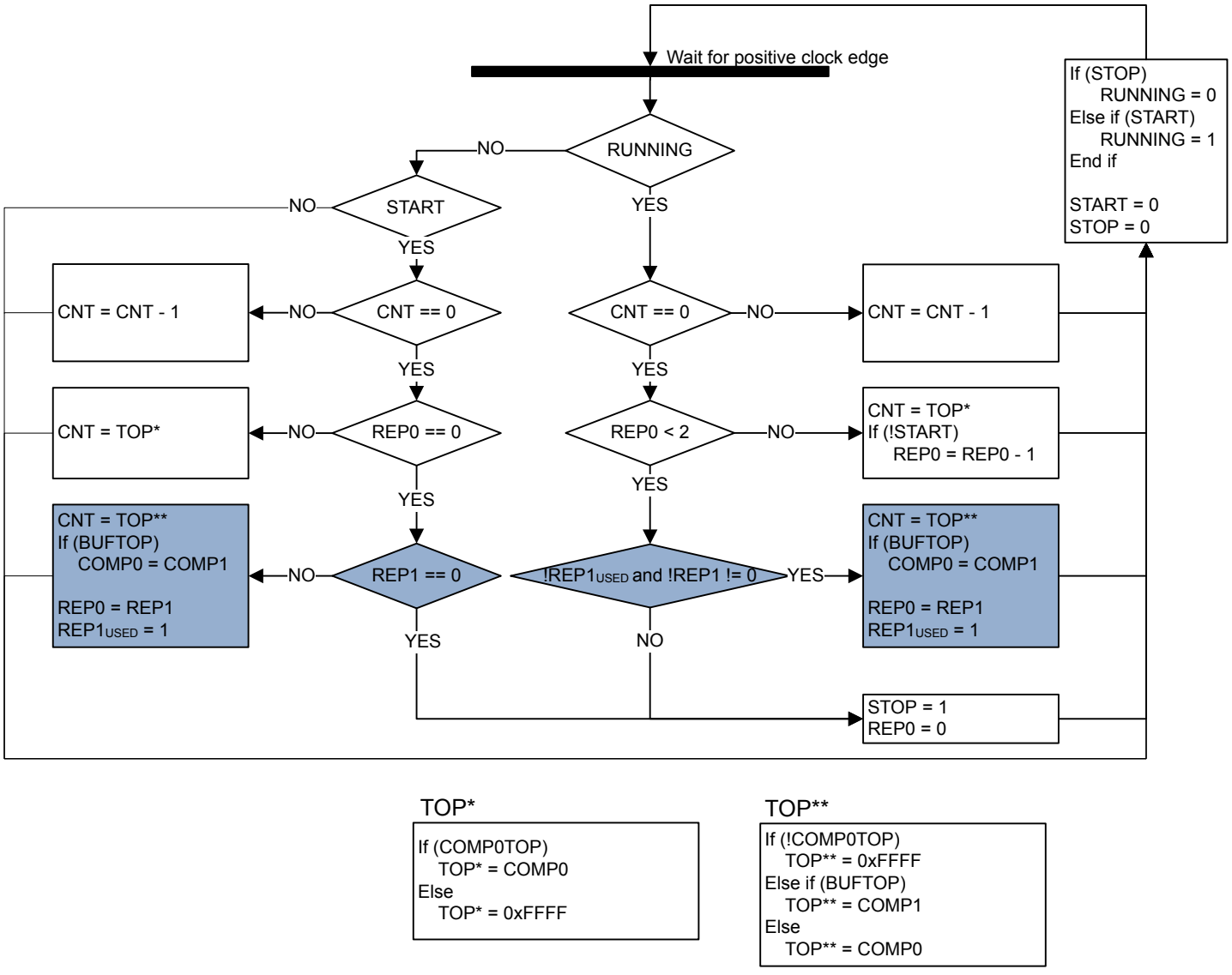


Figure 17.4. LETIMER Buffered Repeat State Machine

17.3.5 Double Mode

The Double repeat mode works much like the one-shot repeat mode. The difference is that, where the one-shot mode counts as long as LETIMERn_REP0 is larger than 0, the double mode counts as long as either LETIMERn_REP0 or LETIMERn_REP1 is larger than 0. As an example, say LETIMERn_REP0 is 3 and LETIMERn_REP1 is 10 when the timer is started. If no further interaction is done with the timer, LETIMERn_REP0 will now be decremented 3 times, and LETIMERn_REP1 will be decremented 10 times. The timer counts a total of 10 times, and LETIMERn_REP0 is 0 after the first three timer underflows and stays at 0. LETIMERn_REP0 and LETIMERn_REP1 can be written at any time. After a write to either of these, the timer is guaranteed to underflow at least the written number of times if the timer is running. Use the Double repeat mode to generate output on both the LETIMER outputs at the same time. The state machine for this repeat mode can be seen in Figure 17.5 LETIMER Double Repeat State Machine on page 451.

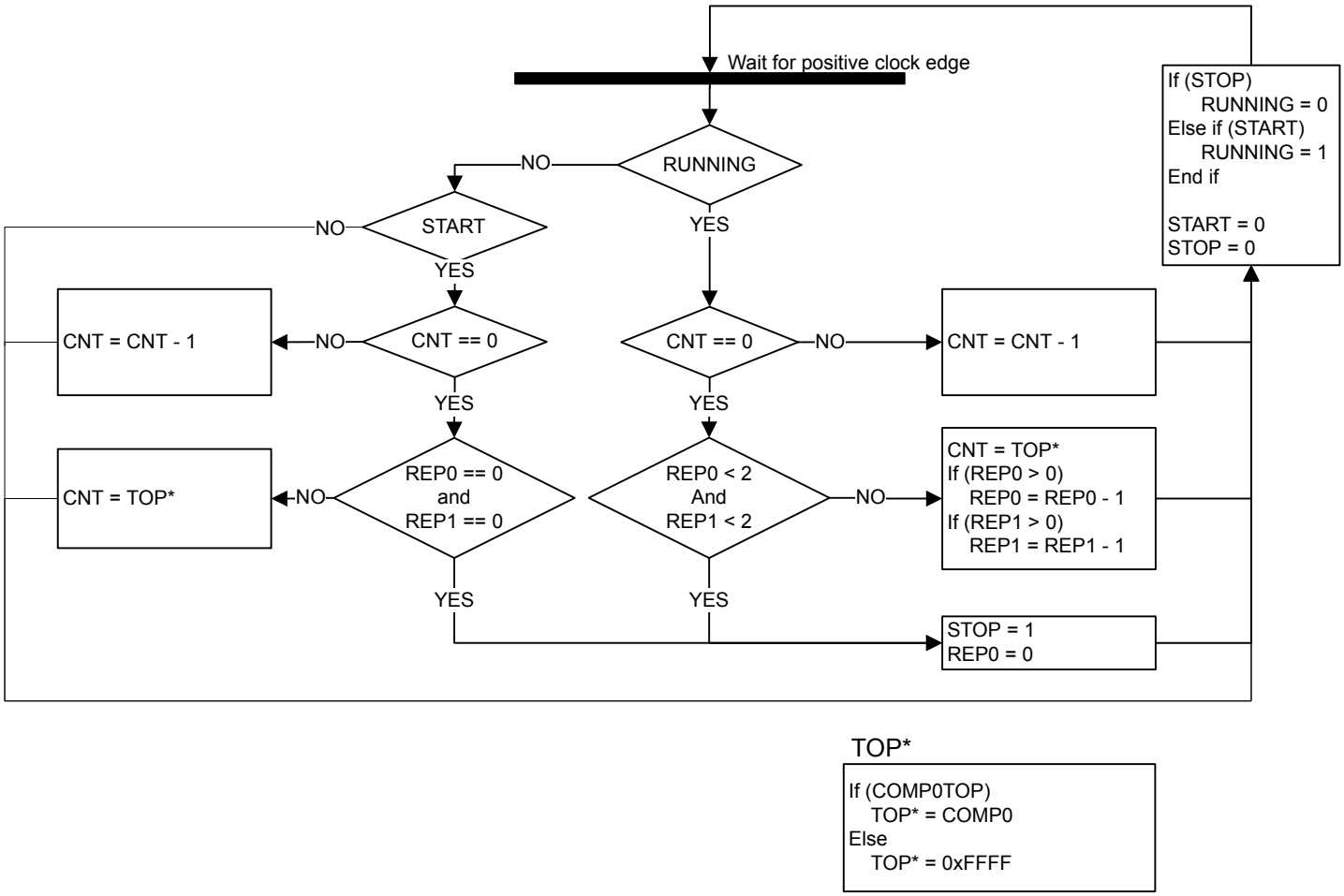


Figure 17.5. LETIMER Double Repeat State Machine

17.4 Clock Frequency

The LETIMER clock source is derived from EM23GRPACLK, which is selected in the Clock Management Unit (CMU), and is typically configured to have a frequency of 32 kHz in EM0/1/2 and 1 kHz in EM3. The LETIMER clock prescaler is defined by LETIMERn_CTRL->CNTPRESC.

The LETIMER Prescaled clock frequency is given by Figure 17.6 LETIMER Clock Frequency on page 452.

EM0/1/2 - Clocked by LFRCO

$$f_{LETIMERn\_CLK} = 32768/2^{CNTPRESC}$$

EM3 - Clocked by ULFRCO

$$f_{LETIMERn\_CLK} = 1024/2^{CNTPRESC}$$

Figure 17.6. LETIMER Clock Frequency

The exponent CNTPRESC is a 4 bit value in the LETIMERn_CTRL->CNTPRESC register bits.

To use this module, the LETIMERn_CLK must be enabled by writing 1 to LETIMERn_EN->EN.

17.5 PRS Input Triggers

The LETIMER can be configured to start, stop, and/or clear based on PRS inputs. The diagram showing the functions of the PRS input triggers is shown in Figure 17.7 LETIMER PRS input triggers. on page 453.

There are 3 PRS inputs to the LETIMER, allowing the LETIMER to be started, stopped, or cleared based on the PRS inputs. The PRSSTARTMODE, PRSSTOPMODE, and PRSCLEARMODE bitfields in LETIMERn->PRSMODE select which edge or edge(s) will trigger the start, stop, and/or clear action.

The PRS channel inputs can be configured in the PRS_CONSUMER_LETIMERn_CLEAR, PRS_CONSUMER_LETIMERn_START, and PRS_CONSUMER_LETIMERn_STOP registers in the PRS module.

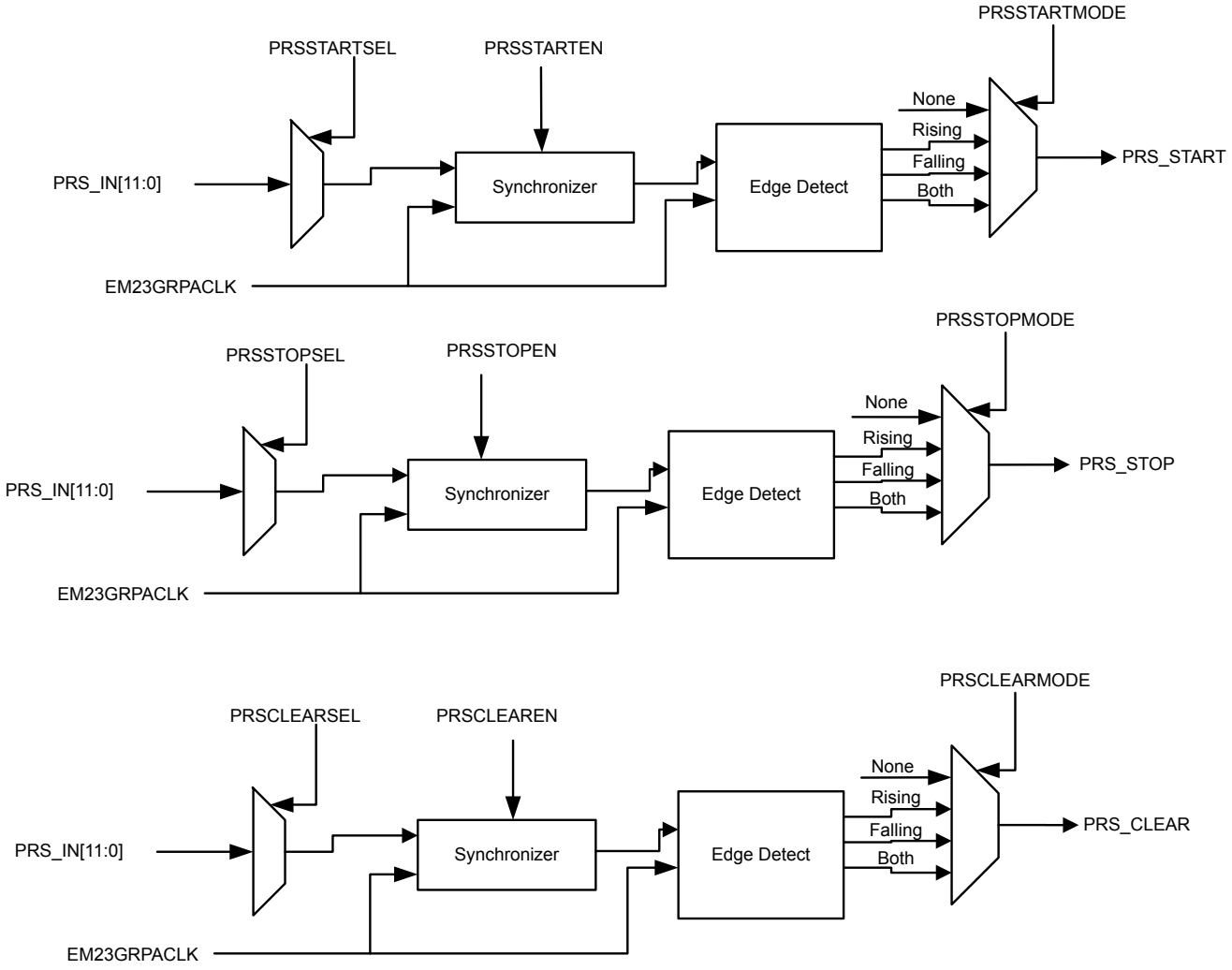


Figure 17.7. LETIMER PRS input triggers.

17.6 Debug

If DEBUGRUN in LETIMERn_CTRL is cleared, the LETIMER automatically stops counting when the CPU is halted during a debug session, and resumes operation when the CPU continues. Because of synchronization, the LETIMER is halted two clock cycles after the CPU is halted, and continues running two clock cycles after the CPU continues. RUNNING in LETIMERn_STATUS is not cleared when the LETIMER stops because of a debug-session.

Set DEBUGRUN in LETIMERn_CTRL to allow the LETIMER to continue counting even when the CPU is halted in debug mode.

17.7 Output Action

For each of the Outputs, an output action can be set.

The output actions can be set by configuring UFOA0 and UFOA1 in LETIMERn_CTRL. UFOA0 defines the action on output 0, while UFOA1 defines the action on output 1. The possible actions are defined in Table 17.2 LETIMER Underflow Output Actions on page 454.

Table 17.2. LETIMER Underflow Output Actions

UF0A0/UF0A1	Mode	Description
0b00	Idle	The output is held at its idle value
0b01	Toggle	The output is toggled on LETIMERn_CNT underflow
0b10	Pulse	The output is held active for one LF clock cycle on LETIMERn_CNT underflow It then returns to its idle value
0b11	PWM	The output is set idle on LETIMERn_CNT underflow and active on compare match with LETIMERn_COMP0/1.

Note: For the Pulse output Disabling LETIMER, Clearing Output while pulse output is generated can affect the pulse width.

Note: For Double mode, OUT0/1 generation is enabled when LETIMERn_REP0/1 != 0 respectively.

The polarity of the outputs can be set individually by configuring OPOL0 and OPOL1 in LETIMERn_CTRL. When these are cleared, their respective outputs have a low idle value and a high active value. When they are set, the idle value is high, and the active value is low. It is recommended to Clear outputs after changing polarity to makesure outputs take their default value.

When using the toggle action, the outputs can be driven to their idle values by setting their respective CTO0/CTO1 command bits in LETIMERn_CTRL. This can be used to put the output in a well-defined state before beginning to generate toggle output, which may be important in some applications. The command bit can also be used while the timer is running.

17.8 PRS Output

The LETIMER outputs can be routed out onto the PRS system. LETn_O0 can be routed to PRS channel 0, and LETn_O1 can be routed to PRS channel 1. Enabling the PRS connection can be done by setting SOURCESEL to LETIMERx and SIGSEL to LETIMERxCHn in PRS_CHx_CTRL.

17.9 Interrupts

The interrupts generated by the LETIMER are combined into one interrupt vector. If the interrupt for the LETIMER is enabled, an interrupt will be made if one or more of the interrupt flags in LETIMERn_IF and their corresponding bits in LETIMER_IEN are set.

17.10 Using the LETIMER in EM3

The LETIMER can be enabled all the way down to EM3 by using the ULFRCO as clock source. This is done by setting CMU_EM23GRPACLKCTRL.CLKSEL to ULFRCO before enabling the LETIMER block.

17.11 Register Access

This module is a Low Energy Peripheral, and supports immediate synchronization. For description regarding immediate synchronization, refer to 4.2.4.4 Peripheral Access Performance.

Since this module is a Low Energy Peripheral, and runs off a clock which is asynchronous to the APB register clock, special considerations must be taken when accessing registers.

17.12 Programmer's Model

Important Note : Before writing any LFSYNC register, the module must be enabled (LETIMER_EN->EN) and the LETIMER_SYNCBUSY register should be polled to ensure the SYNC busy of that particular register field is not high.

Write LETIMER Configuration into LETIMER_CTRL Register

Enable clock to LETIMER module by setting LETIMER_EN->EN = 1

If used, write compare values into LETIMER_COMP0 and LETIMER_COMP1

If used, write repeat values into LETIMER_REP0 and LETIMER_REP1

If used, write LETIMER_TOP and LETIMER_TOPBUFF

If PRS is used as a trigger, configure LETIMER_PRSMODE accordingly

Enable Interrupts in LETIMER_IEN

Write LETIMER_CMD register to START Timer

17.12.1 Free Running Mode

LETIMER operation in Free running Mode with different output modes are shown in Figure 17.8 LETIMER - Free Running Mode Waveform on page 455. In this example, REPMODE in LETIMERn_CTRL is set to FREE, CNTTOPEN also in LETIMERn_CTRL has been set and LETIMERn_TOP has been written to 3. As seen in the figure, LETIMERn_TOP now decides the length of the signal periods. For the toggle mode, the period of the output signal is 2(LETIMERn_TOP + 1), and for the pulse modes, the periods of the output signals are LETIMERn_TOP+1. Note that the pulse outputs are delayed by one period relative to the toggle output. The pulses come at the end of their periods.

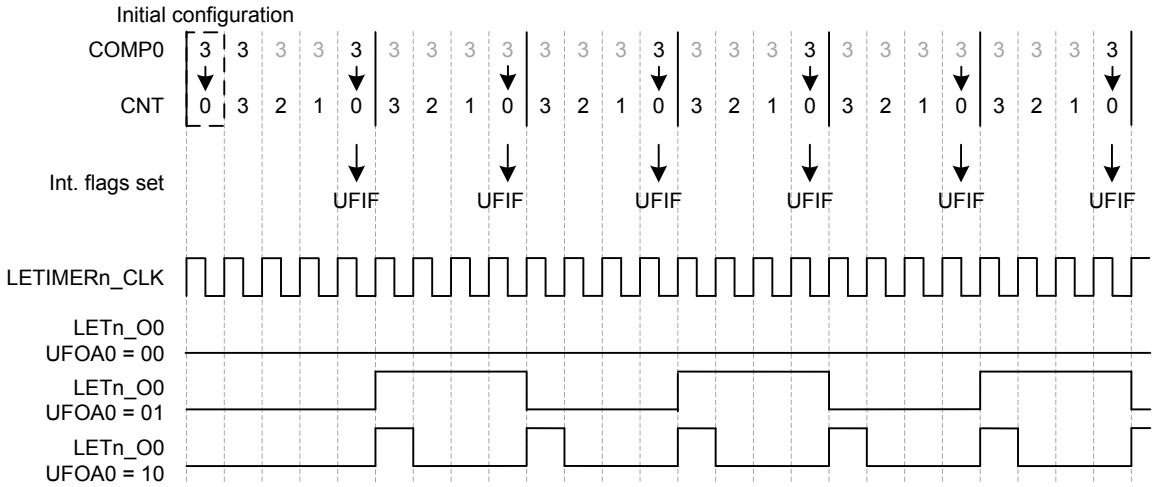


Figure 17.8. LETIMER - Free Running Mode Waveform

17.12.2 One Shot Mode

LETIMER operation in ONESHOT Mode with different output modes are shown in Figure 17.9 LETIMER - One Shot Mode Waveform on page 456. In this example, REPMODE in LETIMERn_CTRL is set to ONESHOT, CNTTOPEN also in LETIMERn_CTRL has been set and LETIMERn_TOP has been written to 3 and LETIMERn_REP0 has been written to 3. The resulting behavior is pretty similar to that shown in Figure 6, but in this case, the timer stops after counting to zero LETIMERn_REP0 times. By using LETIMERn_REP0 the user has full control of the number of pulses/toggles generated on the output.

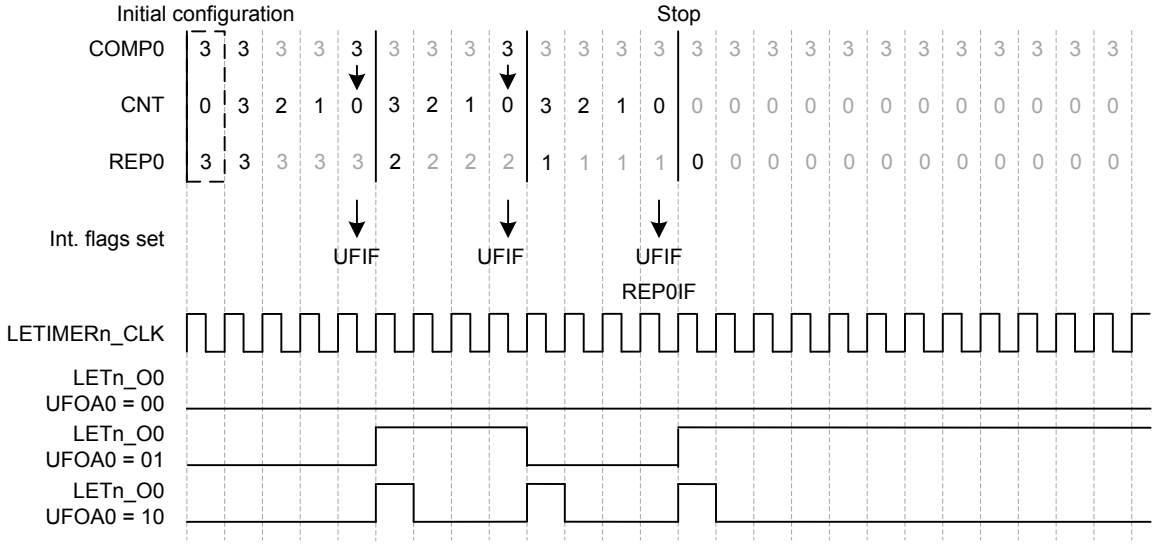


Figure 17.9. LETIMER - One Shot Mode Waveform

17.12.3 DOUBLE Mode

LETIMER operation in DOUBLE Mode with both outputs is shown in Figure 17.10 LETIMER - Double Mode Waveform on page 456. UFOA0 and UFOA1 in LETIMERn_CTRL are configured for pulse output and the outputs are configured for low idle polarity. As seen in the figure, the number written to the repeat registers determine the number of pulses generated on each of the outputs.

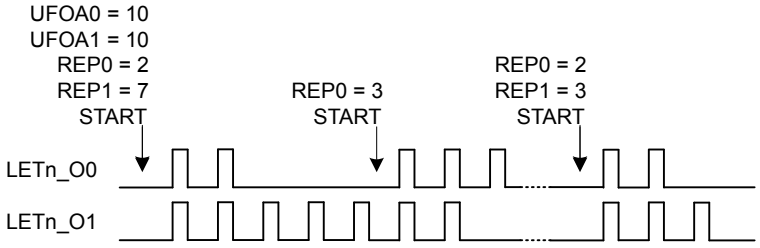


Figure 17.10. LETIMER - Double Mode Waveform

17.12.4 BUFFERED Mode

In BUFFERED Mode LETIMERn_TOPBUFF and LETIMERn_REP1 registers are used as Buffers for LETIMERn_TOP and LETIMERn_REP0 respectiverly. If both LETIMERn_TOP and LETIMERn_REP0 are 0 in buffered mode, and CNTTOPEN and BUFTOP in LETIMERn_CTRL are set, the values of LETIMERn_TOPBUFF and LETIMERn_REP1 are loaded into LETIMERn_TOP and LETIMERn_REP0 respectively when the timer is started. If no additional writes to LETIMERn_REP1 are done before the timer stops, LETIMERn_REP1 determines the number of pulses/toggles generated on the output, and LETIMERn_TOPBUFF determines the period lengths.

As the SYSRTC can also be used via PRS to start the LETIMER, the SYSRTC and LETIMER can thus be combined to generate specific pulse-trains at given intervals. Software can update LETIMERn_TOPBUFF and LETIMERn_REP1 to change the number of pulses and pulse-period in each train, but if changes are not required, software does not have to update the registers between each pulse train.

For the example in Figure 17.11 LETIMER - Buffered Mode Waveform on page 457, the initial values cause the LETIMER to generate two pulses with 3 cycle periods, or a single pulse 3 cycles wide every time the LETIMER is started. After the output has been generated, the LETIMER stops, and is ready to be triggered again.

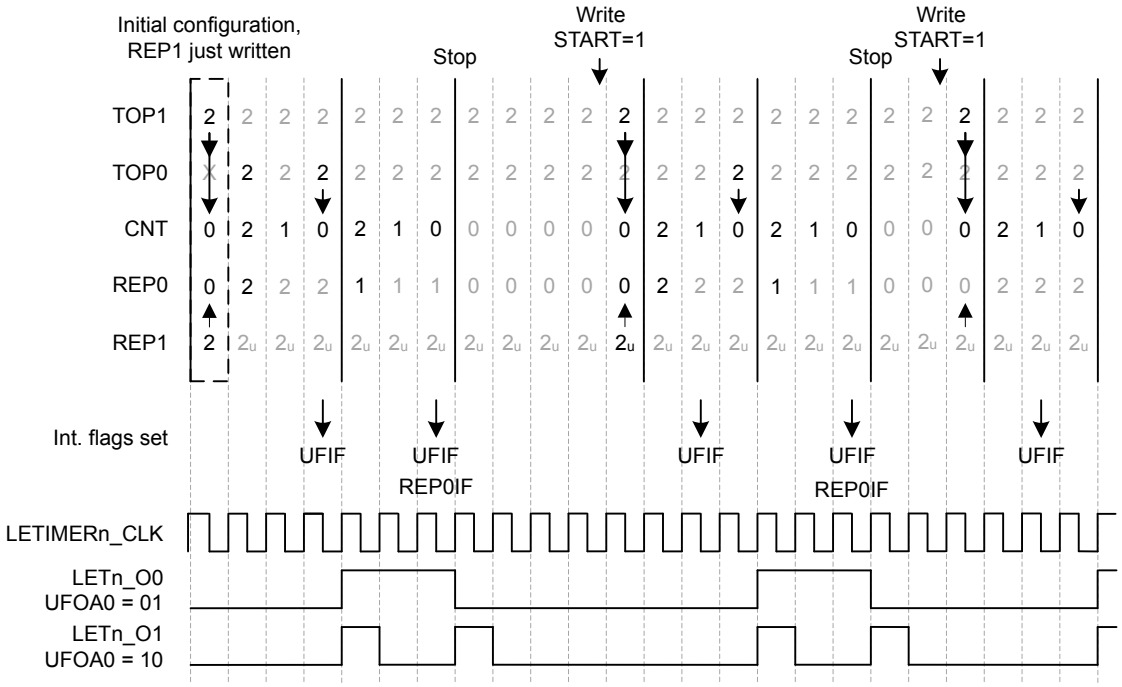


Figure 17.11. LETIMER - Buffered Mode Waveform

17.12.5 Continuous Output Generation

In some scenarios, it might be desired to make LETIMER generate a continuous waveform. Very simple constant waveforms can be generated without the repeat counter as shown in Figure 17.8 LETIMER - Free Running Mode Waveform on page 455, but to generate changing waveforms, using the repeat counter and buffer registers can prove advantageous.

For the example in Figure 17.12 LETIMER - Continuous Operation on page 458, the goal is to produce a pulse train consisting of 3 sequences with the following properties:

- 3 pulses with periods of 3 cycles
- 4 pulses with periods of 2 cycles
- 2 pulses with periods of 3 cycles

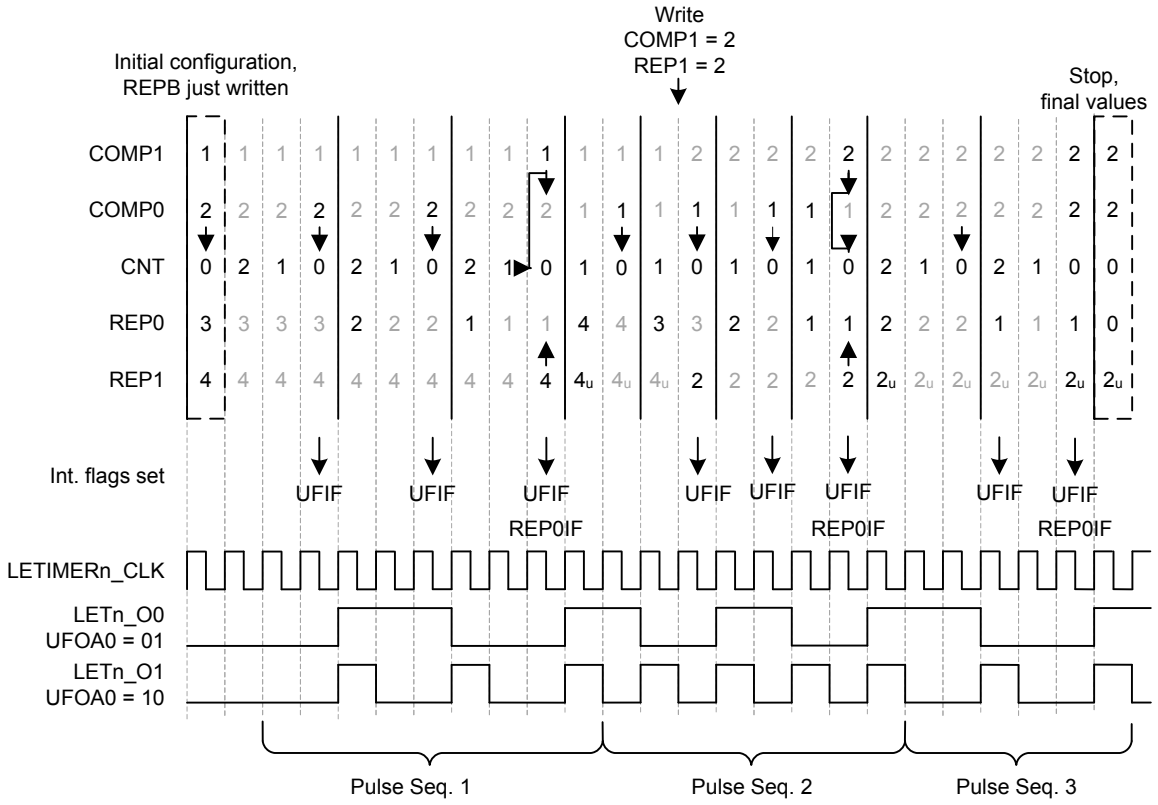


Figure 17.12. LETIMER - Continuous Operation

The first two sequences are loaded into the LETIMER before the timer is started.

LETIMERn_TOP is set to 2 (cycles – 1), and LETIMERn_REP0 is set to 3 for the first sequence, and the second sequence is loaded into the buffer registers, i.e. TOPBUFF is set to 1 and LETIMERn_REP1 is set to 4.

The LETIMER is set to trigger an interrupt when LETIMERn_REP0 is done by setting REP0 in LETIMERn_IEN. This interrupt is a good place to update the values of the buffers. Last but not least REPMODE in LETIMERn_CTRL is set to buffered mode, and the timer is started.

In the interrupt routine the buffers are updated with the values for the third sequence. If this had not been done, the timer would have stopped after the second sequence.

The final result is shown in Figure 17.12 LETIMER - Continuous Operation on page 458. The pulse output is grouped to show which sequence generated which output. Toggle output is also shown in the figure. Note that the toggle output is not aligned with the pulse outputs.

Note: Multiple LETIMER cycles are required to write a value to the LETIMER registers. The example in Figure 17.12 LETIMER - Continuous Operation on page 458 assumes that writes are done in advance so they arrive in the LETIMER as described in the figure.

Figure 17.13 LETIMERn_CNT Not Initialized to 0 on page 459 shows an example where the LETIMER is started while LETIMERn_CNT is nonzero. In this case the length of the first repetition is given by the value in LETIMERn_CNT.

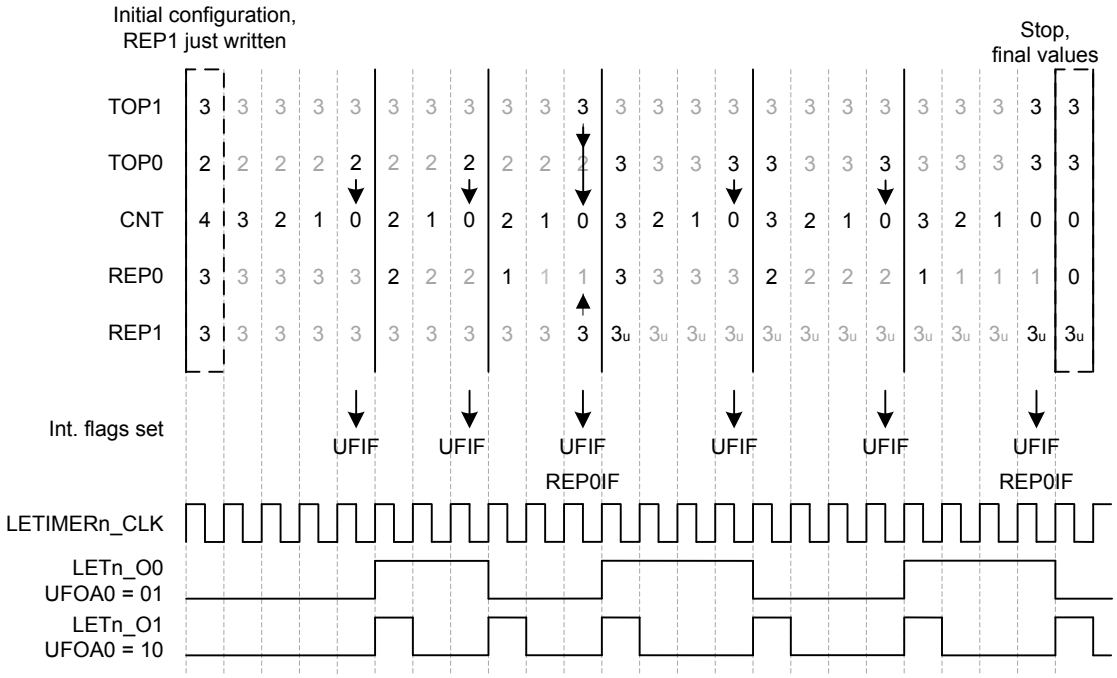


Figure 17.13. LETIMERn_CNT Not Initialized to 0

17.12.6 PWM Output

There are several ways of generating PWM output with the LETIMER, but the most straight-forward way is to use the PWM output mode. This mode is enabled by setting UFOA0 or UFOA1 in LETIMERn_CTRL to 3. In PWM mode, the output is set to idle on timer underflow, and active on LETIMERn_COMP0/1 match, so if for instance CNTTOPEN = 1 and OPOL0 = 0 in LETIMERn_CTRL, LETIMERn_TOP determines the PWM period, and LETIMERn_COMP0/1 determines the active period.

The PWM period in PWM mode is LETIMERn_TOP + 1. There is no special handling of the case where LETIMERn_COMP0/1 > LETIMERn_TOP, so if LETIMERn_COMP0/1 > LETIMERn_TOP, the PWM output is given by the idle output value. This means that for OPOLx = 0 in LETIMERn_CTRL, the PWM output will always be 0 for at least one clock cycle, and for OPOLx = 1 LETIMERn_CTRL, the PWM output will always be 1 for at least one clock cycle.

To generate a PWM signal using the full PWM range, invert OPOLx when LETIMERn_COMP0/1 is set to a value larger than LETIMERn_TOP.

17.13 LETIMER Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	LETIMER_IPVERSION	R	IP Version
0x004	LETIMER_EN	RW ENABLE	Module En
0x008	LETIMER_SWRST	RW SWRST	Software Reset Register
0x00C	LETIMER_CTRL	RW	Control Register
0x010	LETIMER_CMD	W LFSYNC	Command Register
0x014	LETIMER_STATUS	RH	Status Register
0x018	LETIMER_CNT	RWH LFSYNC	Counter Value Register
0x01C	LETIMER_COMP0	RW	Compare Value Register 0
0x020	LETIMER_COMP1	RW	Compare Value Register 1
0x024	LETIMER_TOP	RWH LFSYNC	Counter TOP Value Register
0x028	LETIMER_TOPBUFF	RW	Buffered Counter TOP Value
0x02C	LETIMER_REP0	RWH LFSYNC	Repeat Counter Register 0
0x030	LETIMER_REP1	RWH LFSYNC	Repeat Counter Register 1
0x034	LETIMER_IF	RWH INTFLAG	Interrupt Flag Register
0x038	LETIMER_IEN	RW	Interrupt Enable Register
0x03C	LETIMER_LOCK	W	Configuration Lock Register
0x040	LETIMER_SYNCBUSY	RH	Synchronization Busy Register
0x050	LETIMER_PRSMODE	RW	PRS Input Mode Select Register
0x1000	LETIMER_IPVERSION_SET	R	IP Version
0x1004	LETIMER_EN_SET	RW ENABLE	Module En
0x1008	LETIMER_SWRST_SET	RW SWRST	Software Reset Register
0x100C	LETIMER_CTRL_SET	RW	Control Register
0x1010	LETIMER_CMD_SET	W LFSYNC	Command Register
0x1014	LETIMER_STATUS_SET	RH	Status Register
0x1018	LETIMER_CNT_SET	RWH LFSYNC	Counter Value Register
0x101C	LETIMER_COMP0_SET	RW	Compare Value Register 0
0x1020	LETIMER_COMP1_SET	RW	Compare Value Register 1
0x1024	LETIMER_TOP_SET	RWH LFSYNC	Counter TOP Value Register
0x1028	LETIMER_TOPBUFF_SET	RW	Buffered Counter TOP Value
0x102C	LETIMER_REP0_SET	RWH LFSYNC	Repeat Counter Register 0
0x1030	LETIMER_REP1_SET	RWH LFSYNC	Repeat Counter Register 1
0x1034	LETIMER_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1038	LETIMER_IEN_SET	RW	Interrupt Enable Register
0x103C	LETIMER_LOCK_SET	W	Configuration Lock Register
0x1040	LETIMER_SYNCBUSY_SET	RH	Synchronization Busy Register

Offset	Name	Type	Description
0x1050	LETIMER_PRSMODE_SET	RW	PRS Input Mode Select Register
0x2000	LETIMER_IPVERSION_CLR	R	IP Version
0x2004	LETIMER_EN_CLR	RW ENABLE	Module En
0x2008	LETIMER_SWRST_CLR	RW SWRST	Software Reset Register
0x200C	LETIMER_CTRL_CLR	RW	Control Register
0x2010	LETIMER_CMD_CLR	W LFSYNC	Command Register
0x2014	LETIMER_STATUS_CLR	RH	Status Register
0x2018	LETIMER_CNT_CLR	RWH LFSYNC	Counter Value Register
0x201C	LETIMER_COMP0_CLR	RW	Compare Value Register 0
0x2020	LETIMER_COMP1_CLR	RW	Compare Value Register 1
0x2024	LETIMER_TOP_CLR	RWH LFSYNC	Counter TOP Value Register
0x2028	LETIMER_TOPBUFF_CLR	RW	Buffered Counter TOP Value
0x202C	LETIMER_REP0_CLR	RWH LFSYNC	Repeat Counter Register 0
0x2030	LETIMER_REP1_CLR	RWH LFSYNC	Repeat Counter Register 1
0x2034	LETIMER_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2038	LETIMER_IEN_CLR	RW	Interrupt Enable Register
0x203C	LETIMER_LOCK_CLR	W	Configuration Lock Register
0x2040	LETIMER_SYNCBUSY_CLR	RH	Synchronization Busy Register
0x2050	LETIMER_PRSMODE_CLR	RW	PRS Input Mode Select Register
0x3000	LETIMER_IPVERSION_TGL	R	IP Version
0x3004	LETIMER_EN_TGL	RW ENABLE	Module En
0x3008	LETIMER_SWRST_TGL	RW SWRST	Software Reset Register
0x300C	LETIMER_CTRL_TGL	RW	Control Register
0x3010	LETIMER_CMD_TGL	W LFSYNC	Command Register
0x3014	LETIMER_STATUS_TGL	RH	Status Register
0x3018	LETIMER_CNT_TGL	RWH LFSYNC	Counter Value Register
0x301C	LETIMER_COMP0_TGL	RW	Compare Value Register 0
0x3020	LETIMER_COMP1_TGL	RW	Compare Value Register 1
0x3024	LETIMER_TOP_TGL	RWH LFSYNC	Counter TOP Value Register
0x3028	LETIMER_TOPBUFF_TGL	RW	Buffered Counter TOP Value
0x302C	LETIMER_REP0_TGL	RWH LFSYNC	Repeat Counter Register 0
0x3030	LETIMER_REP1_TGL	RWH LFSYNC	Repeat Counter Register 1
0x3034	LETIMER_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3038	LETIMER_IEN_TGL	RW	Interrupt Enable Register
0x303C	LETIMER_LOCK_TGL	W	Configuration Lock Register
0x3040	LETIMER_SYNCBUSY_TGL	RH	Synchronization Busy Register
0x3050	LETIMER_PRSMODE_TGL	RW	PRS Input Mode Select Register

17.14 LETIMER Register Description

17.14.1 LETIMER_IPVERSION - IP Version

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP Version**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

17.14.2 LETIMER_EN - Module En

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status**
	When EN is cleared, DISABLING is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs, FIFOs etc.			
0	EN	0x0	RW	**module en**
	Enable the LETIMER module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit. When EN is cleared(disablement), it halts module operation immediately, and initialize the core domain such that when the is re-enabled, it starts cleanly.			

17.14.3 LETIMER_SWRST - Software Reset Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	W
Name																															RESETTING	SWRST

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESETTING	0x0	R	**Software reset busy status** When SWRST command is issued, resetting logic sets RESETTING status immediately, and later it is cleared when reset process finishes.
0	SWRST	0x0	W	**Software reset command** A software reset command field resets the module back to the initial condition, similar to a power on reset condition

17.14.4 LETIMER_CTRL - Control Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0							0x0			0x0	0x0	0x0	0x0	0x0		0x0		0x0	
Access													RW							RW			RW	RW	RW	RW	RW		RW		RW	
Name													CNTPRESC							DEBUGRUN			CNTTOPEN	BUFTOP	OPOL1	OPOL0	UFOA1		UFOA0		REPMODE	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	CNTPRESC	0x0	RW	**Counter prescaler value**

Configure counting frequency of the CNT register. - Note - its not recommended to change this setting on the fly.

Value	Mode	Description
0	DIV1	CLK_CNT = (LETIMER LF CLK)/1
1	DIV2	CLK_CNT = (LETIMER LF CLK)/2
2	DIV4	CLK_CNT = (LETIMER LF CLK)/4
3	DIV8	CLK_CNT = (LETIMER LF CLK)/8
4	DIV16	CLK_CNT = (LETIMER LF CLK)/16
5	DIV32	CLK_CNT = (LETIMER LF CLK)/32
6	DIV64	CLK_CNT = (LETIMER LF CLK)/64
7	DIV128	CLK_CNT = (LETIMER LF CLK)/128
8	DIV256	CLK_CNT = (LETIMER LF CLK)/256

15:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	DEBUGRUN	0x0	RW	**Debug Mode Run Enable**

Set to keep the LETIMER running in debug mode.

Value	Mode	Description
0	DISABLE	LETIMER is frozen in debug mode
1	ENABLE	LETIMER is running in debug mode

11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	CNTTOPEN	0x0	RW	**Compare Value 0 Is Top Value**

When set, TOP value will be used as Counter Top Value

Value	Mode	Description
0	DISABLE	The top value of the LETIMER is 65535 (0xFFFF)

Bit	Name	Reset	Access	Description
	1	ENABLE		The top value of the LETIMER is given by COMP0
8	BUFTOP	0x0	RW	**Buffered Top**
	Set to load TOPBUFF into TOP when REP0 reaches 0 in BUFFERED mode, allowing a buffered top value.			
	Value	Mode		Description
	0	DISABLE		COMP0 is only written by software
	1	ENABLE		COMP0 is set to COMP1 when REP0 reaches 0
7	OPOL1	0x0	RW	**Output 1 Polarity**
	Defines the idle value of output 1.			
6	OPOL0	0x0	RW	**Output 0 Polarity**
	Defines the idle value of output 0.			
5:4	UFOA1	0x0	RW	**Underflow Output Action 1**
	Defines the action on OUT1 on a LETIMER underflow - IDLE/TOGGLE/PULSE/PWM			
	Value	Mode		Description
	0	NONE		LETIMERn_OUT1 is held at its idle value as defined by OPOL1
	1	TOGGLE		LETIMERn_OUT1 is toggled on CNT underflow
	2	PULSE		LETIMERn_OUT1 is held active for one LETIMER0 clock cycle on CNT underflow. The output then returns to its idle value as defined by OPOL1
	3	PWM		LETIMERn_OUT1 is set idle on CNT underflow, and active on compare match with COMP1
3:2	UFOA0	0x0	RW	**Underflow Output Action 0**
	Defines the action on OUT0 on a LETIMER underflow - IDLE/TOGGLE/PULSE/PWM			
	Value	Mode		Description
	0	NONE		LETIMERn_OUT0 is held at its idle value as defined by OPOL0
	1	TOGGLE		LETIMERn_OUT0 is toggled on CNT underflow
	2	PULSE		LETIMERn_OUT0 is held active for one LETIMER0 clock cycle on CNT underflow. The output then returns to its idle value as defined by OPOL0
	3	PWM		LETIMERn_OUT0 is set idle on CNT underflow, and active on compare match with COMP1
1:0	REPMODE	0x0	RW	**Repeat Mode**
	Repeat Mode - FREE/ONESHOT/BUFFERED/DOUBLE			
	Value	Mode		Description
	0	FREE		When started, the LETIMER counts down until it is stopped by software
	1	ONESHOT		The counter counts REP0 times. When REP0 reaches zero, the counter stops

Bit	Name	Reset	Access	Description
	2	BUFFERED		The counter counts REP0 times. If REP1 has been written, it is loaded into REP0 when REP0 reaches zero, otherwise the counter stops
	3	DOUBLE		Both REP0 and REP1 are decremented when the LETIMER wraps around. The LETIMER counts until both REP0 and REP1 are zero

17.14.5 LETIMER_CMD - Command Register

Offset	Bit Position																														
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												W(nB)	W(nB)	W(nB)	W(nB)	W(nB)
Name																												CTO1	CTO0	CLEAR	STOP	START

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	CTO1	0x0	W(nB)	**Clear Toggle Output 1**
	Set to drive toggle output 1 to its idle value			
3	CTO0	0x0	W(nB)	**Clear Toggle Output 0**
	Set to drive toggle output 0 to its idle value			
2	CLEAR	0x0	W(nB)	**Clear LETIMER**
	Set to clear LETIMER			
1	STOP	0x0	W(nB)	**Stop LETIMER**
	Set to stop LETIMER			
0	START	0x0	W(nB)	**Start LETIMER**
	Set to start LETIMER			

17.14.6 LETIMER_STATUS - Status Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	R
Name																															LETIMERLOCKSTATUS	RUNNING

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	LETIMERLOCKSTATUS	0x0	R	**LETIMER Lock Status**
	Indicates the current status of LETIMER Lock			

Value	Mode	Description
0	UNLOCKED	LETIMER registers are unlocked
1	LOCKED	LETIMER registers are locked

Bit	Name	Reset	Access	Description
0	RUNNING	0x0	R	**LETIMER Running**
	Set when LETIMER is running.			

17.14.7 LETIMER_CNT - Counter Value Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																							
Access									RW																							
Name									CNT																							

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:0	CNT	0x0	RW	**Counter Value**
	Use to read the current value of the LETIMER.			

17.14.8 LETIMER_COMP0 - Compare Value Register 0

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																						
Access									RW																						
Name									COMP0																						

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:0	COMP0	0x0	RW	**Compare Value 0**
	Compare value for LETIMER.			

17.14.9 LETIMER_COMP1 - Compare Value Register 1

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																						
Access									RW																						
Name									COMP1																						

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:0	COMP1	0x0	RW	**Compare Value 1**
	Compare and optionally buffered top value for LETIMER.			

17.14.10 LETIMER_TOP - Counter TOP Value Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																						
Access									RW																						
Name									TOP																						

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:0	TOP	0x0	RW	**Counter TOP Value**
	TOP will be used as Counter TOP Value if CNTTOPEN is set to 1			

17.14.11 LETIMER_TOPBUFF - Buffered Counter TOP Value

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																						
Access									RW																						
Name									TOPBUFF																						

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:0	TOPBUFF	0x0	RW	**Buffered Counter TOP Value**
	TOPBUFF will be used as Counter TOP Value in BUFFERED Mode if CNTTOPEN and BUFFTOP is set set to 1			

17.14.12 LETIMER_REP0 - Repeat Counter Register 0

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									REP0							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	REP0	0x0	RW	**Repeat Counter 0**
	Optional repeat counter.			

17.14.13 LETIMER_REP1 - Repeat Counter Register 1

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									REP1							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	REP1	0x0	RW	**Repeat Counter 1**
	Optional repeat counter or buffer for REP0.			

17.14.14 LETIMER_IF - Interrupt Flag Register

Offset	Bit Position																														
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												REP1	REP0	UF	COMP1	COMP0

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	REP1	0x0	RW	**Repeat Counter 1 Interrupt Flag**
	Set when repeat counter 1 reaches zero.			
3	REP0	0x0	RW	**Repeat Counter 0 Interrupt Flag**
	Set when repeat counter 0 reaches zero or when the REP1 interrupt flag is loaded into the REP0 interrupt flag.			
2	UF	0x0	RW	**Underflow Interrupt Flag**
	Set on LETIMER underflow.			
1	COMP1	0x0	RW	**Compare Match 1 Interrupt Flag**
	Set when LETIMER reaches the value of COMP1.			
0	COMP0	0x0	RW	**Compare Match 0 Interrupt Flag**
	Set when LETIMER reaches the value of COMP0.			

17.14.15 LETIMER_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												REP1	REP0	UF	COMP1	COMP0

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	REP1	0x0	RW	**Repeat Counter 1 Interrupt Enable**
	Repeat Counter 1 Interrupt Enable			
3	REP0	0x0	RW	**Repeat Counter 0 Interrupt Enable**
	Repeat Counter 0 Interrupt Enable			
2	UF	0x0	RW	**Underflow Interrupt Enable**
	Underflow Interrupt Enable			
1	COMP1	0x0	RW	**Compare Match 1 Interrupt Enable**
	Compare Match 1 Interrupt Enable			
0	COMP0	0x0	RW	**Compare Match 0 Interrupt Enable**
	Compare Match 0 Interrupt Enable			

17.14.16 LETIMER_LOCK - Configuration Lock Register

Offset	Bit Position																													
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W							
Name																									LETIMERLOCKKEY							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LETIMERLOCKKEY	0x0	W	**Configuration Lock Key**

Write any other value than the unlock code to lock LETIMER_EN, LETIMER_SWRST, LETIMER_CTRL, LETIMER_CMD, LETIMER_CNT, LETIMER_COMP0, LETIMER_COMP1, LETIMER_TOP, LETIMER_TOPBUFF, LETIMER_REP0, LETIMER_REP1 and PRSMODE registers from editing. Write the unlock code to unlock.

Value	Mode	Description
52476	UNLOCK	Write to unock LETIMER lockable registers

17.14.17 LETIMER_SYNCBUSY - Synchronization Busy Register

Offset	Bit Position																													
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0		0x0
Access																							R	R	R	R	R	R	R	R		R
Name																							CTO1	CTO0	CLEAR	STOP	START	REP1	REP0	TOP		CNT

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	CTO1	0x0	R	**Sync busy for CTO1**
	Sync busy for CTO1			
8	CTO0	0x0	R	**Sync busy for CTO0**
	Sync busy for CTO0			
7	CLEAR	0x0	R	**Sync busy for CLEAR**
	Sync busy for CLEAR			
6	STOP	0x0	R	**Sync busy for STOP**
	Sync busy for STOP			
5	START	0x0	R	**Sync busy for START**
	Sync busy for START			
4	REP1	0x0	R	**Sync busy for REP1**
	Sync busy for REP1			
3	REP0	0x0	R	**Sync busy for REP0**
	Sync busy for REP0			
2	TOP	0x0	R	**Sync busy for TOP**
	Sync busy for TOP			
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	CNT	0x0	R	**Sync busy for CNT**
	Sync busy for CNT			

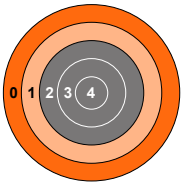
17.14.18 LETIMER_PRSMODE - PRS Input Mode Select Register

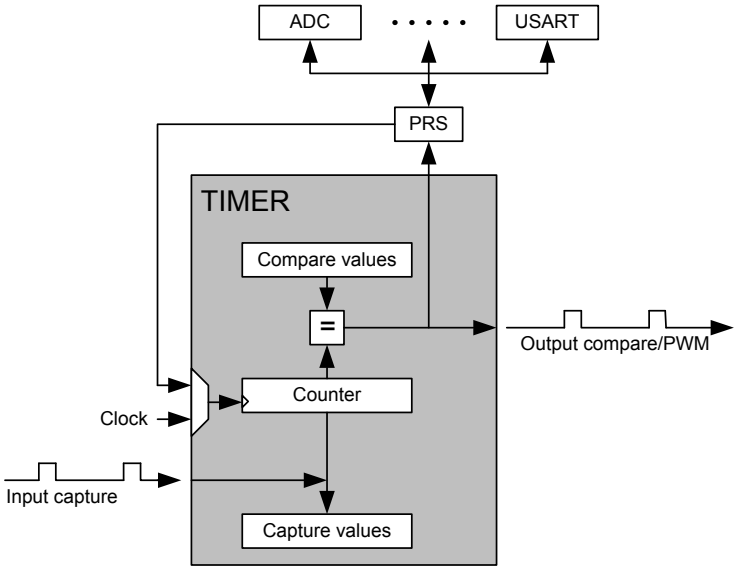
Offset	Bit Position																													
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0				0x0				0x0																			
Access					RW				RW				RW																			
Name					PRSCLEARMODE				PRSSTOPMODE				PRSSTARTMODE																			

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:26	PRSCLEARMODE	0x0	RW	**PRS Clear Mode**
	Mode-NONE/RISING/FALLING/BOTH			
	Value	Mode		Description
	0	NONE		PRS cannot clear the LETIMER
	1	RISING		Rising edge of selected PRS input can clear the LETIMER
	2	FALLING		Falling edge of selected PRS input can clear the LETIMER
	3	BOTH		Both the rising or falling edge of the selected PRS input can clear the LETIMER
25:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:22	PRSSTOPMODE	0x0	RW	**PRS Stop Mode**
	Mode-NONE/RISING/FALLING/BOTH			
	Value	Mode		Description
	0	NONE		PRS cannot stop the LETIMER
	1	RISING		Rising edge of selected PRS input can stop the LETIMER
	2	FALLING		Falling edge of selected PRS input can stop the LETIMER
	3	BOTH		Both the rising or falling edge of the selected PRS input can stop the LETIMER
21:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:18	PRSSTARTMODE	0x0	RW	**PRS Start Mode**
	Mode-NONE/RISING/FALLING/BOTH			
	Value	Mode		Description
	0	NONE		PRS cannot start the LETIMER
	1	RISING		Rising edge of selected PRS input can start the LETIMER

Bit	Name	Reset	Access	Description
	2	FALLING		Falling edge of selected PRS input can start the LETIMER
	3	BOTH		Both the rising or falling edge of the selected PRS input can start the LETIMER
17:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

18. TIMER - Timer/Counter





Quick Facts

What?

The TIMER (Timer/Counter) keeps track of timing and counts events, generates output waveforms, and triggers timed actions in other peripherals.

Why?

Most applications have activities that need to be timed accurately with as little CPU intervention and energy consumption as possible.

How?

The flexible 16/32-bit timer can be configured to provide PWM waveforms with optional dead-time insertion (e.g. motor control) or work as a frequency generator. The timer can also count events and control other peripherals through the PRS, which offloads the CPU and reduces energy consumption.

18.1 Introduction

The general purpose timer has 3 or 4 compare/capture channels for input capture and compare/Pulse-Width Modulation (PWM) output.

The TIMER module may be 16 or 32 bits wide. Some timers also include a Dead-Time Insertion module suitable for motor control applications.

Refer to the device data sheet to determine the capabilities (capture/compare channel count, width, and DTI) of each timer instance.

18.2 Features

- 16/32-bit auto reload up/down counter
  - Dedicated 16/32-bit reload register which serves as counter maximum
- 3 or 4 Compare/Capture channels
  - Individually configurable as either input capture or output compare/PWM
- Multiple Counter modes
  - Count up
  - Count down
  - Count up/down
  - Quadrature Decoder
  - Direction and count from external pins
- 2x Count Mode
- Counter control from PRS or external pin
  - Start
  - Stop
  - Reload and start
- Inter-Timer connection
  - Allows 32-bit counter mode
  - Start/stop synchronization between several timers
- Input Capture
  - Period measurement
  - Pulse width measurement
  - Two capture registers for each capture channel
    - Capture on either positive or negative edge
    - Capture on both edges
  - Optional digital noise filtering on capture inputs
- Output Compare
  - Compare output toggle/pulse on compare match
  - Immediate update of compare registers
- PWM
  - Up-count PWM
  - Up/down-count PWM
  - Predictable initial PWM output state (configured by SW)
  - Buffered compare register to ensure glitch-free update of compare values
  - Output re-timing to mitigate RF interference
- Clock sources
  - $HFPERCLK_{TIMERn}$
    - 10-bit Prescaler
  - External pin
  - Peripheral Reflex System
- Debug mode
  - Configurable to either run or stop when processor is stopped (halt/breakpoint)
- Interrupts, PRS output and/or DMA request on:
  - Underflow
  - Overflow
  - Compare/Capture event

- Dead-Time Insertion Unit
  - Complementary PWM outputs with programmable dead-time
    - Dead-time is specified independently for rising and falling edge
      - 10-bit prescaler
      - 6-bit time value
    - Outputs have configurable polarity
    - Outputs can be set inactive individually by software.
  - Configurable action on fault
    - Set outputs inactive
    - Clear output
    - Tristate output
  - Individual fault sources
    - One or two PRS signals
    - Debugger
      - Support for automatic restart
    - Core lockup
    - EM2/EM3 entry
  - Configuration lock

18.3 Functional Description

An overview of the TIMER module is shown in Figure 18.1 TIMER Block Overview on page 479 and it consists of a 16/32 bit up/down counter with 3 compare/capture channels connected to pins TIMn_CC0, TIMn_CC1, and TIMn_CC2.

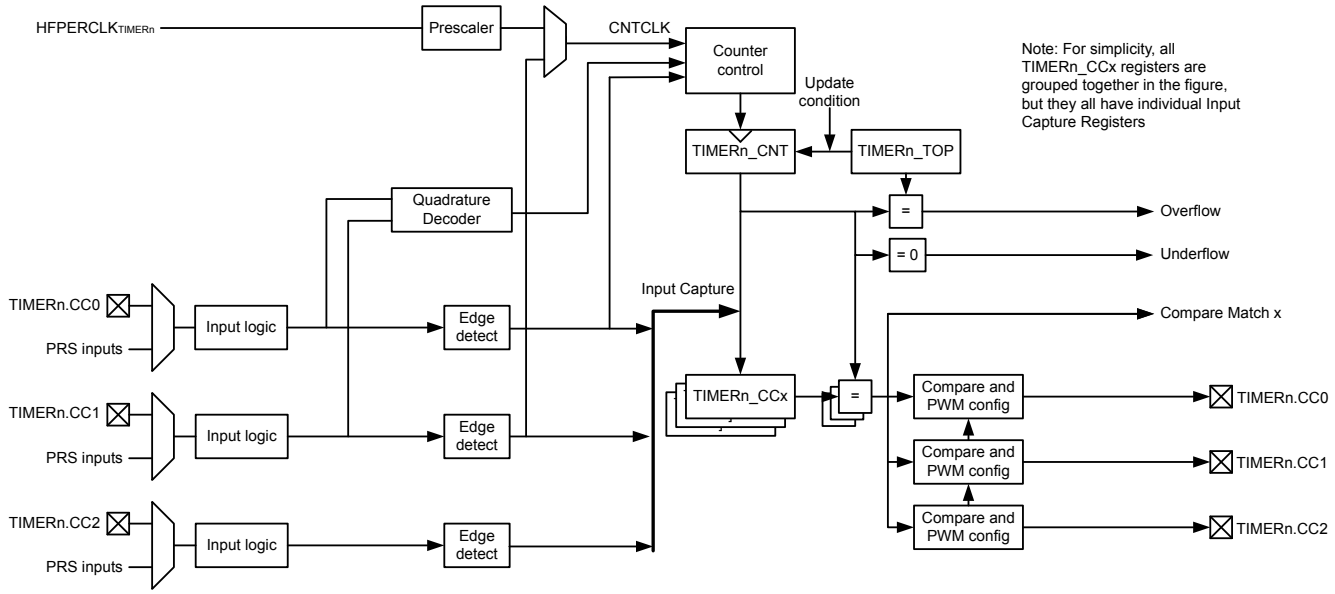


Figure 18.1. TIMER Block Overview

18.3.1 Register Access

The timer module interface consists of multiple register types. Registers of type "RW CONFIG" should only be written when the module is disabled (TIMERn_EN_EN = 0). Registers of type "W SYNC", "R SYNC" or "RW SYNC" should only be read or written when the module is enabled (TIMERn_EN_EN = 1). A typical setup sequence for a TIMER module is as follows:

1. With the TIMER disabled (TIMERn_EN_EN = 0), program any CONFIG registers required for the application.
2. Enable the TIMER by setting EN in TIMERn_EN to 1.
3. Program any non-CONFIG registers required for the application.
4. The TIMER is then ready for use.

18.3.2 Counter Modes

The timer consists of a counter that can be configured to the following modes, using the MODE field in TIMERn_CFG:

- Up-count: Counter counts up until it reaches the value in TIMERn_TOP, where it is reset to 0 before counting up again.
- Down-count: The counter starts at the value in TIMERn_TOP and counts down. When it reaches 0, it is reloaded with the value in TIMERn_TOP.
- Up/Down-count: The counter starts at 0 and counts up. When it reaches the value in TIMERn_TOP, it counts down until it reaches 0 and starts counting up again.
- Quadrature Decoder: Two input channels where one determines the count direction, while the other pin triggers a clock event.

In addition to the TIMER modes listed above, the TIMER also supports a 2x count mode. In this mode the counter increments/decrements by 2 on each clock edge. The 2x count mode can be used to double the PWM frequency when the compare/capture channel is put into PWM mode. The 2x count mode is enabled by setting the X2CNT bitfield in the TIMERn_CTRL register.

The counter value can be read or written by software any time the module is enabled by accessing the CNT field in TIMERn_CNT.

18.3.2.1 Events

The main counter can generate overflow and underflow events during operation.

Overflow (TIMERn_IF_OF) is set when the counter value shifts from TIMERn_TOP to the next value when counting up. In up-count mode and quadrature decoder mode the next value is 0. In up/down-count mode, the next value is TIMERn_TOP-1.

Underflow (TIMERn_IF_UF) is set when the counter value shifts from 0 to the next value when counting down. In down-count mode and quadrature decoder mode, the next value is TIMERn_TOP. In up/down-count mode the next value is 1.

An update event occurs on overflow in up-count mode and on underflow in down-count or up/down count mode. Additionally, an update event also occurs on overflow and underflow in quadrature decoder . This event is used to time updates of buffered values.

18.3.2.2 Operation

Figure 18.2 TIMER Hardware Timer/Counter Control on page 481 shows the hardware timer/counter control. Software can start or stop the counter by setting the START or STOP bits in TIMERn_CMD. The counter value (CNT in TIMERn_CNT) can always be written by software to any 16/32-bit value.

It is also possible to control the counter through either an external pin or PRS input. This is done through the input logic for the compare/capture Channel 0. The timer/counter allows individual actions (start, stop, reload) to be taken for rising and falling input edges. This is configured in the RISEA and FALLA fields in TIMERn_CTRL. The reload value is 0 in up-count and up/down-count mode and TOP in down-count mode.

The RUNNING bit in TIMERn_STATUS indicates if the timer is running or not. If the SYNC bit in TIMERn_CFG is set, the timer is started/stopped/reloaded (external pin or PRS) when any of the other timers are started/stopped/reloaded.

The DIR bit in TIMERn_STATUS indicates the counting direction of the timer at any given time. The counter value can be read or written by software through the CNT field in TIMERn_CNT. In Up/Down-Count mode the count direction will be set to up if the CNT value is written by software.

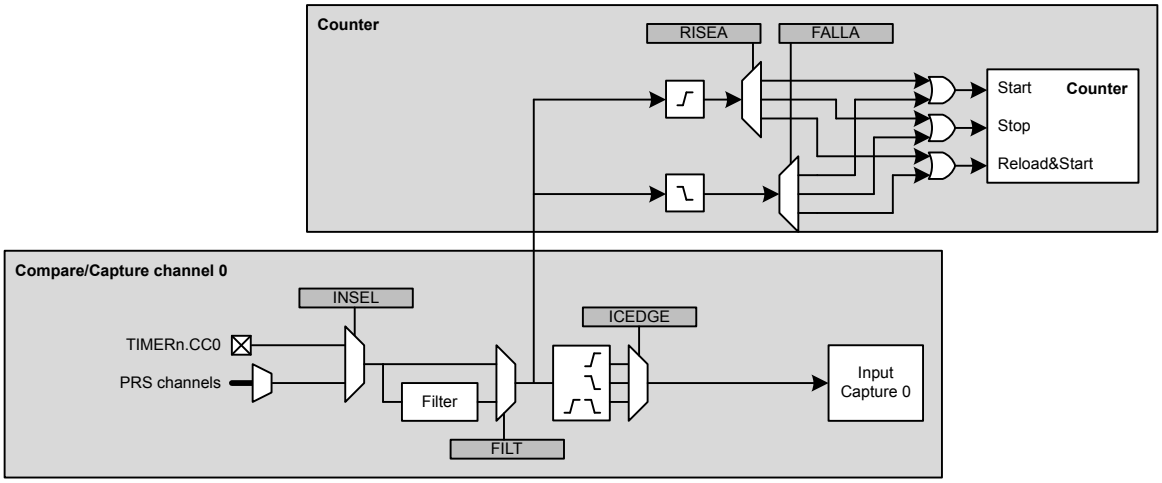


Figure 18.2. TIMER Hardware Timer/Counter Control

18.3.2.3 Clock Source

The counter can be clocked from several sources, which are all synchronized with the incoming peripheral clock for the timer. See Figure 18.3 TIMER Clock Selection on page 481.

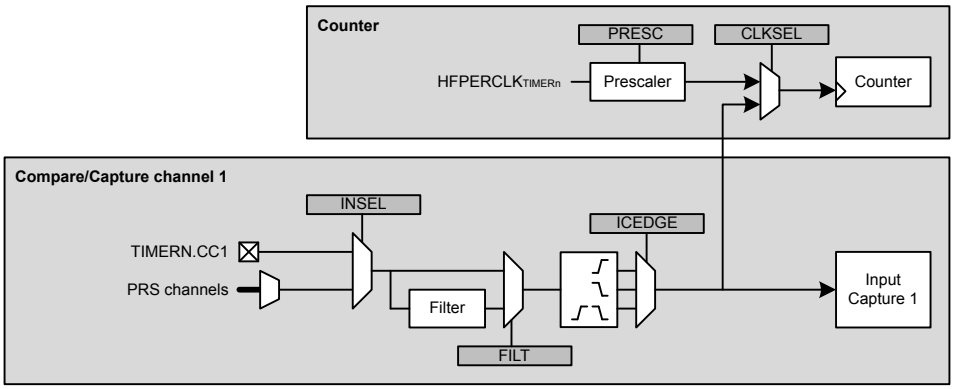


Figure 18.3. TIMER Clock Selection

18.3.2.4 Peripheral Clock

The peripheral clock for the timer ($HFPERCLK_{TIMERn}$) clocks the logic for the timer block, even when it is not the selected clock source.

All TIMER instances in this device family use EM01GRPACLK selected in CMU_EM01GRPACLKCTRL_CLKSEL as their peripheral clock source ($HFPERCLK_{TIMERn}$).

The peripheral clock to each timer can be used as a source with a configurable 10-bit prescaler. The PRESC bitfield in TIMERn_CFG sets the prescaler value, and the incoming peripheral clock will be divided by a factor of (PRESC+1). However, if 2x count mode is enabled and the compare/capture channels are configured for PWM mode, the CC output is updated on both clock edges, so prescaling the peripheral clock will produce an incorrect result. The internal prescale counter is stopped and reset when the timer is stopped.

18.3.2.5 Compare/Capture Channel 1 Input

The timer can also be clocked by positive and/or negative edges on the compare/capture channel 1 input. This input can either come from the TIMn_CC1 pin or one of the PRS channels. The input signal must not have a higher frequency than $f_{HFPERCLK\_TIMERn}/3$ when running from a pin input or a PRS input with FILT enabled in TIMERn_CCx_CFG. When running from PRS without FILT, the frequency can be as high as $f_{HFPERCLK\_TIMERn}$. Note that when clocking the timer from the same pulse that triggers a start (through RISEA/FALLA in TIMERn_CTRL), the starting pulse will not update the counter value.

18.3.2.6 Underflow/Overflow From Neighboring Timer

All timers are linked together (see Figure 18.4 TIMER Connections on page 482), allowing timers to count on overflow/underflow from the lower numbered neighbouring timers to form a larger timer. Note that all timers must be set to count the same direction and less significant timer(s) can only be set to count up or down.

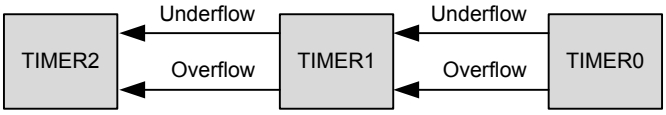


Figure 18.4. TIMER Connections

18.3.2.7 One-Shot Mode

By default, the counter counts continuously until it is stopped. If the OSMEN bit is set in the TIMERn_CFG register, however, the counter is disabled by hardware on the first *update event* (see 18.3.2.1 Events). Note that when the counter is running with CC1 as clock source and OSMEN is set, a CC1 capture event will not take place on the *update event* (CC1 rising edge) that stops the timer.

18.3.2.8 Top Value Buffer

The TIMERn_TOP register can be altered either by writing it directly or by writing to the TIMER_TOPB (buffer) register. When writing to the buffer register the TIMERn_TOPB register will be written to TIMERn_TOP on the next *update event*. Buffering ensures that the TOP value is not set below the actual count value. The TOPBV flag in TIMERn_STATUS indicates whether the TIMERn_TOPB register contains data that has not yet been written to the TIMERn_TOP register (see Figure 18.5 TIMER TOP Value Update Functionality on page 483).

Note: When writing to TIMERn_TOP register directly, the TIMERn_TOPB register value will be invalidated and the TOPBV flag will be cleared. This prevents TIMERn_TOP register from being immediately updated by an existing valid TIMERn_TOPB value during the next *update event*.

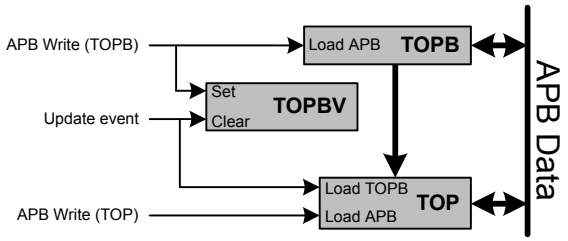


Figure 18.5. TIMER TOP Value Update Functionality

18.3.2.9 Quadrature Decoder

Quadrature decoding mode is used to track motion and determine both rotation direction and position. The quadrature decoder uses two input channels that are 90 degrees out of phase (see Figure 18.6 TIMER Quadrature Encoded Inputs on page 484).

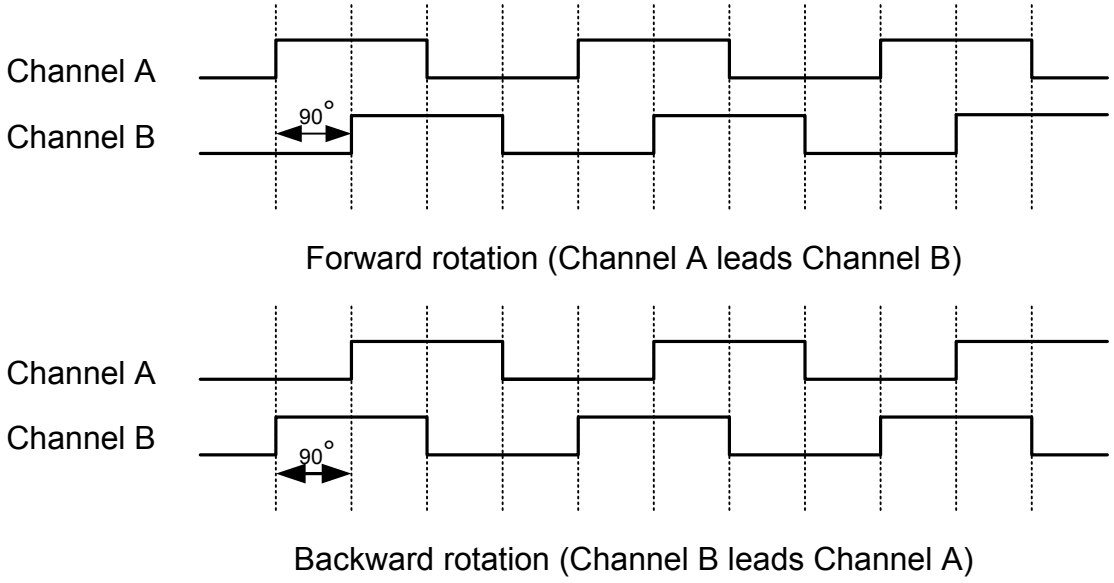


Figure 18.6. TIMER Quadrature Encoded Inputs

In the timer these inputs are tapped from the compare/capture channel 0 (Channel A) and 1 (Channel B) inputs before edge detection. The timer/counter then increments or decrements the counter, based on the phase relation between the two inputs. The DIRCHG flag in TIMERn_IF is set if the count direction changes in quadrature decoder mode. The quadrature decoder supports two channels, but if a third channel (Z-terminal) is available, this can be connected to an external interrupt and trigger a counter reset from the interrupt service routine. By connecting a periodic signal from another timer as input capture on compare/capture Channel 2, it is also possible to calculate speed and acceleration.

Note: In quadrature decoder mode, overflow and underflow triggers an *update event.*

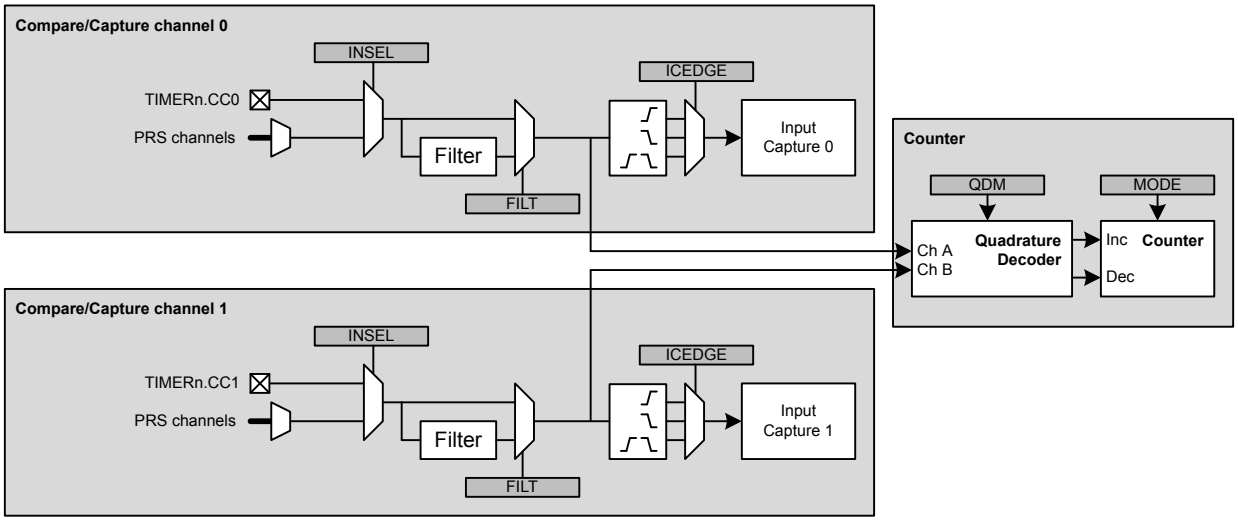


Figure 18.7. TIMER Quadrature Decoder Configuration

The quadrature decoder can be set in either X2 or X4 mode, which is configured in the QDM bit in TIMERn_CFG. See Figure 18.7 TIMER Quadrature Decoder Configuration on page 484

18.3.2.10 X2 Decoding Mode

In X2 Decoding mode, the counter increments or decrements on every edge of Channel A, see Table 18.1 TIMER Counter Response in X2 Decoding Mode on page 485 and Figure 18.8 TIMER X2 Decoding Mode on page 485.

Table 18.1. TIMER Counter Response in X2 Decoding Mode

Channel B	Channel A	
	Rising	Falling
0	Increment	Decrement
1	Decrement	Increment

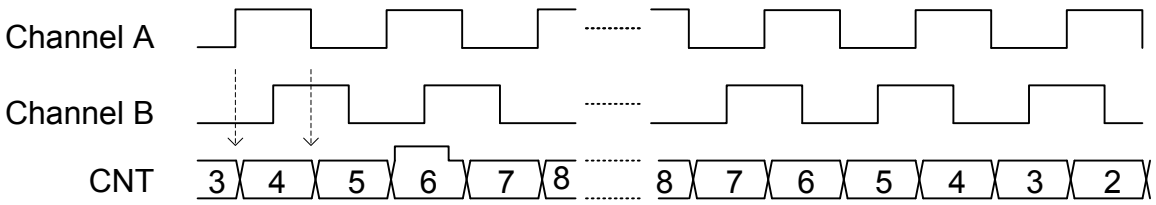


Figure 18.8. TIMER X2 Decoding Mode

18.3.2.11 X4 Decoding Mode

In X4 Decoding mode, the counter increments or decrements on every edge of Channel A and Channel B, see Figure 18.9 TIMER X4 Decoding Mode on page 485 and Table 18.2 TIMER Counter Response in X4 Decoding Mode on page 485.

Table 18.2. TIMER Counter Response in X4 Decoding Mode

Opposite Channel	Channel A		Channel B	
	Rising	Falling	Rising	Falling
Channel A = 0			Decrement	Increment
Channel A = 1			Increment	Decrement
Channel B = 0	Increment	Decrement		
Channel B = 1	Decrement	Increment		

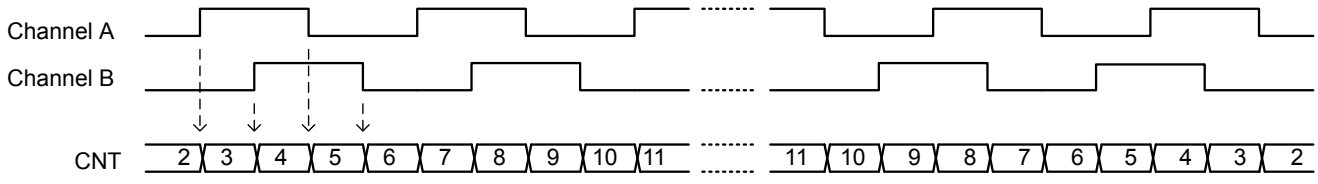


Figure 18.9. TIMER X4 Decoding Mode

18.3.2.12 Rotational Position

To calculate a position Figure 18.10 TIMER Rotational Position Equation on page 486 can be used.

$$pos° = (CNT/X \times N) \times 360°$$

Figure 18.10. TIMER Rotational Position Equation

where X = Encoding type and N = Number of pulses per revolution.

18.3.3 Compare/Capture Channels

The timer contains compare/capture channels, which can be independently configured in the following modes:

1. Input Capture
2. Output Compare
3. PWM

18.3.3.1 Input Pin Logic

Each compare/capture channel can be configured as an input source for the Capture Unit or as external clock source for the timer (see Figure 18.11 TIMER Input Pin Logic on page 486). Compare/capture channels 0 and 1 are the inputs for the quadrature decoder . The input channel can be filtered before it is used, which requires the input to remain stable for up to 5 cycles in a row before the input is propagated to the output.

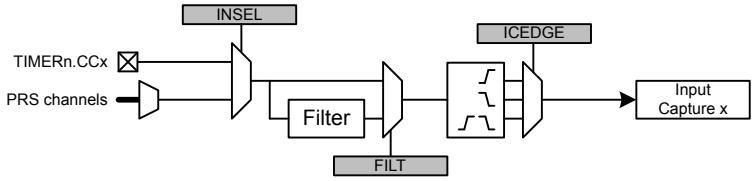


Figure 18.11. TIMER Input Pin Logic

The capture input to the timer may be selected from the dedicated CCx signal for the channel, or a PRS signal. INSEL in TIMERn_CCx_CFG determines the input to the channel. When set to PIN, the selected CCx pin will be used. When INSEL is set to PRSSYNC, a synchronous PRS channel is selected as the source. The synchronous PRS channel is determined by the SPRSSEL field in the PRS_TIMERn_CCx register. Setting INSEL to PRSASYNCLEVEL or PRSASYNCPULSE selects an asynchronous PRS channel as the source. The aynchronous PRS channel is determined by the PRSSEL field in the PRS_TIMERn_CCx register.

The PIN and PRSASYNCLEVEL selections are qualified by a 2-clock input sampler. To recognize and capture the incoming signal, it must be at the new level for at least 2 $HFPERCLK_{TIMERn}$ clock cycles. An additional 5 $HFPERCLK_{TIMERn}$ cycles of filtering can be applied to the signal by enabling the FILT bit in TIMERn_CCx_CFG.

The PRSASYNCPULSE selection can be used to capture higher-speed pulses on an asynchronous PRS input. The input logic for this selection does not qualify the level of the incoming signal. Instead, it will recognize positive or negative edges directly. While the pulse time can be shorter than 1 $HFPERCLK_{TIMERn}$, this mode requires at least 3 $HFPERCLK_{TIMERn}$ clocks between adjacent events. The FILT option is not used in this mode.

Synchronous PRS signals are inherently synchronized to the module clock, and the 2-clock input sampler is not used. However, it is possible to use FILT to enable the 5 $HFPERCLK_{TIMERn}$ filter when using the PRSSYNC option.

18.3.3.2 Compare/Capture Registers

The compare/capture channel registers are prefixed with TIMERn_CCx_, where the x stands for the channel number. Since the compare/capture channels serve three functions (input capture, compare, PWM), different registers are used, depending on the mode the channel is set in.

18.3.3.3 Input Capture

In input capture, the counter value (TIMERn_CNT) can be captured in the Input Capture Register (TIMERn_CCx_ICF) (see Figure 18.12 TIMER Input Capture on page 487). The CCPOL bits in TIMERn_STATUS indicate the polarity of the edge that triggered the capture in TIMERn_CCx_ICF.

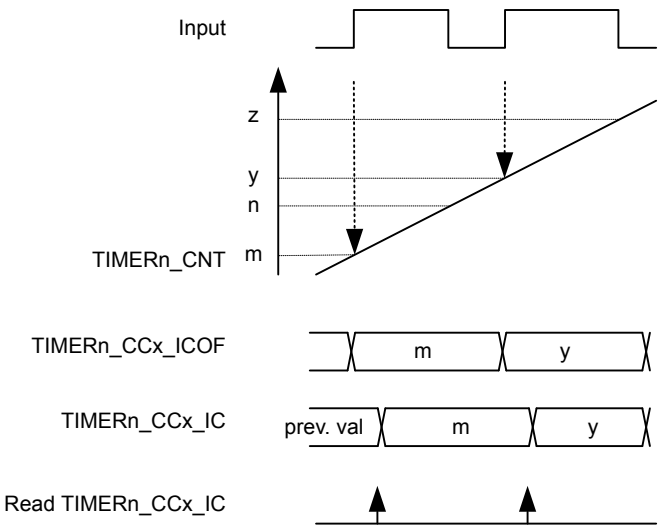


Figure 18.12. TIMER Input Capture

Input captures are buffered into a 2-entry FIFO, allowing 2 subsequent capture events to take place before a read-out is required. Reading TIMERn_CCx_ICF from software or DMA pops the oldest unread value from the FIFO. If TIMERn_CCx_ICF is read when the FIFO is empty (ICFEMPTY in TIMERn_STATUS = 1), the FIFO underflow flag for the channel (ICFUF in TIMERn_IF) will be set. The Input Capture Overflow Register (TIMERn_CCx_ICOF) always contains the newest value in the FIFO. If a new capture is triggered while the FIFO is full, the value in TIMERn_CCx_ICOF will be over-written with the latest value and the FIFO overflow flag (ICFOF in TIMERn_IF) for the channel will be set. Reading TIMERn_CCx_ICOF does not alter the FIFO contents.

The input capture FIFO also has a programmable watermark level that can be configured to generate interrupts or trigger DMA requests when a certain number of empty spots are left in the FIFO. The ICFWLFULL flag inTIMERn_IF will be set when the number of empty spots left in the FIFO is less than or equal to the watermark level programmed in TIMERn_CCx_CFG_ICFWL. At a minimum, a TIMER module will have two FIFO entries, but may have more on future devices.

The ICFEMPTY flag in TIMERn_STATUS indicates when the capture buffer is empty. When this bit reads '0', there is a valid unread capture in the FIFO.

Note: In input capture mode, the timer will only trigger interrupts when it is running.

18.3.3.4 Period/Pulse-Width Capture

Period and/or pulse-width capture can only be possible with Channel 0 (CC0), because this is the only channel that can start and stop the timer. This can be done by setting the RISEA field in TIMERn_CTRL to Clear&Start, and selecting the desired input from either external pin or PRS, see Figure 18.13 TIMER Period and/or Pulse Width Capture on page 488. For period capture, the compare/capture channel should then be set to input capture on a rising edge of the same input signal. To capture the width of a high pulse, the compare/capture channel should be set to capture on a falling edge of the input signal. To measure the low pulse-width of a signal, opposite polarities should be chosen.

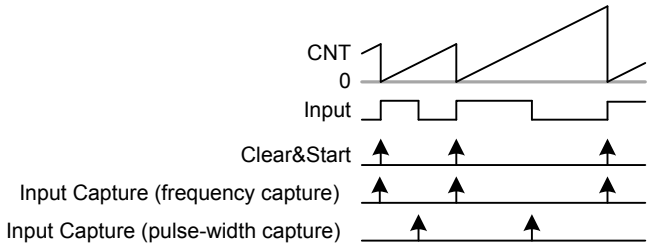


Figure 18.13. TIMER Period and/or Pulse Width Capture

18.3.3.5 Compare

Each compare/capture channel contains a comparator which outputs a compare match if the contents of TIMERn_CCx_OC matches the counter value, see Figure 18.14 TIMER Block Diagram Showing Comparison Functionality on page 489. In compare mode, each compare channel can be configured to either set, clear or toggle the output on an event (compare match, overflow or underflow). The output from each channel is represented as an alternative function on the port it is connected to, which needs to be enabled for the CC outputs to propagate to the pins.

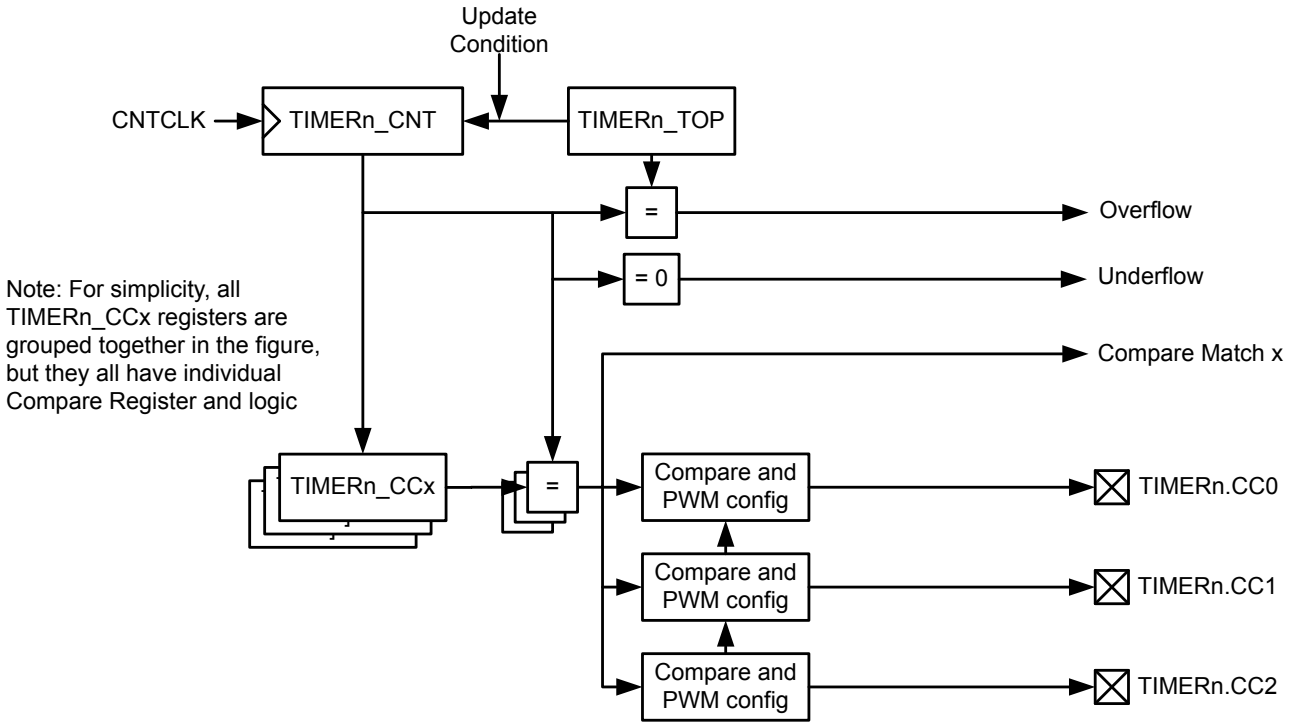


Figure 18.14. TIMER Block Diagram Showing Comparison Functionality

The compare output is delayed by one cycle to allow for full 0% to 100% PWM generation. If occurring in the same cycle, match action will have priority over overflow or underflow action.

The input selected (through PRSSEL in PRS_CONSUMER_TIMERn_CCx, INSEL and FILT in TIMERn_CCx_CFG) for the CC channel will also be sampled on compare match and the result is found in the CCPOL bits in TIMERn_STATUS. It is also possible to configure the CCPOL to always track the inputs by setting ATI in TIMERn_CFG.

Note: When using synchronous PRS sources, it is recommended to configure the PRS consumer registers prior to selecting PRS triggering to avoid any false triggers.

The COIST bit in TIMERn_CCx_CFG is the initial state of the compare/PWM output. The COIST bit can also be used as an initial value to the compare outputs on a reload-start when RSSCOIST is set in TIMERn_CFG. Also the resulting output can be inverted by setting OUTINV in TIMERn_CCx_CTRL. It is recommended to turn off the CC channel before configuring the output state to avoid any unwanted pulses on the output. The CC channel can be turned off by setting MODE to OFF in TIMER_CCx_CFG. The following figure shows the output logic for the TIMER module.

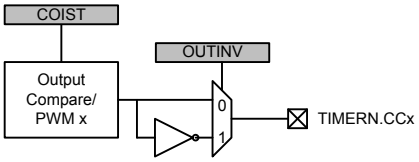


Figure 18.15. TIMER Output Logic

18.3.3.6 Compare Mode Registers

When running in output compare or PWM mode, the value in TIMERn_CCx_OC will be compared against the count value. In Compare mode the output can be configured to toggle, clear or set on compare match, overflow, and underflow through the CMOA, COFOA and CUFOA fields in TIMERn_CCx_CTRL. TIMERn_CCx_OC can be accessed directly or through the buffer register TIMERn_CCx_OCB, see Figure 18.16 TIMER Output Compare/PWM Buffer Functionality Detail on page 490. When writing to the buffer register, the value in TIMERn_CCx_OCB will be written to TIMERn_CCx_OC on the next *update event*. This functionality ensures glitch free PWM outputs. The OCBV flag in TIMERn_STATUS indicates whether the TIMERn_CCx_OCB register contains data that has not yet been written to the TIMERn_CCx_OC register. Note that when writing 0 to TIMERn_CCx_OCB in up-down count mode the OC value is updated when the timer counts from 0 to 1. Thus, the compare match for the next period will not happen until the timer reaches 0 again on the way down.

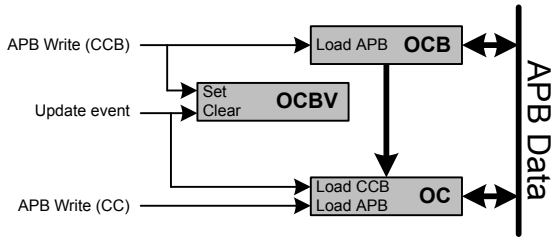


Figure 18.16. TIMER Output Compare/PWM Buffer Functionality Detail

18.3.3.7 Frequency Generation (FRG)

Frequency generation (see Figure 18.17 TIMER Up-count Frequency Generation on page 491) can be achieved in compare mode by:

- Setting the counter in up-count mode
- Enabling buffering of the TOP value.
- Setting the CC channels overflow action to toggle

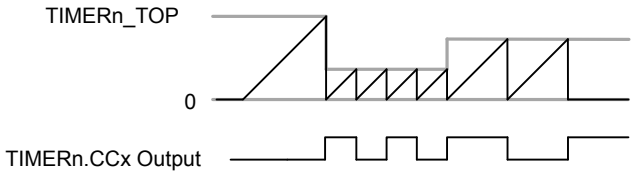


Figure 18.17. TIMER Up-count Frequency Generation

The output frequency is given by Figure 18.18 TIMER Up-count Frequency Generation Equation on page 491

$$f_{FRG} = f_{HFPERCLK\_TIMERn} / [2 \times (PRESC + 1) \times (TOP + 1)]$$

Figure 18.18. TIMER Up-count Frequency Generation Equation

The figure below provides cycle accurate timing and event generation information for frequency generation.

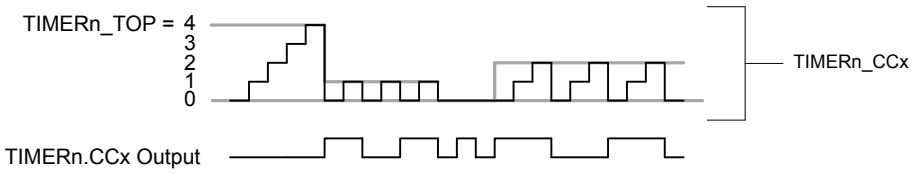


Figure 18.19. TIMER Up-count Frequency Generation Detail

18.3.3.8 Pulse-Width Modulation (PWM)

In PWM mode, TIMERn_CCx_OC is buffered to avoid glitches in the output. The settings in the Compare Output Action configuration bits are ignored in PWM mode and PWM generation is only supported for up-count and up/down-count mode.

18.3.3.9 Up-count (Single-slope) PWM

If the counter is set to up-count and the compare/capture channel is put in PWM mode, single slope PWM output will be generated (see Figure 18.20 TIMER Up-count PWM Generation on page 492). In up-count mode the PWM period is TOP+1 cycles and the PWM output will be high for a number of cycles equal to TIMERn_CCx_OC. This means that a constant high output is achieved by setting TIMERn_CCx_OC to TOP+1 or higher. The PWM resolution (in bits) is then given by Figure 18.21 TIMER Up-count PWM Resolution Equation on page 492.

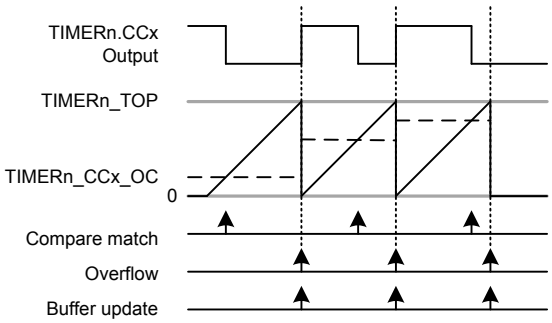


Figure 18.20. TIMER Up-count PWM Generation

$$R_{PWM_{up}} = \log(TOP+1)/\log(2)$$

Figure 18.21. TIMER Up-count PWM Resolution Equation

The PWM frequency is given by Figure 18.22 TIMER Up-count PWM Frequency Equation on page 492:

$$f_{PWM_{up}} = f_{HFPERCLK\_TIMERn} / [(PRESC + 1) \times (TOP + 1)]$$

Figure 18.22. TIMER Up-count PWM Frequency Equation

The high duty cycle is given by Figure 18.23 TIMER Up-count Duty Cycle Equation on page 492

$$DS_{up} = OCx/(TOP+1)$$

Figure 18.23. TIMER Up-count Duty Cycle Equation

The figure below provides cycle accurate timing and event generation information for up-count mode.

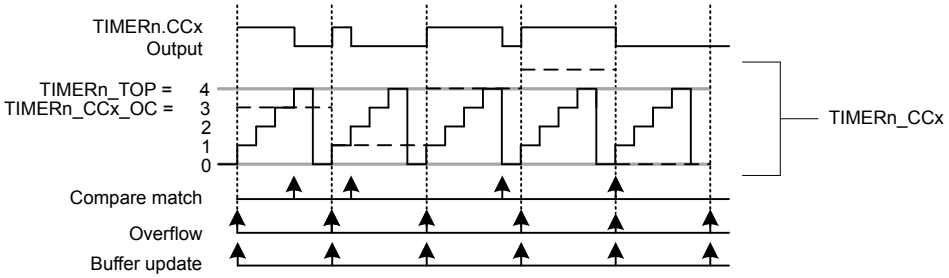


Figure 18.24. TIMER Up-count PWM Generation Detail

18.3.3.10 2x Count Mode (Up-count)

When the timer is set in 2x mode, the TIMER will count up by two for every (prescaled) clock. This will in effect make any odd Top value be rounded down to the closest even number. Similarly, any odd OC value will generate a match on the closest lower even value as shown in Figure 18.25 TIMER CC Out in 2x Mode on page 493

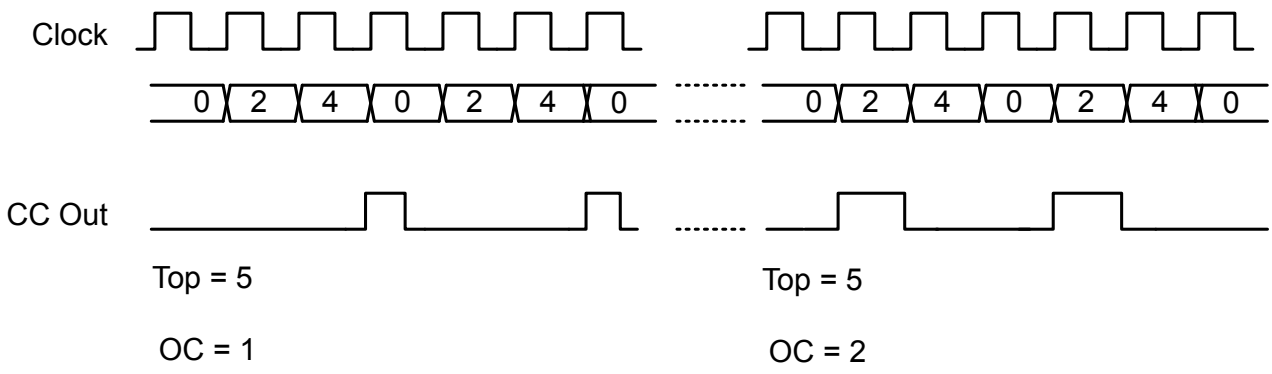


Figure 18.25. TIMER CC Out in 2x Mode

The PWM resolution is given by Figure 18.26 TIMER 2x PWM Resolution Equation on page 493.

$$R_{PWM_{2xmode}} = \log(TOP/2+1)/\log(2)$$

Figure 18.26. TIMER 2x PWM Resolution Equation

The PWM frequency is given by Figure 18.27 TIMER 2x Mode PWM Frequency Equation (Up-count) on page 493:

$$f_{PWM_{2xmode}} = f_{HFPERCLK\_TIMERn} / [(PRESC + 1) \times (\text{floor}(TOP/2)+1)]$$

Figure 18.27. TIMER 2x Mode PWM Frequency Equation (Up-count)

The high duty cycle is given by Figure 18.28 TIMER 2x Mode Duty Cycle Equation on page 493

$$DS_{2xmode} = OCx/((\text{floor}(TOP/2)+1)*2)$$

Figure 18.28. TIMER 2x Mode Duty Cycle Equation

18.3.3.11 Up/Down-count (Dual-slope) PWM

If the counter is set to up-down count and the compare/capture channel is put in PWM mode, dual slope PWM output will be generated by Figure 18.29 TIMER Up/Down-count PWM Generation on page 494. The resolution (in bits) is given by Figure 18.30 TIMER Up/Down-count PWM Resolution Equation on page 494.

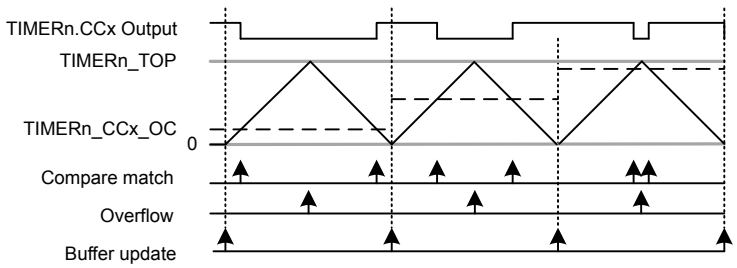


Figure 18.29. TIMER Up/Down-count PWM Generation

$$R_{PWM_{up/down}} = \log(TOP+1)/\log(2)$$

Figure 18.30. TIMER Up/Down-count PWM Resolution Equation

The PWM frequency is given by Figure 18.31 TIMER Up/Down-count PWM Frequency Equation on page 494:

$$f_{PWM_{up/down}} = f_{HFPERCLK\_TIMERn} / (2 \times (PRESC + 1) \times TOP))$$

Figure 18.31. TIMER Up/Down-count PWM Frequency Equation

The high duty cycle is given by Figure 18.32 TIMER Up/Down-count Duty Cycle Equation on page 494

$$DS_{up/down} = OCx/TOP$$

Figure 18.32. TIMER Up/Down-count Duty Cycle Equation

The figure below provides cycle accurate timing and event generation information for up-count mode.

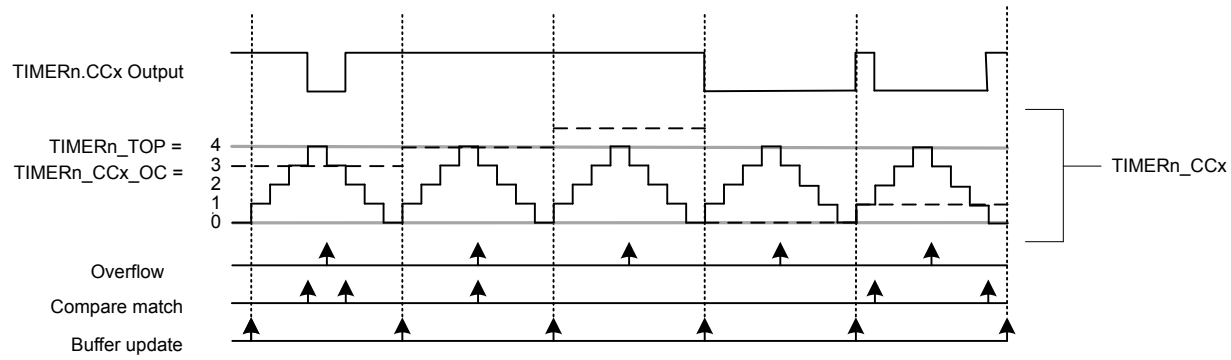


Figure 18.33. TIMER Up/Down-count PWM Generation

18.3.3.12 2x Count Mode (Up/Down-count)

When the timer is set in 2x mode, the TIMER will count up/down by two. This will in effect make any odd Top value be rounded down to the closest even number. Similarly, any odd OC value will generate a match on the closest lower even value as shown in Figure 18.34 TIMER CC Out in 2x mode on page 495

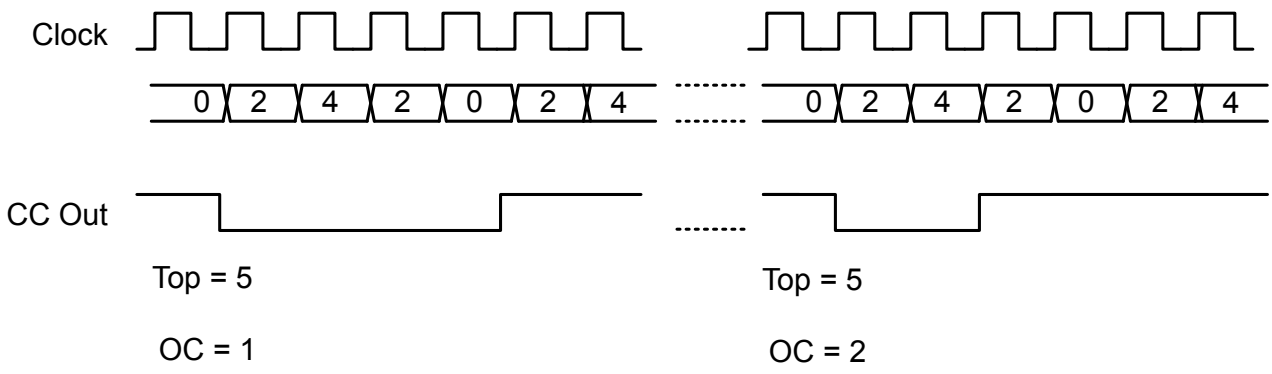


Figure 18.34. TIMER CC Out in 2x mode

Figure 18.35 TIMER 2x PWM Resolution Equation on page 495.

$$R_{PWM_{2xmode}} = \log(TOP/2+1)/\log(2)$$

Figure 18.35. TIMER 2x PWM Resolution Equation

The PWM frequency is given by Figure 18.36 TIMER 2x Mode PWM Frequency Equation (Up/Down-count) on page 495:

$$f_{PWM_{2xmode}} = f_{HFPERCLK\_TIMERn} / (2 \times (PRESC + 1) \times (\text{floor}(TOP/2)))$$

Figure 18.36. TIMER 2x Mode PWM Frequency Equation (Up/Down-count)

The high duty cycle is given by two equations based on the OCx values.Figure 18.37 TIMER 2x Mode Duty Cycle Equation for OCx = 1 or OCx = Even on page 495 and Figure 18.38 TIMER 2x Mode Duty Cycle Equation for all Other OCx = Odd Values on page 495

$$DS_{2xmode} = (OCx*2)/(\text{floor}(TOP/2)*4)$$

Figure 18.37. TIMER 2x Mode Duty Cycle Equation for OCx = 1 or OCx = Even

$$DS_{2xmode} = (OCx*2 - OCx)/(\text{floor}(TOP/2)*4)$$

Figure 18.38. TIMER 2x Mode Duty Cycle Equation for all Other OCx = Odd Values

18.3.3.13 Timer Configuration Lock

To prevent software errors from making changes to the timer configuration, a configuration lock is available. Writing any value but 0xCE80 to LOCKKEY in TIMERn_LOCK will lock writes to TIMERn_CTRL, TIMERn_CFG, TIMERn_CMD, TIMERn_TOP, TIMERn_TOPB, TIMERn_CNT, TIMERn_CCx_CTRL, TIMERn_CCx_CFG, TIMERn_CCx_OC, and TIMERn_CCx_OCB. To unlock the registers, write 0xCE80 to LOCKKEY in TIMERn_LOCK. The value of TIMERLOCKSTATUS in TIMERn_STATUS is 1 when the lock is active, and 0 when the registers are unlocked.

18.3.4 Dead-Time Insertion Unit

Some timer modules include a Dead-Time Insertion unit suitable for motor control applications. Refer to the device data sheet to check which timer instances have this feature. The example settings in this section are for TIMER0, but identical settings can be used for other timers with DTI as well. The Dead-Time Insertion Unit aims to make control of brushless DC (BLDC) motors safer and more efficient by introducing complementary PWM outputs with dead-time insertion and fault handling, see Figure 18.39 TIMER Dead-Time Insertion Unit Overview on page 496.

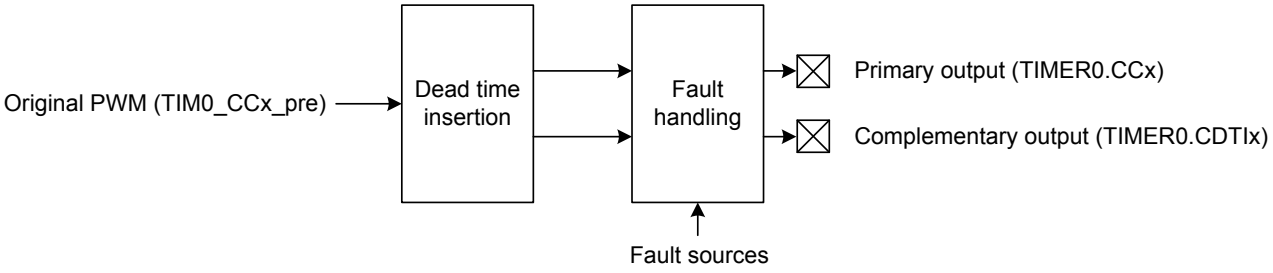


Figure 18.39. TIMER Dead-Time Insertion Unit Overview

When used for motor control, the PWM outputs TIM0_CC0, TIM0_CC1 and TIM0_CC2 are often connected to the high-side transistors of a triple half-bridge setup (UH, VH and WH), and the complementary outputs connected to the respective low-side transistors (UL, VL, WL shown in Figure 18.40 TIMER Triple Half-Bridge on page 496). Transistors used in such a bridge often do not open/close instantaneously, and using the exact complementary inputs for the high and low side of a half-bridge may result in situations where both gates are open. This can give unnecessary current-draw and short circuit the power supply. The DTI unit provides dead-time insertion to deal with this problem.

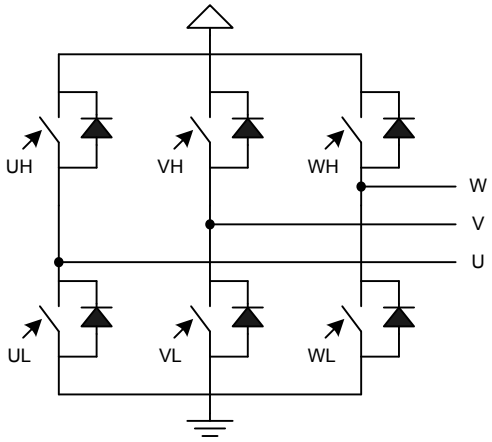


Figure 18.40. TIMER Triple Half-Bridge

For each of the 3 compare-match outputs of TIMER0, an additional complementary output is provided by the DTI unit. These outputs, named TIM0_CDTI0, TIM0_CDTI1 and TIM0_CDTI2 are provided to make control of e.g. 3-channel BLDC or permanent magnet AC (PMAC) motors possible using only a single timer, see Figure 18.41 TIMER Overview of Dead-Time Insertion Block for a Single PWM Channel on page 497.

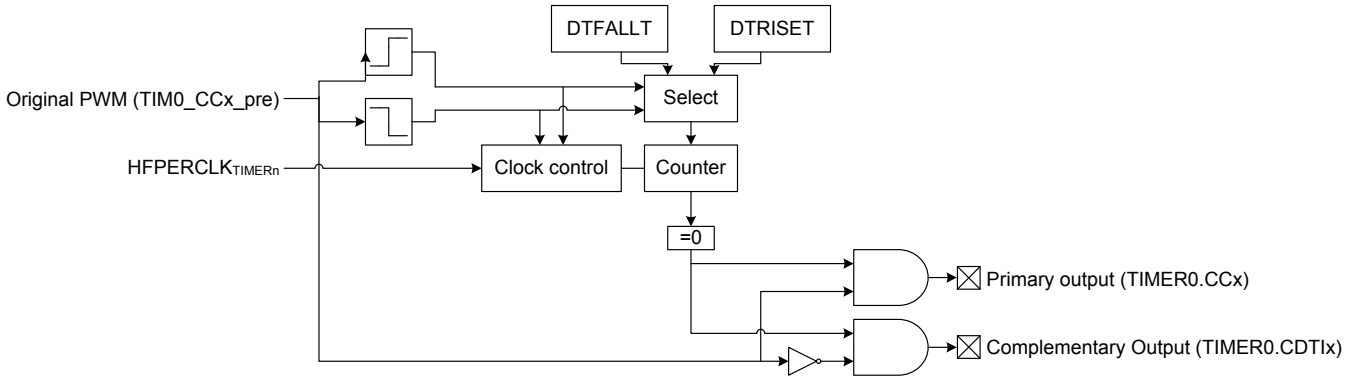


Figure 18.41. TIMER Overview of Dead-Time Insertion Block for a Single PWM Channel

The DTI unit is enabled by setting DTEN in TIMER0_DTCFG. In addition to providing the complementary outputs, the DTI unit then also overrides the compare match outputs from the timer.

The DTI unit gives the rising edges of the PWM outputs and the rising edges of the complementary PWM outputs a configurable time delay. By doing this, the DTI unit introduces a dead-time where both the primary and complementary outputs in a pair are inactive as seen in Figure 18.42 TIMER Polarity of Both Signals are Set as Active-High on page 497.

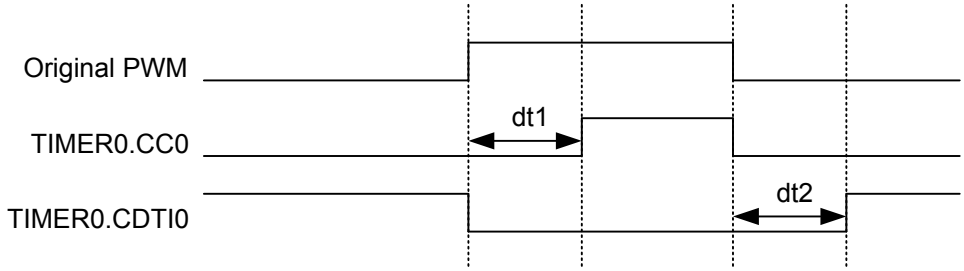


Figure 18.42. TIMER Polarity of Both Signals are Set as Active-High

Dead-time is specified individually for the rising and falling edge of the original PWM. These values are shared across all the three PWM channels of the DTI unit. A single prescaler value is provided for the DTI unit, meaning that both the rising and falling edge dead-times share prescaler value. The prescaler divides the $HFPERCLK_{TIMER0}$ by a configurable factor between 1 and 1024, which is set in the DTPRESC field in TIMER0_DTTIMECFG. The rising and falling edge dead-times are configured in DTRISET and DTFALLT in TIMER0_DTTIMECFG to any number between 1-64 $HFPERCLK_{TIMER0}$ cycles.

The DTAR and DTFATS bits in TIMER0_DTCFG control the DTI output behavior when the timer stops. By default the DTI block stops when the timer is stopped. Setting the DTAR bit will cause the DTI output on channel 0 to continue when the timer is stopped. DTAR effects only channel 0. See 18.3.4.2 PRS Channel as a Source for an example of when this can be used. While in this mode the undivided $HFPERCLK_{TIMER0}$ (DTPRESC=0) is always used regardless of the programmed DTPRESC value in TIMER0_DTTIMECFG. This means that rise and fall dead times are calculated assuming DTPRESC = 0.

When the timer stops, DTI outputs are frozen by default, preserving their last state. To allow the outputs to go to a safe state, program the DTFA field of the TIMER0_DTFCFG register to the safe values and set the DTFATS bitfield in the TIMER0_DTCFG register. Note that when DTAR is also set, DTAR has priority over DTFATS for DTI channel 0 output.

The following table shows the DTI output when the timer is halted.

Table 18.3. DTI Output When Timer Halted

DTAR	DTFATS	State
0	0	frozen
0	1	safe
1	0	running
1	1	running

18.3.4.1 Output Polarity

The value of the primary and complementary outputs in a pair will never be set active at the same time by the DTI unit. The polarity of the outputs can be changed if this is required by the application. The active values of the primary and complementary outputs are set by the DTIPOL and DTCINV bits in the TIMER0_DTCTRL register. The DTIPOL bit of this register specifies the base polarity. If DTIPOL = 0, then the outputs are active-high, and if DTIPOL = 1 they are active-low. The relative phase of the primary and complementary outputs is not changed by DTIPOL, as the polarity of both outputs is changed, see Figure 18.43 TIMER Output Polarities on page 498.

In some applications, it may be required that the primary outputs are active-high, while the complementary outputs are active-low. This can be accomplished by manipulating the DTCINV bit of the TIMER0_DTCTRL register, which inverts the polarity of the complementary outputs relative to the primary outputs. As an example, DTIPOL = 0 and DTCINV = 0 results in outputs with opposite phase and active-high states. Similarly, DTIPOL = 1 and DTCINV = 1 results in outputs with equal phase and the primary output will be active-high while the complementary will be active-low.

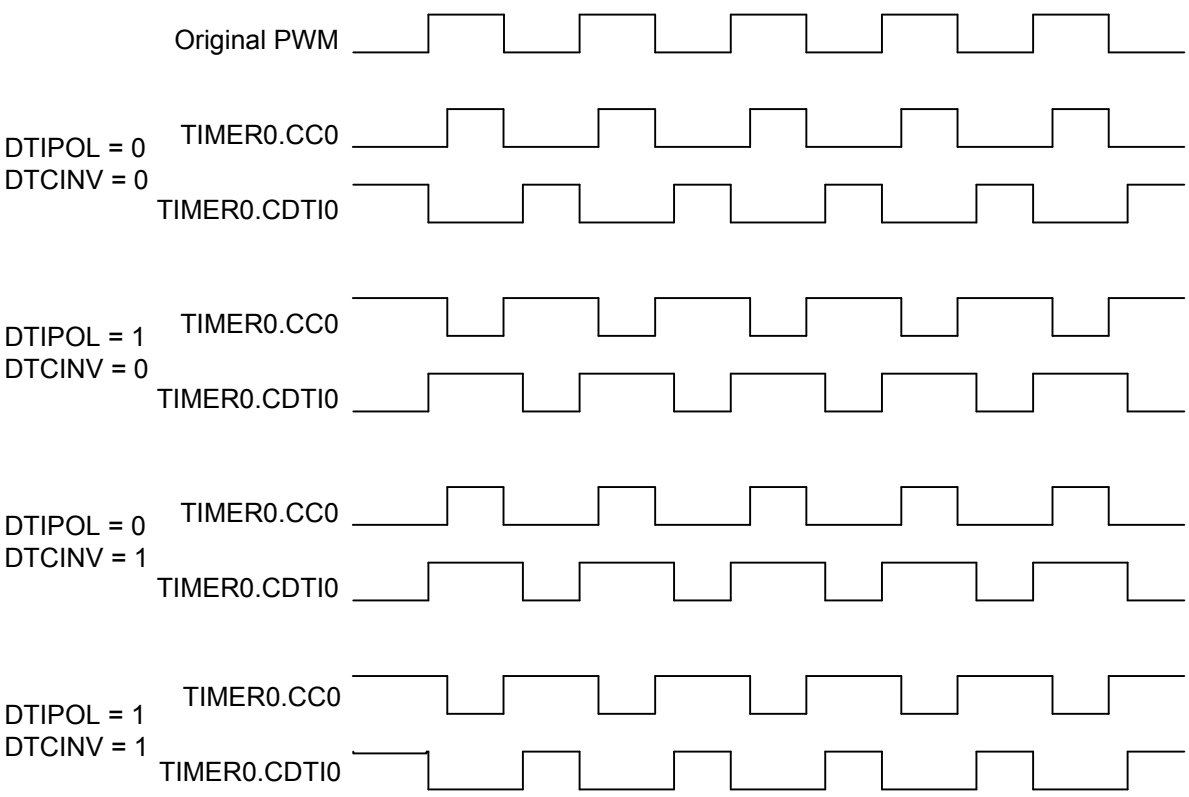


Figure 18.43. TIMER Output Polarities

Output generation on the individual DTI outputs can be disabled by configuring TIMER0_DTOGEN. When output generation on an output is disabled that output will go to and stay in its inactive state.

18.3.4.2 PRS Channel as a Source

A PRS channel can be used as input to the DTI module instead of the PWM output from the timer for DTI channel 0. Setting DTPRSEN in TIMER0_DTCFG will override the source of the first DTI channel, driving TIM0_CC0 and TIM0_CDTI0, with the value on the PRS channel. The rest of the DTI channels will continue to be driven by the PWM output from the timer. The input PRS channel is chosen within the PRS module with PRSSEL in the PRS_CONSUMER_TIMERn_DTI register. Note that the timer must be running even when PRS is used as the DTI source. However, if it is required to keep the DTI channel 0 running even when the timer is stopped, set DTAR in TIMER0_DTCFG. When this bit is set, it uses DTPRESC=0 regardless of the value programmed in DTPRESC in TIMER0_DTTIMECFG.

Note: When using synchronous PRS sources, it is recommended to configure the PRS consumer registers prior to selecting PRS triggering to avoid any false triggers.

The DTI prescaler, set by DTPRESC in TIMER0_DTTIMECFG determines the accuracy with which the DTI can insert dead-time into a PRS signal. The maximum dead-time error equals DTIPRESC+1 clock cycles. With DTIPRESC = 0, the inserted dead-times are therefore accurate, but they may be inaccurate for larger prescaler settings.

18.3.4.3 Fault Handling

The fault handling system of the DTI unit allows the outputs of the DTI unit to be put in a well-defined state in case of a fault. This hardware fault handling system enables a fast reaction to faults, reducing the possibility of damage to the system.

The fault sources which trigger a fault in the DTI module are determined by the bitfields of TIMER0_DTFCFG register. Any combination of the available error sources can be selected:

- PRS source 1, determined by PRSSEL in PRS_CONSUMER_TIMERn_DTIFS1
- PRS source 2, determined by PRSSEL in PRS_CONSUMER_TIMERn_DTIFS2
- Debugger
- Core Lockup
- EM2 or EM3 Entry

One or two PRS channels can be used as an error source. When PRS source 1 is selected as an error source, PRSSEL in PRS_CONSUMER_TIMERn_DTIFS1 determines which PRS channel is used for this source. PRSSEL in PRS_CONSUMER_TIMERn_DTIFS2 determines which PRS channel is selected as PRS source 2. Note that for Core Lockup, the LOCKUPRDIS in RMU_CTRL must be set. Otherwise this will generate a full reset of the chip.

Note: When using synchronous PRS sources, it is recommended to configure the PRS consumer registers prior to selecting PRS triggering to avoid any false triggers.

18.3.4.4 Action on Fault

When a fault occurs, the bit representing the fault source is set in TIMER0_DTFAULT register, and the outputs from the DTI unit are set to a well-defined state. The following options are available, and can be enabled by configuring DTFA in TIMER0_DTFCFG:

- Set outputs to inactive level
- Clear outputs
- Tristate outputs

With the first option enabled, the output state in case of a fault depends on the polarity settings for the individual outputs. An output set to be active high will be set low if a fault is detected, while an output set to be active low will be driven high.

When a fault occurs, the fault source(s) can be read out from TIMER0_DTFAULT register.

Additionally a fault action can also be triggered when the timer stops if DTFATS in TIMER0_DTCFG is set. This allows the DTI output to go to safe state specified by DTFA in TIMER0_DTFCFG when the timer stops. When DTAR and DTFATS in TIMER0_DTCFG are both set, DTI channel 0 keeps running even when the timer stops. This is useful when DTI channel 0 has an input coming from PRS.

18.3.4.5 Exiting Fault State

When a fault is triggered by the PRS system, software intervention is required to re-enable the outputs of the DTI unit. This is done by manually clearing bits in the TIMER0_DTFAULT register. If the fault source as determined by checking TIMER0_DTFAULT is the debugger alone, the outputs can be automatically restarted when the debugger exits. To enable automatic restart set DTDAS in TIMER0_DCTFG. When an automatic restart occurs the DTDBGF bit in TIMER0_DTFAULT will be automatically cleared by hardware. If any other bits in the TIMER0_DTFAULT register are set when the hardware clears DTDBGF the DTI module will not exit the fault state.

18.3.4.6 DTI Configuration Lock

To prevent software errors from making changes to the DTI configuration, a configuration lock is available. Writing any value but 0xCE80 to LOCKKEY in TIMER0_DTLOCK locks writes to registers TIMER0_DTCFG, TIMER0_DTFCFG, TIMER0_DTCTRL, and TIMER0_DTTIMECFG. To unlock the registers, write 0xCE80 to LOCKKEY in TIMER0_DTLOCK. The value of DTILOCKSTATUS in TIMERn_STATUS is 1 when the lock is active, and 0 when the registers are unlocked.

18.3.5 Debug Mode

When the CPU is halted in debug mode, the timer can be configured to either continue to run or to be frozen. This is configured in DEBUGRUN in TIMERn_CFG.

18.3.6 Interrupts, DMA and PRS Output

The timer can generate several type of output events:

- Counter Underflow
- Counter Overflow
- Quadrature Decoder Direction Change
- Compare match or input capture (one per compare/capture channel)

Each of the events has its own interrupt flag. Also, there are interrupt flags for each compare/capture channel which are set on FIFO overflow or underflow in capture mode. FIFO overflow happens when a new capture over-writes an old unread capture in TIMERn_CCx_ICF. FIFO underflow happens when software reads TIMERn_CCx_ICF while the FIFO is empty.

If the interrupt flags are set and the corresponding interrupt enable bits in TIMERn_IEN are set high, the timer will send out an interrupt request. Each of the events may optionally trigger signals to PRS channels. The PRSCONF field in TIMERn_CCx_CFG determines how PRS events are generated. When PRSCONF is set to PULSE, and event will lead to a one $HFPERCLK_{TIMERn}$ cycle high pulse on individual PRS outputs. Setting PRSCONF to LEVEL will make the PRS output follow the compare match output. Interrupts are cleared by setting the corresponding bit in the TIMERn_IFC register.

Each of the events will also set a DMA request when they occur. The different DMA requests are cleared when certain acknowledge conditions are met, see Table 18.4 TIMER DMA Events on page 500. Events which clear the DMA requests do not clear interrupt flags. Software must still manually clear the interrupt flag if interrupts are in use.

If DMACLRACT is set in TIMERn_CFG, the DMA request is cleared when the triggered DMA channel is active, without having to access any timer registers. This is useful in cases where a timer event is used to trigger a DMA transfer in output compare or PWM mode that does not target the OC or OCB registers. DMACLRACT is not applicable in input capture mode.

Table 18.4. TIMER DMA Events

Event	Acknowledge/Clear
Underflow/Overflow	Read or write to TIMERn_CNT or TIMERn_TOPB
CC0 Input Capture - ICFWLFULL0 flag set	ICFEMPTY0 flag set (read FIFO via TIMERn_CC0_ICF)
CC1 Input Capture - ICFWLFULL1 flag set	ICFEMPTY1 flag set (read FIFO via TIMERn_CC1_ICF)
CC2 Input Capture - ICFWLFULL2 flag set	ICFEMPTY2 flag set (read FIFO via TIMERn_CC2_ICF)
CC3 Input Capture - ICFWLFULL3 flag set	ICFEMPTY3 flag set (read FIFO via TIMERn_CC3_ICF)
CC0 Output Compare / PWM - Match event	Write TIMERn_CC0_OC or TIMERn_CC0_OCB
CC1 Output Compare / PWM - Match event	Write TIMERn_CC1_OC or TIMERn_CC1_OCB
CC2 Output Compare / PWM - Match event	Write TIMERn_CC2_OC or TIMERn_CC2_OCB
CC3 Output Compare / PWM - Match event	Write TIMERn_CC3_OC or TIMERn_CC3_OCB

18.3.7 GPIO Input/Output

The TIMn_CCx inputs/outputs and TIMn_CDTIx outputs are accessible as alternate functions through GPIO. Each pin connection can be enabled/disabled separately using the GPIO module control registers. See the device data sheet for the available locations for each signal.

18.4 TIMER Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	TIMER_IPVERSION	R	IP Version ID
0x004	TIMER_CFG	RW CONFIG	Configuration Register
0x008	TIMER_CTRL	RW SYNC	Control Register
0x00C	TIMER_CMD	W SYNC	Command Register
0x010	TIMER_STATUS	RH	Status Register
0x014	TIMER_IF	RWH INTFLAG	Interrupt Flag Register
0x018	TIMER_IEN	RW	Interrupt Enable Register
0x01C	TIMER_TOP	RWH SYNC	Counter Top Value Register
0x020	TIMER_TOPB	RW SYNC	Counter Top Value Buffer Register
0x024	TIMER_CNT	RWH SYNC	Counter Value Register
0x02C	TIMER_LOCK	W	TIMER Configuration Lock Register
0x030	TIMER_EN	RW ENABLE	Module En
0x060	TIMER_CCx_CFG	RW CONFIG	CC Channel Configuration Register
0x064	TIMER_CCx_CTRL	RW SYNC	CC Channel Control Register
0x068	TIMER_CCx_OC	RWH SYNC	OC Channel Value Register
0x070	TIMER_CCx_OCB	RW SYNC	OC Channel Value Buffer Register
0x074	TIMER_CCx_ICF	RH(r)	IC Channel Value Register
0x078	TIMER_CCx_ICOF	RH SYNC	IC Channel Value Overflow Register
0x0E0	TIMER_DTCFG	RW CONFIG	DTI Configuration Register
0x0E4	TIMER_DTTIMECFG	RW CONFIG	DTI Time Configuration Register
0x0E8	TIMER_DTFCFG	RW CONFIG	DTI Fault Configuration Register
0x0EC	TIMER_DTCTRL	RW SYNC	DTI Control Register
0x0F0	TIMER_DTOGEN	RW SYNC	DTI Output Generation Enable Register
0x0F4	TIMER_DTFAULT	RH	DTI Fault Register
0x0F8	TIMER_DTFAULTC	W SYNC	DTI Fault Clear Register
0x0FC	TIMER_DTLOCK	W	DTI Configuration Lock Register
0x1000	TIMER_IPVERSION_SET	R	IP Version ID
0x1004	TIMER_CFG_SET	RW CONFIG	Configuration Register
0x1008	TIMER_CTRL_SET	RW SYNC	Control Register
0x100C	TIMER_CMD_SET	W SYNC	Command Register
0x1010	TIMER_STATUS_SET	RH	Status Register
0x1014	TIMER_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1018	TIMER_IEN_SET	RW	Interrupt Enable Register
0x101C	TIMER_TOP_SET	RWH SYNC	Counter Top Value Register
0x1020	TIMER_TOPB_SET	RW SYNC	Counter Top Value Buffer Register

Offset	Name	Type	Description
0x1024	TIMER_CNT_SET	RWH SYNC	Counter Value Register
0x102C	TIMER_LOCK_SET	W	TIMER Configuration Lock Register
0x1030	TIMER_EN_SET	RW ENABLE	Module En
0x1060	TIMER_CCx_CFG_SET	RW CONFIG	CC Channel Configuration Register
0x1064	TIMER_CCx_CTRL_SET	RW SYNC	CC Channel Control Register
0x1068	TIMER_CCx_OC_SET	RWH SYNC	OC Channel Value Register
0x1070	TIMER_CCx_OCB_SET	RW SYNC	OC Channel Value Buffer Register
0x1074	TIMER_CCx_ICF_SET	RH(r)	IC Channel Value Register
0x1078	TIMER_CCx_ICOF_SET	RH SYNC	IC Channel Value Overflow Register
0x10E0	TIMER_DTCFG_SET	RW CONFIG	DTI Configuration Register
0x10E4	TIMER_DTTIMECFG_SET	RW CONFIG	DTI Time Configuration Register
0x10E8	TIMER_DTFCFG_SET	RW CONFIG	DTI Fault Configuration Register
0x10EC	TIMER_DTCTRL_SET	RW SYNC	DTI Control Register
0x10F0	TIMER_DTOGEN_SET	RW SYNC	DTI Output Generation Enable Register
0x10F4	TIMER_DTFAULT_SET	RH	DTI Fault Register
0x10F8	TIMER_DTFAULTC_SET	W SYNC	DTI Fault Clear Register
0x10FC	TIMER_DTLOCK_SET	W	DTI Configuration Lock Register
0x2000	TIMER_IPVERSION_CLR	R	IP Version ID
0x2004	TIMER_CFG_CLR	RW CONFIG	Configuration Register
0x2008	TIMER_CTRL_CLR	RW SYNC	Control Register
0x200C	TIMER_CMD_CLR	W SYNC	Command Register
0x2010	TIMER_STATUS_CLR	RH	Status Register
0x2014	TIMER_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2018	TIMER_IEN_CLR	RW	Interrupt Enable Register
0x201C	TIMER_TOP_CLR	RWH SYNC	Counter Top Value Register
0x2020	TIMER_TOPB_CLR	RW SYNC	Counter Top Value Buffer Register
0x2024	TIMER_CNT_CLR	RWH SYNC	Counter Value Register
0x202C	TIMER_LOCK_CLR	W	TIMER Configuration Lock Register
0x2030	TIMER_EN_CLR	RW ENABLE	Module En
0x2060	TIMER_CCx_CFG_CLR	RW CONFIG	CC Channel Configuration Register
0x2064	TIMER_CCx_CTRL_CLR	RW SYNC	CC Channel Control Register
0x2068	TIMER_CCx_OC_CLR	RWH SYNC	OC Channel Value Register
0x2070	TIMER_CCx_OCB_CLR	RW SYNC	OC Channel Value Buffer Register
0x2074	TIMER_CCx_ICF_CLR	RH(r)	IC Channel Value Register
0x2078	TIMER_CCx_ICOF_CLR	RH SYNC	IC Channel Value Overflow Register
0x20E0	TIMER_DTCFG_CLR	RW CONFIG	DTI Configuration Register
0x20E4	TIMER_DTTIMECFG_CLR	RW CONFIG	DTI Time Configuration Register

Offset	Name	Type	Description
0x20E8	TIMER_DTFCFG_CLR	RW CONFIG	DTI Fault Configuration Register
0x20EC	TIMER_DTCTRL_CLR	RW SYNC	DTI Control Register
0x20F0	TIMER_DTOGEN_CLR	RW SYNC	DTI Output Generation Enable Register
0x20F4	TIMER_DTFAULT_CLR	RH	DTI Fault Register
0x20F8	TIMER_DTFAULTC_CLR	W SYNC	DTI Fault Clear Register
0x20FC	TIMER_DTLOCK_CLR	W	DTI Configuration Lock Register
0x3000	TIMER_IPVERSION_TGL	R	IP Version ID
0x3004	TIMER_CFG_TGL	RW CONFIG	Configuration Register
0x3008	TIMER_CTRL_TGL	RW SYNC	Control Register
0x300C	TIMER_CMD_TGL	W SYNC	Command Register
0x3010	TIMER_STATUS_TGL	RH	Status Register
0x3014	TIMER_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3018	TIMER_IEN_TGL	RW	Interrupt Enable Register
0x301C	TIMER_TOP_TGL	RWH SYNC	Counter Top Value Register
0x3020	TIMER_TOPB_TGL	RW SYNC	Counter Top Value Buffer Register
0x3024	TIMER_CNT_TGL	RWH SYNC	Counter Value Register
0x302C	TIMER_LOCK_TGL	W	TIMER Configuration Lock Register
0x3030	TIMER_EN_TGL	RW ENABLE	Module En
0x3060	TIMER_CCx_CFG_TGL	RW CONFIG	CC Channel Configuration Register
0x3064	TIMER_CCx_CTRL_TGL	RW SYNC	CC Channel Control Register
0x3068	TIMER_CCx_OC_TGL	RWH SYNC	OC Channel Value Register
0x3070	TIMER_CCx_OCB_TGL	RW SYNC	OC Channel Value Buffer Register
0x3074	TIMER_CCx_ICF_TGL	RH(r)	IC Channel Value Register
0x3078	TIMER_CCx_ICOF_TGL	RH SYNC	IC Channel Value Overflow Register
0x30E0	TIMER_DTCFG_TGL	RW CONFIG	DTI Configuration Register
0x30E4	TIMER_DTTIMECFG_TGL	RW CONFIG	DTI Time Configuration Register
0x30E8	TIMER_DTFCFG_TGL	RW CONFIG	DTI Fault Configuration Register
0x30EC	TIMER_DTCTRL_TGL	RW SYNC	DTI Control Register
0x30F0	TIMER_DTOGEN_TGL	RW SYNC	DTI Output Generation Enable Register
0x30F4	TIMER_DTFAULT_TGL	RH	DTI Fault Register
0x30F8	TIMER_DTFAULTC_TGL	W SYNC	DTI Fault Clear Register
0x30FC	TIMER_DTLOCK_TGL	W	DTI Configuration Lock Register

18.5 TIMER Register Description

18.5.1 TIMER_IPVERSION - IP Version ID

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP Version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

18.5.2 TIMER_CFG - Configuration Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0										0x0	0x0					0x0	0x0	0x0		0x0	0x0	0x0	0x0	0x0		0x0	
Access					RW										RW	RW					RW	RW	RW		RW	RW	RW	RW	RW		RW	
Name					PRESC										RSSCOIST	ATI					DISSYNCOUT	RETIMEEN	CLKSEL		DMACLRACT	DEBUGRUN	QDM	OSMEN	SYNC		MODE	

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:18	PRESC	0x0	RW	**Prescaler Setting**

These bits select the prescaling factor for the counter clock. The selected timer clock will be divided by PRESC+1 before clocking the counter. The following modes are provided for easier software porting from Series 0 or Series 1 devices. However, the prescaler is not limited to these options.

Value	Mode	Description
0	DIV1	No prescaling
1	DIV2	Prescale by 2
3	DIV4	Prescale by 4
7	DIV8	Prescale by 8
15	DIV16	Prescale by 16
31	DIV32	Prescale by 32
63	DIV64	Prescale by 64
127	DIV128	Prescale by 128
255	DIV256	Prescale by 256
511	DIV512	Prescale by 512
1023	DIV1024	Prescale by 1024

17	RSSCOIST	0x0	RW	**Reload-Start Sets COIST**

When enabled, compare output is set to COIST value on a Reload-Start event.

16	ATI	0x0	RW	**Always Track Inputs**

Enabling ATI makes CCPOL always track the polarity of the inputs.

15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	DISSYNCOUT	0x0	RW	**Disable Timer Start/Stop/Reload output**

When this bit is set, the Timer does not start/stop/reload other timers with SYNC bit set.

Value	Mode	Description
0	EN	Timer can start/stop/reload other timers with SYNC bit set

Bit	Name	Reset	Access	Description
	1	DIS		Timer cannot start/stop/reload other timers with SYNC bit set
10	RETIMEEN	0x0	RW	**PWM output retimed enable**
	Enable retiming of PWM output.			
	Value	Mode		Description
	0	DISABLE		PWM outputs are not re-timed.
	1	ENABLE		PWM outputs are re-timed.
9:8	CLKSEL	0x0	RW	**Clock Source Select**
	These bits select the clock source for the timer.			
	Value	Mode		Description
	0	PRESCEM01GRPACLK		Prescaled EM01GRPACLK
	1	CC1		Compare/Capture Channel 1 Input
	2	TIMEROUF		Timer is clocked by underflow(down-count) or overflow(up-count) in the lower numbered neighbor Timer
7	DMACLRACT	0x0	RW	**DMA Request Clear on Active**
	When this bit is set, the DMA requests are cleared when the corresponding DMA channel is active. This enables the timer DMA requests to be cleared without accessing the timer.			
6	DEBUGRUN	0x0	RW	**Debug Mode Run Enable**
	Set this bit to enable timer to run in debug mode.			
	Value	Mode		Description
	0	HALT		Timer is halted in debug mode
	1	RUN		Timer is running in debug mode
5	QDM	0x0	RW	**Quadrature Decoder Mode Selection**
	This bit sets the mode for the quadrature decoder.			
	Value	Mode		Description
	0	X2		X2 mode selected
	1	X4		X4 mode selected
4	OSMEN	0x0	RW	**One-shot Mode Enable**
	Enable/disable one shot mode.			
3	SYNC	0x0	RW	**Timer Start/Stop/Reload Synchronization**
	When this bit is set, the Timer is started/stopped/reloaded by start/stop/reload commands in the other timers.			
	Value	Mode		Description
	0	DISABLE		Timer operation is unaffected by other timers.
	1	ENABLE		Timer may be started, stopped and re-loaded from other timer instances.

Bit	Name	Reset	Access	Description
2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	MODE	0x0	RW	**Timer Mode**

These bits set the counting mode for the Timer. Note, when Quadrature Decoder Mode is selected (MODE = 'b11), the CLKSEL is don't care. The Timer is clocked by the Decoder Mode clock output.

Value	Mode	Description
0	UP	Up-count mode
1	DOWN	Down-count mode
2	UPDOWN	Up/down-count mode
3	QDEC	Quadrature decoder mode

18.5.3 TIMER_CTRL - Control Register

Offset	Bit Position																															
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0		0x0	
Access																												RW	RW		RW	
Name																												X2CNT	FALLA		RISEA	

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	X2CNT	0x0	RW	**2x Count Mode**
	Enable 2x count mode			
3:2	FALLA	0x0	RW	**Timer Falling Input Edge Action**
	These bits select the action taken in the counter when a falling edge occurs on the input.			

Value	Mode	Description
0	NONE	No action
1	START	Start counter without reload
2	STOP	Stop counter without reload
3	RELOADSTART	Reload and start counter

Bit	Name	Reset	Access	Description
1:0	RISEA	0x0	RW	**Timer Rising Input Edge Action**
	These bits select the action taken in the counter when a rising edge occurs on the input.			

Value	Mode	Description
0	NONE	No action
1	START	Start counter without reload
2	STOP	Stop counter without reload
3	RELOADSTART	Reload and start counter

18.5.4 TIMER_CMD - Command Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															W(nB)	W(nB)
Name																															STOP	START

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	STOP	0x0	W(nB)	**Stop Timer**
	Write a 1 to this bit to stop timer			
0	START	0x0	W(nB)	**Start Timer**
	Write a 1 to this bit to start timer			

18.5.5 TIMER_STATUS - Status Register

Offset	Bit Position																														
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset						0x0	0x0	0x0						0x0	0x0	0x0						0x0	0x0	0x0		0x0	0x0	0x0		0x0	0x0	0x0
Access						R	R	R						R	R	R						R	R	R		R	R	R		R	R	R
Name						CCPOL2	CCPOL1	CCPOL0						ICFEMPTY2	ICFEMPTY1	ICFEMPTY0						OCBV2	OCBV1	OCBV0		SYNCBUSY	DTILOCKSTATUS	TIMERLOCKSTATUS		TOPBV	DIR	RUNNING

Bit	Name	Reset	Access	Description
31:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26	CCPOL2	0x0	R	**Compare/Capture Polarity**
	In Input Capture mode, this bit indicates the polarity of the edge that triggered capture in TIMERn_CCx_CCV. In Compare/PWM mode, this bit indicates the polarity of the selected input to CC channel x. These bits are cleared when CCMODE is written to 0b00 (Off).			
	Value	Mode		Description
	0	LOWRISE		CCx polarity low level/rising edge
	1	HIGHFALL		CCx polarity high level/falling edge
25	CCPOL1	0x0	R	**Compare/Capture Polarity**
	In Input Capture mode, this bit indicates the polarity of the edge that triggered capture in TIMERn_CCx_CCV. In Compare/PWM mode, this bit indicates the polarity of the selected input to CC channel x. These bits are cleared when CCMODE is written to 0b00 (Off).			
	Value	Mode		Description
	0	LOWRISE		CCx polarity low level/rising edge
	1	HIGHFALL		CCx polarity high level/falling edge
24	CCPOL0	0x0	R	**Compare/Capture Polarity**
	In Input Capture mode, this bit indicates the polarity of the edge that triggered capture in TIMERn_CCx_CCV. In Compare/PWM mode, this bit indicates the polarity of the selected input to CC channel x. These bits are cleared when CCMODE is written to 0b00 (Off).			
	Value	Mode		Description
	0	LOWRISE		CCx polarity low level/rising edge
	1	HIGHFALL		CCx polarity high level/falling edge
23:19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18	ICFEMPTY2	0x0	R	**Input capture fifo empty**
	Set when input capture FIFO is empty			

Bit	Name	Reset	Access	Description
17	ICFEMPTY1	0x0	R	**Input capture fifo empty**
	Set when input capture FIFO is empty			
16	ICFEMPTY0	0x0	R	**Input capture fifo empty**
	Set when input capture FIFO is empty			
15:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	OCBV2	0x0	R	**Output Compare Buffer Valid**
	This field indicates that the TIMERn_CCx_CCVB registers contain data which have not been written to TIMERn_CCx_CCV. These bits are only used in OUTPUTCOMPARE or PWM mode and are cleared when CCMODE is written to 0b00 (Off).			
9	OCBV1	0x0	R	**Output Compare Buffer Valid**
	This field indicates that the TIMERn_CCx_CCVB registers contain data which have not been written to TIMERn_CCx_CCV. These bits are only used in OUTPUTCOMPARE or PWM mode and are cleared when CCMODE is written to 0b00 (Off).			
8	OCBV0	0x0	R	**Output Compare Buffer Valid**
	This field indicates that the TIMERn_CCx_CCVB registers contain data which have not been written to TIMERn_CCx_CCV. These bits are only used in OUTPUTCOMPARE or PWM mode and are cleared when CCMODE is written to 0b00 (Off).			
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	SYNCBUSY	0x0	R	**Sync Busy**
	Indicates synchronization ongoing			
5	DTILOCKSTATUS	0x0	R	**DTI lock status**
	Indicates current status of DTI lock			

Value	Mode	Description
0	UNLOCKED	DTI registers are unlocked
1	LOCKED	DTI registers are locked

4	TIMERLOCKSTATUS	0x0	R	**Timer lock status**
	Indicates current status of Timer lock			

Value	Mode	Description
0	UNLOCKED	TIMER registers are unlocked
1	LOCKED	TIMER registers are locked

3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	TOPBV	0x0	R	**TOP Buffer Valid**
	This indicates that TIMERn_TOPB contains valid data that has not been written to TIMERn_TOP. This bit is also cleared when TIMERn_TOP is written.			
1	DIR	0x0	R	**Direction**
	Indicates count direction.			

Value	Mode	Description

Bit	Name	Reset	Access	Description
	0	UP		Counting up
	1	DOWN		Counting down
0	RUNNING	0x0	R	**Running**
	Indicates if timer is running or not.			

18.5.6 TIMER_IF - Interrupt Flag Register

Offset	Bit Position																														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset						0x0	0x0	0x0		0x0	0x0	0x0		0x0	0x0	0x0										0x0	0x0	0x0		0x0	0x0	0x0
Access						RW	RW	RW		RW	RW	RW		RW	RW	RW										RW	RW	RW		RW	RW	RW
Name						ICFUF2	ICFUF1	ICFUF0		ICFOF2	ICFOF1	ICFOF0		ICFWLFULL2	ICFWLFULL1	ICFWLFULL0										CC2	CC1	CC0		DIRCHG	UF	OF

Bit	Name	Reset	Access	Description
31:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26	ICFUF2	0x0	RW	**Input capture FIFO underflow**
	Indicates that software tried to read an empty FIFO on channel 2.			
25	ICFUF1	0x0	RW	**Input capture FIFO underflow**
	Indicates that software tried to read an empty FIFO on channel 1.			
24	ICFUF0	0x0	RW	**Input capture FIFO underflow**
	Indicates that software tried to read an empty FIFO on channel 0.			
23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22	ICFOF2	0x0	RW	**Input Capture FIFO overflow**
	Indicates that input capture FIFO for channel 2 has overflown, and a prior captured value was lost. The latest captured value can be read from the ICOF register.			
21	ICFOF1	0x0	RW	**Input Capture FIFO overflow**
	Indicates that input capture FIFO for channel 1 has overflown, and a prior captured value was lost. The latest captured value can be read from the ICOF register.			
20	ICFOF0	0x0	RW	**Input Capture FIFO overflow**
	Indicates that input capture FIFO for channel 0 has overflown, and a prior captured value was lost. The latest captured value can be read from the ICOF register.			
19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18	ICFWLFULL2	0x0	RW	**Input Capture Watermark Level Full**
	This bit indicates that the Input capture FIFO watermark for channel 2 has been exceeded.			
17	ICFWLFULL1	0x0	RW	**Input Capture Watermark Level Full**
	This bit indicates that the Input capture FIFO watermark for channel 1 has been exceeded.			
16	ICFWLFULL0	0x0	RW	**Input Capture Watermark Level Full**
	This bit indicates that the Input capture FIFO watermark for channel 0 has been exceeded.			
15:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	CC2	0x0	RW	**Capture Compare Channel 2 Interrupt Flag**

Bit	Name	Reset	Access	Description
	In INPUT CAPTURE mode this bit indicates that a new Capture event has taken place. In OUTPUTCOMPARE or PWM mode this bit indicates that a match event has taken place			
5	CC1	0x0	RW	**Capture Compare Channel 1 Interrupt Flag**
	In INPUT CAPTURE mode this bit indicates that a new Capture event has taken place. In OUTPUTCOMPARE or PWM mode this bit indicates that a match event has taken place			
4	CC0	0x0	RW	**Capture Compare Channel 0 Interrupt Flag**
	In INPUT CAPTURE mode this bit indicates that a new Capture event has taken place. In OUTPUTCOMPARE or PWM mode this bit indicates that a match event has taken place			
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	DIRCHG	0x0	RW	**Direction Change Detect Interrupt Flag**
	This bit is set when count direction changes. Set only in Quadrature Decoder mode			
1	UF	0x0	RW	**Underflow Interrupt Flag**
	This bit indicates that there has been an underflow.			
0	OF	0x0	RW	**Overflow Interrupt Flag**
	This bit indicates that there has been an overflow.			

18.5.7 TIMER_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset						0x0	0x0	0x0		0x0	0x0	0x0		0x0	0x0	0x0										0x0	0x0	0x0		0x0	0x0	0x0
Access						RW	RW	RW		RW	RW	RW		RW	RW	RW										RW	RW	RW		RW	RW	RW
Name						ICFUF2	ICFUF1	ICFUF0		ICFOF2	ICFOF1	ICFOF0		ICFWLFULL2	ICFWLFULL1	ICFWLFULL0										CC2	CC1	CC0		DIRCHG	UF	OF

Bit	Name	Reset	Access	Description
31:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26	ICFUF2	0x0	RW	**ICFUF2 Interrupt Enable**
	Enable/Disable the ICFUF2 interrupt			
25	ICFUF1	0x0	RW	**ICFUF1 Interrupt Enable**
	Enable/Disable the ICFUF1 interrupt			
24	ICFUF0	0x0	RW	**ICFUF0 Interrupt Enable**
	Enable/Disable the ICFUF0 interrupt			
23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22	ICFOF2	0x0	RW	**ICFOF2 Interrupt Enable**
	Enable/Disable the ICFOF2 interrupt			
21	ICFOF1	0x0	RW	**ICFOF1 Interrupt Enable**
	Enable/Disable the ICFOF1 interrupt			
20	ICFOF0	0x0	RW	**ICFOF0 Interrupt Enable**
	Enable/Disable the ICFOF0 interrupt			
19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18	ICFWLFULL2	0x0	RW	**ICFWLFULL2 Interrupt Enable**
	Enable/Disable the ICFWLFULL2 interrupt			
17	ICFWLFULL1	0x0	RW	**ICFWLFULL1 Interrupt Enable**
	Enable/Disable the ICFWLFULL1 interrupt			
16	ICFWLFULL0	0x0	RW	**ICFWLFULL0 Interrupt Enable**
	Enable/Disable the ICFWLFULL0 interrupt			
15:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	CC2	0x0	RW	**CC2 Interrupt Enable**
	Enable/Disable the CC2 interrupt			
5	CC1	0x0	RW	**CC1 Interrupt Enable**

Bit	Name	Reset	Access	Description
	Enable/Disable the CC1 interrupt			
4	CC0	0x0	RW	**CC0 Interrupt Enable**
	Enable/Disable the CC0 interrupt			
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	DIRCHG	0x0	RW	**Direction Change Detect Interrupt Enable**
	Enable/Disable the DIRCHG interrupt			
1	UF	0x0	RW	**Underflow Interrupt Enable**
	Enable/Disable the UF interrupt			
0	OF	0x0	RW	**Overflow Interrupt Enable**
	Enable/Disable the OF interrupt			

18.5.8 TIMER_TOP - Counter Top Value Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFF																															
Access	RW																															
Name	TOP																															

Bit	Name	Reset	Access	Description
31:0	TOP	0xFFFF	RW	**Counter Top Value**
	These bits hold the TOP value for the counter			

18.5.9 TIMER_TOPB - Counter Top Value Buffer Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	TOPB																															

Bit	Name	Reset	Access	Description
31:0	TOPB	0x0	RW	**Counter Top Buffer Register**
	These bits hold the TOP buffer value.			

18.5.10 TIMER_CNT - Counter Value Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	CNT																															

Bit	Name	Reset	Access	Description
31:0	CNT	0x0	RW	**Counter Value**
	These bits hold the counter value.			

18.5.11 TIMER_LOCK - TIMER Configuration Lock Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x0	W	**Timer Lock Key**
	Write any other value than the unlock code to lock TIMERn_CTRL, TIMERn_CFG, TIMERn_CMD, TIMERn_TOP, TIMERn_CNT, TIMERn_CCx_CTRL, TIMERn_CCx_CFG, and TIMERn_CCx_OC from editing. Write the unlock code to unlock these registers.			

Value	Mode	Description
52864	UNLOCK	Write to unlock TIMER registers

18.5.12 TIMER_EN - Module En

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status** When EN is cleared, DISABLING status is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs and FIFO
0	EN	0x0	RW	**Timer Module Enable** The ENABLE bit enables the module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit.

18.5.13 TIMER_CCx_CFG - CC Channel Configuration Register

Offset	Bit Position																															
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0	0x0	0x0	0x0														0x0			0x0	
Access											RW	RW	RW	RW														RW			RW	
Name											ICFWL	FILT	PRSCONF	INSEL														COIST			MODE	

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	ICFWL	0x0	RW	**Input Capture FIFO watermark level**
	Sets the watermark level for generation of the ICFWLFULL interrupt and DMA requests. ICFWLFULL will be set and DMA requests may be generated if the number of free FIFO entries is less than or equal to ICFWL.			
20	FILT	0x0	RW	**Digital Filter**
	Enable digital filter.			

Value	Mode	Description
0	DISABLE	Digital Filter Disabled
1	ENABLE	Digital Filter Enabled

Bit	Name	Reset	Access	Description
19	PRSCONF	0x0	RW	**PRS Configuration**
	Select PRS pulse or level for PRS output.			

Value	Mode	Description
0	PULSE	Each CC event will generate a one EM01GRPACLK cycle high pulse
1	LEVEL	The PRS channel will follow CC out

Bit	Name	Reset	Access	Description
18:17	INSEL	0x0	RW	**Input Selection**
	Select Compare/Capture channel input.			

Value	Mode	Description
0	PIN	TIMERnCCx pin is selected
1	PRSSYNC	Synchornous PRS selected
2	PRSASYNCLEVEL	Asynchronous Level PRS selected
3	PRSASYNCPULSE	Asynchronous Pulse PRS selected

Bit	Name	Reset	Access	Description
16:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	COIST	0x0	RW	**Compare Output Initial State**

Bit	Name	Reset	Access	Description
	This bit is only used in Output Compare and PWM mode. When this bit is set in Compare or PWM mode,the output is set high when the counter is disabled. When counting resumes, this value will represent the initial value for the output. If the bit is cleared, the output will be cleared when the counter is disabled.			
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	MODE	0x0	RW	**CC Channel Mode**

These bits select the mode for Compare/Capture channel.

Value	Mode	Description
0	OFF	Compare/Capture channel turned off
1	INPUTCAPTURE	Input Capture
2	OUTPUTCOMPARE	Output Compare
3	PWM	Pulse-Width Modulation

18.5.14 TIMER_CCx_CTRL - CC Channel Control Register

Offset	Bit Position																													
0x064	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0		0x0												0x0		0x0		0x0							0x0		
Access					RW		RW												RW		RW		RW							RW		
Name					ICEVCTRL		ICEDGE												CUFOA		COFOA		CMOA							OUTINV		

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:26	ICEVCTRL	0x0	RW	**Input Capture Event Control**

These bits control when a Compare/Capture PRS output pulse and interrupt flag is set. DMA request however is set on every capture.

Value	Mode	Description
0	EVERYEDGE	PRS output pulse and interrupt flag set on every capture
1	EVERYSECONDEDGE	PRS output pulse and interrupt flag set on every second capture
2	RISING	PRS output pulse and interrupt flag set on rising edge only (if ICEDGE = BOTH)
3	FALLING	PRS output pulse and interrupt flag set on falling edge only (if ICEDGE = BOTH)

25:24	ICEDGE	0x0	RW	**Input Capture Edge Select**

These bits control which edges the edge detector triggers on. The output is used for input capture and external clock input.

Value	Mode	Description
0	RISING	Rising edges detected
1	FALLING	Falling edges detected
2	BOTH	Both edges detected
3	NONE	No edge detection, signal is left as it is

23:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:12	CUFOA	0x0	RW	**Counter Underflow Output Action**

Select output action on counter underflow.

Value	Mode	Description
0	NONE	No action on counter underflow
1	TOGGLE	Toggle output on counter underflow
2	CLEAR	Clear output on counter underflow
3	SET	Set output on counter underflow

Bit	Name	Reset	Access	Description
11:10	COFOA	0x0	RW	**Counter Overflow Output Action**
	Select output action on counter overflow.			

Value	Mode	Description
0	NONE	No action on counter overflow
1	TOGGLE	Toggle output on counter overflow
2	CLEAR	Clear output on counter overflow
3	SET	Set output on counter overflow

9:8	CMOA	0x0	RW	**Compare Match Output Action**
	Select output action on compare match.			

Value	Mode	Description
0	NONE	No action on compare match
1	TOGGLE	Toggle output on compare match
2	CLEAR	Clear output on compare match
3	SET	Set output on compare match

7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	OUTINV	0x0	RW	**Output Invert**
	Setting this bit inverts the output from the CC channel (Output compare or PWM mode).			
1:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

18.5.15 TIMER_CCx_OC - OC Channel Value Register

Offset	Bit Position																														
0x068	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	OC																															

Bit	Name	Reset	Access	Description
31:0	OC	0x0	RW	**Output Compare Value**
	This fields holds the output compare value			

18.5.16 TIMER_CCx_OCB - OC Channel Value Buffer Register

Offset	Bit Position																													
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	OCB																															

Bit	Name	Reset	Access	Description
31:0	OCB	0x0	RW	**Output Compare Value Buffer**
	This field holds the Output Compare buffer value which will be written to TIMERn_CCx_OC on an update event if TIMERn_CCx_OCB contains valid data			

18.5.17 TIMER_CCx_ICF - IC Channel Value Register

Offset	Bit Position																													
0x074	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R(r)																															
Name	ICF																															

Bit	Name	Reset	Access	Description
31:0	ICF	0x0	R(r)	**Input Capture FIFO**
	This FIFO holds captured values in input capture mode. Reading this register will pop the oldest unread value from the FIFO.			

18.5.18 TIMER_CCx_ICOF - IC Channel Value Overflow Register

Offset	Bit Position																													
0x078	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	ICOF																															

Bit	Name	Reset	Access	Description
31:0	ICOF	0x0	R	**Input Capture FIFO Overflow**
	This register always contains the most recent input capture value. If the input capture FIFO is full and a new capture occurs, this register will be updated and the previous caputre value is over-written.			

18.5.19 TIMER_DTCFG - DTI Configuration Register

Offset	Bit Position																													
0x0E0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0	0x0	0x0								0x0	0x0
Access																					RW	RW	RW								RW	RW
Name																					DTPRSEN	DTFATS	DTAR								DTDAS	DTEN

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	DTPRSEN	0x0	RW	**DTI PRS Source Enable**
	Enable/disable PRS as DTI input.			
10	DTFATS	0x0	RW	**DTI Fault Action on Timer Stop**
	When Timer stops, DTI block outputs go to safe state as programmed in DTFA field of TIMERn_DTFC register. However, when DTAR is also set,DTAR having higher priority allows channel0 to output the incoming PRS input while the other channels go to safe state			
9	DTAR	0x0	RW	**DTI Always Run**
	This is used only for DTI channel 0. It Allows DTI channel 0 to keep running even when the timer is stopped. This is useful when its input source is PRS. However, here the undivided peripheral clock is always used regardless of the programmed value in DTPRESC.			
8:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DTDAS	0x0	RW	**DTI Automatic Start-up Functionality**
	Configure DTI restart on debugger exit.			

Value	Mode	Description
0	NORESTART	No DTI restart on debugger exit
1	RESTART	DTI restart on debugger exit

0	DTEN	0x0	RW	**DTI Enable**
	Enable/disable DTI.			

18.5.20 TIMER_DTTIMECFG - DTI Time Configuration Register

Offset	Bit Position																															
0x0E4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0						0x0						0x0									
Access											RW						RW						RW									
Name											DTFALLT						DTRISET						DTPRESC									

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:16	DTFALLT	0x0	RW	**DTI Fall-time**
	Set time span for the falling edge. The fall time is DTFALLT+1 prescaled peripheral clock cycles			
15:10	DTRISET	0x0	RW	**DTI Rise-time**
	Set time span for the rising edge. The rise time is DTRISET+1 prescaled peripheral clock cycles			
9:0	DTPRESC	0x0	RW	**DTI Prescaler Setting**
	These bits select the prescaling factor for DTI. The selected timer clock will be divided by DTPRESC+1 before clocking the DTI logic.			

18.5.21 TIMER_DTFCFG - DTI Fault Configuration Register

Offset	Bit Position																															
0x0E8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x0	0x0	0x0	0x0	0x0							0x0																	
Access				RW	RW	RW	RW	RW							RW																	
Name				DTEM23FEN	DTLOCKUPFEN	DTDBGFEN	DTPRS1FEN	DTPRS0FEN							DTFA																	

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28	DTEM23FEN	0x0	RW	**DTI EM23 Fault Enable**
	Set this bit to 1 to enable EM2 or EM3 entry as a fault source			
27	DTLOCKUPFEN	0x0	RW	**DTI Lockup Fault Enable**
	Set this bit to 1 to enable core lockup as a fault source			
26	DTDBGFEN	0x0	RW	**DTI Debugger Fault Enable**
	Set this bit to 1 to enable debugger as a fault source			
25	DTPRS1FEN	0x0	RW	**DTI PRS 1 Fault Enable**
	Set this bit to 1 to enable PRS source 1 as a fault source			
24	DTPRS0FEN	0x0	RW	**DTI PRS 0 Fault Enable**
	Set this bit to 1 to enable PRS source 0 as a fault source			
23:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17:16	DTFA	0x0	RW	**DTI Fault Action**
	Select fault action.			

Value	Mode	Description
0	NONE	No action on fault
1	INACTIVE	Set outputs inactive
2	CLEAR	Clear outputs
3	TRISTATE	Tristate outputs

15:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

18.5.22 TIMER_DTCTRL - DTI Control Register

Offset	Bit Position																													
0x0EC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															DTIPOL	DTCINV

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DTIPOL	0x0	RW	**DTI Inactive Polarity**
	Set inactive polarity of outputs			
0	DTCINV	0x0	RW	**DTI Complementary Output Invert.**
	DTI Complementary Output Invert.			

18.5.23 TIMER_DTOGEN - DTI Output Generation Enable Register

Offset	Bit Position																													
0x0F0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											RW	RW	RW	RW	RW	RW
Name																											DTOGCDTI2EN	DTOGCDTI1EN	DTOGCDTI0EN	DTOGCC2EN	DTOGCC1EN	DTOGCC0EN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	DTOGCDTI2EN	0x0	RW	**DTI CDTIn Output Generation Enable**
	This bit enables/disables output generation for the CDTI output from the DTI.			
4	DTOGCDTI1EN	0x0	RW	**DTI CDTIn Output Generation Enable**
	This bit enables/disables output generation for the CDTI output from the DTI.			
3	DTOGCDTI0EN	0x0	RW	**DTI CDTIn Output Generation Enable**
	This bit enables/disables output generation for the CDTI output from the DTI.			
2	DTOGCC2EN	0x0	RW	**DTI CCn Output Generation Enable**
	This bit enables/disables output generation for the CC output from the DTI.			
1	DTOGCC1EN	0x0	RW	**DTI CCn Output Generation Enable**
	This bit enables/disables output generation for the CC output from the DTI.			
0	DTOGCC0EN	0x0	RW	**DTI CCn Output Generation Enable**
	This bit enables/disables output generation for the CC output from the DTI.			

18.5.24 TIMER_DTFAULT - DTI Fault Register

Offset	Bit Position																													
0x0F4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												R	R	R	R	R
Name																												DTEM23F	DTLOCKUPF	DTDBGF	DTPRS1F	DTPRS0F

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	DTEM23F	0x0	R	**DTI EM23 Entry Fault**
	This bit is set to 1 if EM2 or EM3 entry has occurred and DTEM23FEN is set to 1. The TIMER0_DTFAULTC register can be used to clear fault bits.			
3	DTLOCKUPF	0x0	R	**DTI Lockup Fault**
	This bit is set to 1 if a core lockup fault has occurred and DTLOCKUPFEN is set to 1. The TIMER0_DTFAULTC register can be used to clear fault bits.			
2	DTDBGF	0x0	R	**DTI Debugger Fault**
	This bit is set to 1 if a debugger fault has occurred and DTDBGFEN is set to 1. The TIMER0_DTFAULTC register can be used to clear fault bits.			
1	DTPRS1F	0x0	R	**DTI PRS 1 Fault**
	This bit is set to 1 if a PRS 1 fault has occurred and DTPRS1FEN is set to 1. The TIMER0_DTFAULTC register can be used to clear fault bits.			
0	DTPRS0F	0x0	R	**DTI PRS 0 Fault**
	This bit is set to 1 if a PRS 0 fault has occurred and DTPRS0FEN is set to 1. The TIMER0_DTFAULTC register can be used to clear fault bits.			

18.5.25 TIMER_DTFAULTC - DTI Fault Clear Register

Offset	Bit Position																													
0x0F8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												W(nB)	W(nB)	W(nB)	W(nB)	W(nB)
Name																												DTEM23FC	DTLOCKUPFC	DTDBGFC	DTPRS1FC	DTPRS0FC

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	DTEM23FC	0x0	W(nB)	**DTI EM23 Fault Clear**
	Write 1 to this bit to clear EM23 entry fault.			
3	DTLOCKUPFC	0x0	W(nB)	**DTI Lockup Fault Clear**
	Write 1 to this bit to clear core lockup fault.			
2	DTDBGFC	0x0	W(nB)	**DTI Debugger Fault Clear**
	Write 1 to this bit to clear debugger fault.			
1	DTPRS1FC	0x0	W(nB)	**DTI PRS1 Fault Clear**
	Write 1 to this bit to clear PRS 1 fault.			
0	DTPRS0FC	0x0	W(nB)	**DTI PRS0 Fault Clear**
	Write 1 to this bit to clear PRS 0 fault.			

18.5.26 TIMER_DTLOCK - DTI Configuration Lock Register

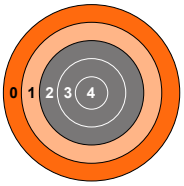
Offset	Bit Position																															
0x0FC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W							
Name																									DTILOCKKEY							

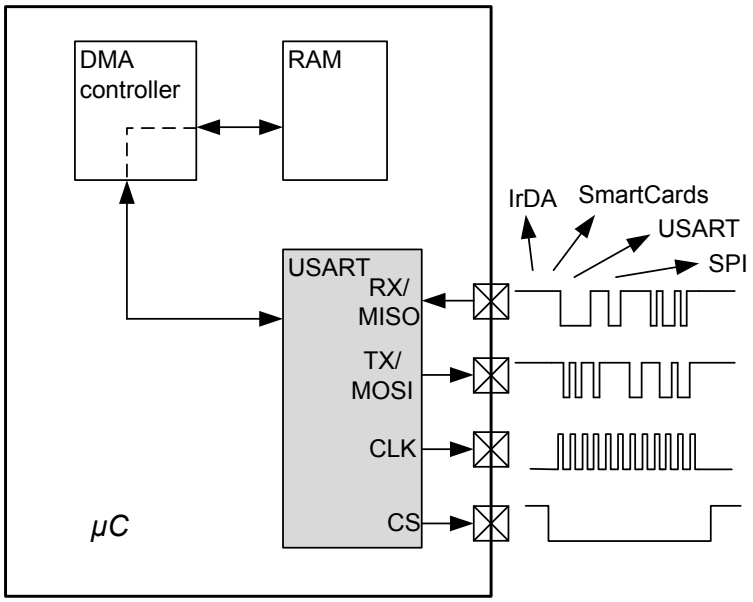
Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	DTILOCKKEY	0x0	W	**DTI Lock Key**

Write any other value than the unlock code to lock TIMER_ROUTE, TIMER_DTCTRL, TIMER_DTCFG, TIMER_DTTIMECFG and TIMER_DTFCFG from editing. Write the unlock code to unlock the DTI registers.

Value	Mode	Description
52864	UNLOCK	Write to unlock TIMER DTI registers

19. USART - Universal Synchronous Asynchronous Receiver/Transmitter





Quick Facts

What?

The USART handles high-speed UART, SPI-bus, SmartCards, and IrDA communication.

Why?

Serial communication is frequently used in embedded systems and the USART allows efficient communication with a wide range of external devices.

How?

The USART has a wide selection of operating modes, frame formats and baud rates. The multiprocessor mode allows the USART to remain idle when not addressed. Triple buffering and DMA support makes high data-rates possible with minimal CPU intervention and it is possible to transmit and receive large frames while the MCU remains in EM1 Sleep.

19.1 Introduction

The Universal Synchronous Asynchronous serial Receiver and Transmitter (USART) is a very flexible serial I/O module. It supports full duplex asynchronous UART communication as well as RS-485, SPI, MicroWire and 3-wire. It can also interface with ISO7816 SmartCards, and IrDA devices.

19.2 Features

- Asynchronous and synchronous (SPI) communication
- Full duplex and half duplex
- Separate TX/RX enable
- Separate receive / transmit multiple entry buffers, with additional separate shift registers
- Programmable baud rate, generated as an fractional division from the peripheral clock ($PCLK_{USARTn}$)
- Max bit-rate
  - Main SPI mode, peripheral clock rate/2
  - Secondary SPI mode, peripheral clock rate/6
  - UART mode, peripheral clock rate/16, 8, 6, or 4
- Asynchronous mode supports
  - Majority vote baud-reception
  - False start-bit detection
  - Break generation/detection
  - Multi-processor mode
- Synchronous mode supports
  - All 4 SPI clock polarity/phase configurations
  - Main and Secondary interface modes
- Data can be transmitted LSB first or MSB first
- Configurable number of data bits, 4-16 (plus the parity bit, if enabled)
  - HW parity bit generation and check
- Configurable number of stop bits in asynchronous mode: 0.5, 1, 1.5, 2
- HW collision detection
- Multi-processor mode
- IrDA modulator
- SmartCard (ISO7816) mode
- I2S mode
- Separate interrupt vectors for receive and transmit interrupts
- Loopback mode
  - Half duplex communication
  - Communication debugging
- PRS RX input
- 8 bit Timer
- Hardware Flow Control
- Automatic Baud Rate Detection

19.3 Functional Description

An overview of the USART module is shown in Figure 19.1 USART Overview on page 534.

This section describes all posible USART features. Please refer to the Device Datasheet to see what features a specific USART instance supports.

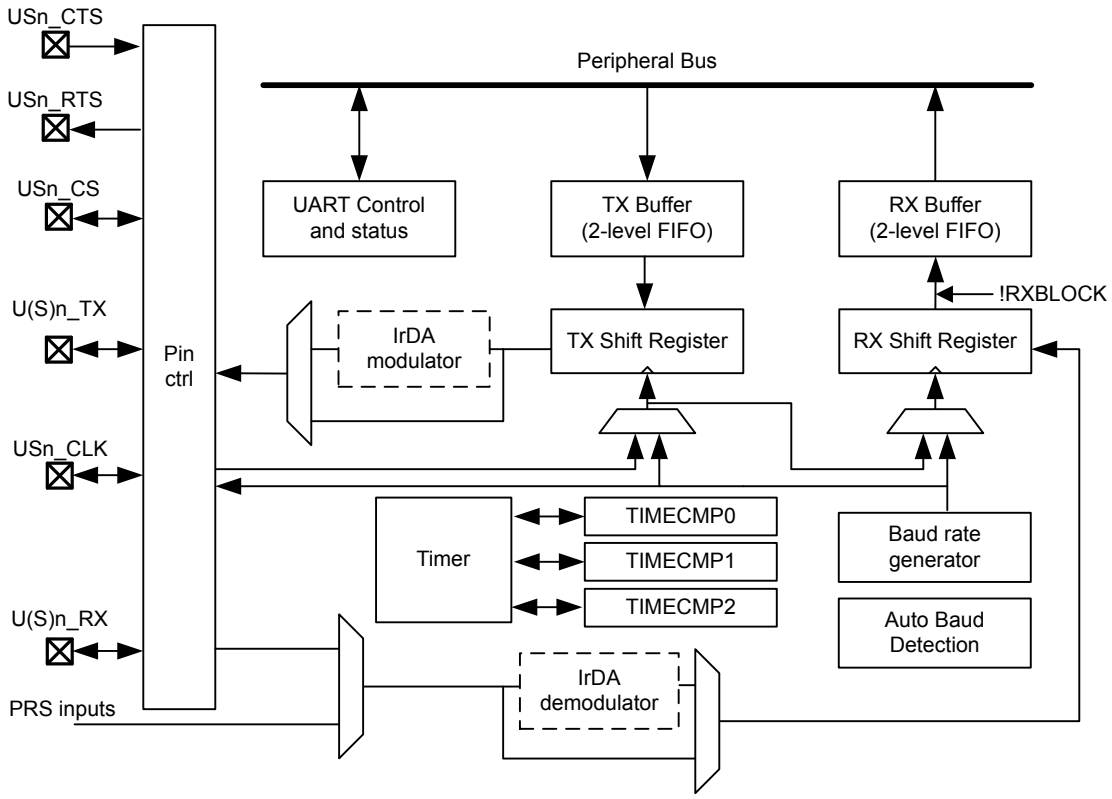


Figure 19.1. USART Overview

19.3.1 Modes of Operation

The USART operates in either asynchronous or synchronous mode.

In synchronous mode, a separate clock signal is transmitted with the data. This clock signal is generated by the main interface on the bus, and both the main and secondary devices sample and transmit data according to this clock. Both main and secondary interface modes are supported by the USART. The synchronous communication mode is compatible with the Serial Peripheral Interface Bus (SPI) standard.

In asynchronous mode, no separate clock signal is transmitted with the data on the bus. The USART receiver thus has to determine where to sample the data on the bus from the actual data. To make this possible, additional synchronization bits are added to the data when operating in asynchronous mode, resulting in a slight overhead.

Asynchronous or synchronous mode can be selected by configuring SYNC in USARTn_CTRL. The options are listed with supported protocols in Table 19.1 USART Asynchronous vs. Synchronous Mode on page 535. Full duplex and half duplex communication is supported in both asynchronous and synchronous mode.

Table 19.1. USART Asynchronous vs. Synchronous Mode

SYNC	Communication Mode	Supported Protocols
0	Asynchronous	RS-232, RS-485 (w/external driver), IrDA, ISO 7816
1	Synchronous	SPI, MicroWire, 3-wire

Table 19.2 USART Pin Usage on page 535 explains the functionality of the different USART pins when the USART operates in different modes. Pin functionality enclosed in square brackets is optional, and depends on additional configuration parameters. LOOPBK and MASTER are discussed in 19.3.2.14 Local Loopback and 19.3.3.3 Synchronous Main Interface Mode respectively.

Table 19.2. USART Pin Usage

SYNC	LOOPBK	MASTER	Pin functionality			
			U(S)n_TX (MOSI)	U(S)n_RX (MISO)	USn_CLK	USn_CS
0	0	x	Data out	Data in	-	[Driver enable]
0	1	x	Data out/in	-	-	[Driver enable]
1	0	0	Data in	Data out	Clock in	Secondary select
1	0	1	Data out	Data in	Clock out	[Auto secondary select]
1	1	0	Data out/in	-	Clock in	Secondary select
1	1	1	Data out/in	-	Clock out	[Auto secondary select]

19.3.2 Asynchronous Operation

19.3.2.1 Frame Format

The frame format used in asynchronous mode consists of a set of data bits in addition to bits for synchronization and optionally a parity bit for error checking. A frame starts with one start-bit (S), where the line is driven low for one bit-period. This signals the start of a frame, and is used for synchronization. Following the start bit are 4 to 16 data bits and an optional parity bit. Finally, a number of stop-bits, where the line is driven high, end the frame. An example frame is shown in Figure 19.2 USART Asynchronous Frame Format on page 536.

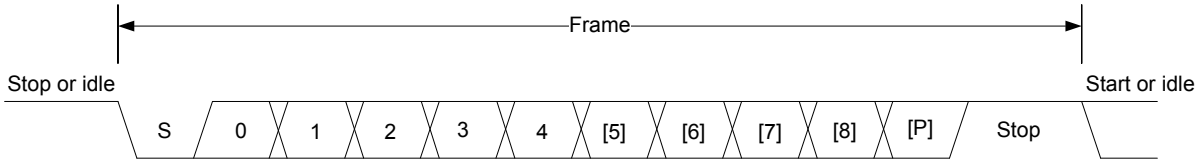


Figure 19.2. USART Asynchronous Frame Format

The number of data bits in a frame is set by DATABITS in USARTn_FRAME, see Table 19.3 USART Data Bits on page 536, and the number of stop-bits is set by STOPBITS in USARTn_FRAME, see Table 19.4 USART Stop Bits on page 536. Whether or not a parity bit should be included, and whether it should be even or odd is defined by PARITY, also in USARTn_FRAME. For communication to be possible, all parties of an asynchronous transfer must agree on the frame format being used.

Table 19.3. USART Data Bits

DATA BITS [3:0]	Number of Data bits
0001	4
0010	5
0011	6
0100	7
0101	8 (Default)
0110	9
0111	10
1000	11
1001	12
1010	13
1011	14
1100	15
1101	16

Table 19.4. USART Stop Bits

STOP BITS [1:0]	Number of Stop bits
00	0.5
01	1 (Default)
10	1.5
11	2

The order in which the data bits are transmitted and received is defined by MSBF in USARTn_CTRL. When MSBF is cleared, data in a frame is sent and received with the least significant bit first. When it is set, the most significant bit comes first.

The frame format used by the transmitter can be inverted by setting TXINV in USARTn_CTRL, and the format expected by the receiver can be inverted by setting RXINV in USARTn_CTRL. These bits affect the entire frame, not only the data bits. An inverted frame has a low idle state, a high start-bit, inverted data and parity bits, and low stop-bits.

19.3.2.2 Parity Bit Calculation and Handling

When parity bits are enabled, hardware automatically calculates and inserts any parity bits into outgoing frames, and verifies the received parity bits in incoming frames. This is true for both asynchronous and synchronous modes, even though it is mostly used in asynchronous communication. The possible parity modes are defined in Table 19.5 USART Parity Bits on page 537. When even parity is chosen, a parity bit is inserted to make the number of high bits (data + parity) even. If odd parity is chosen, the parity bit makes the total number of high bits odd.

Table 19.5. USART Parity Bits

PARITY BITS [1:0]	Description
00	No parity bit (Default)
01	Reserved
10	Even parity
11	Odd parity

19.3.2.3 Clock Generation

The USART clock defines the transmission and reception data rate. When operating in asynchronous mode, the baud rate (bit-rate) is given by Figure 19.3 USART Baud Rate on page 538.

$$br = f_{PCLK}/(oversample \times (1 + USARTn\_CLKDIV/256))$$

Figure 19.3. USART Baud Rate

where f_{PCLK} is the peripheral clock ($PCLK_{USARTn}$) frequency and oversample is the oversampling rate as defined by OVS in USARTn_CTRL, see Table 19.6 USART Oversampling on page 538.

Table 19.6. USART Oversampling

OVS [1:0]	oversample
00	16
01	8
10	6
11	4

The USART has a fractional clock divider to allow the USART clock to be controlled more accurately than what is possible with a standard integral divider.

The clock divider used in the USART is a 20-bit value, with a 15-bit integral part and an 5-bit fractional part. The fractional part is configured in the lower 5 bits of DIV in USART_CLKDIV. The lowest achievable baud rate at 32 MHz is about 61 bauds/sec.

Fractional clock division is implemented by distributing the selected fraction over four baud periods. The fractional part of the divider tells how many of these periods should be extended by one peripheral clock cycle.

Given a desired baud rate brdesired, the clock divider USARTn_CLKDIV can be calculated by using Figure 19.4 USART Desired Baud Rate on page 538:

$$USARTn\_CLKDIV = 256 \times (f_{PCLK}/(oversample \times brdesired) - 1)$$

Figure 19.4. USART Desired Baud Rate

Table 19.7 USART Baud Rates @ 4MHz Peripheral Clock with 20 bit CLKDIV on page 538 shows a set of desired baud rates and how accurately the USART is able to generate these baud rates when running at a 4 MHz peripheral clock, using 16x or 8x oversampling.

Table 19.7. USART Baud Rates @ 4MHz Peripheral Clock with 20 bit CLKDIV

Desired baud rate [baud/s]	USARTn_OVS =00			USARTn_OVS =01		
	USARTn_CLKDIV/256 (to 32nd position)	Actual baud rate [baud/s]	Error %	USARTn_CLKDIV/256 (to 32nd position)	Actual baud rate [baud/s]	Error %
600	415,6563	600,015	0,003	832,3438	599,9925	-0,001
1200	207,3438	1199,94	-0,005	415,6563	1200,03	0,003
2400	103,1563	2400,24	0,010	207,3438	2399,88	-0,005
4800	51,09375	4799,04	-0,020	103,1563	4800,48	0,010
9600	25,03125	9603,842	0,040	51,09375	9598,08	-0,020
14400	16,375	14388,49	-0,080	33,71875	14401,44	0,010
19200	12,03125	19184,65	-0,080	25,03125	19207,68	0,040
28800	7,6875	28776,98	-0,080	16,375	28776,98	-0,080

Desired baud rate [baud/s]	USARTn_OVS =00			USARTn_OVS =01		
	USARTn_CLKDIV/256 (to 32nd position)	Actual baud rate [baud/s]	Error %	USARTn_CLKDIV/256 (to 32nd position)	Actual baud rate [baud/s]	Error %
38400	5,5	38461,54	0,160	12,03125	38369,3	-0,080
57600	3,34375	57553,96	-0,080	7,6875	57553,96	-0,080
76800	2,25	76923,08	0,160	5,5	76923,08	0,160
115200	1,15625	115942	0,644	3,34375	115107,9	-0,080
230400	0,09375	228571,4	-0,794	1,15625	231884,1	0,644

19.3.2.4 Auto Baud Detection

Setting AUTOBAUDEN in USARTn_CLKDIV uses the first frame received to automatically set the baud rate provided that it contains 0x55 (IrDA uses 0x00). AUTOBAUDEN can be used in a simple LIN configuration to auto detect the SYNC byte. The receiver will measure the number of local clock cycles between the beginning of the START bit and the beginning of the 8th data bit. The DIV field in USARTn_CLKDIV will be overwritten with the new value. The OVS in USARTn_CTRL and the +1 count of the Baud Rate equation are already factored into the result that gets written into the DIV field. To restart autobaud detection, clear AUTOBAUDEN and set it high again. Since the auto baud detection is done over 8 baud times, only the upper 3 bits of the fractional part of the clock divider are populated.

19.3.2.5 Data Transmission

Asynchronous data transmission is initiated by writing data to the transmit buffer using one of the methods described in 19.3.2.6 Transmit Buffer Operation. When the transmission shift register is empty and ready for new data, a frame from the transmit buffer is loaded into the shift register, and if the transmitter is enabled, transmission begins. When the frame has been transmitted, a new frame is loaded into the shift register if available, and transmission continues. If the transmit buffer is empty, the transmitter goes to an idle state, waiting for a new frame to become available.

Transmission is enabled through the command register USARTn_CMD by setting TXEN, and disabled by setting TXDIS in the same command register. When the transmitter is disabled using TXDIS, any ongoing transmission is aborted, and any frame currently being transmitted is discarded. When disabled, the TX output goes to an idle state, which by default is a high value. Whether or not the transmitter is enabled at a given time can be read from TXENS in USARTn_STATUS.

When the USART transmitter is enabled and there is no data in the transmit shift register or transmit buffer, the TXC flag in USARTn_STATUS and the TXC interrupt flag in USARTn_IF are set, signaling that the transmission is complete. The TXC status flag is cleared when a new frame becomes available for transmission, but the TXC interrupt flag must be cleared by software.

19.3.2.6 Transmit Buffer Operation

The transmit-buffer is a multiple entry FIFO buffer. A frame can be loaded into the buffer by writing to USARTn_TXDATA, USARTn_TXDATAX, USARTn_TXDOUBLE or USARTn_TXDOUBLEX. Using USARTn_TXDATA allows 8 bits to be written to the buffer, while using USARTn_TXDOUBLE will write 2 frames of 8 bits to the buffer. If 9-bit frames are used, the 9th bit of the frames will in these cases be set to the value of BIT8DV in USARTn_CTRL.

To set the 9th bit directly and/or use transmission control, USARTn_TXDATAX and USARTn_TXDOUBLEX must be used. USARTn_TXDATAX allows 9 data bits to be written, as well as a set of control bits regarding the transmission of the written frame. Every frame in the buffer is stored with 9 data bits and additional transmission control bits. USARTn_TXDOUBLEX allows two frames, complete with control bits to be written at once. When data is written to the transmit buffer using USARTn_TXDATAX and USARTn_TXDOUBLEX, the 9th bit(s) written to these registers override the value in BIT8DV in USARTn_CTRL, and alone define the 9th bits that are transmitted if 9-bit frames are used. Figure 19.5 USART Transmit Buffer Operation on page 540 shows the basics of the transmit buffer when DATABITS in USARTn_FRAME is configured to less than 10 bits.

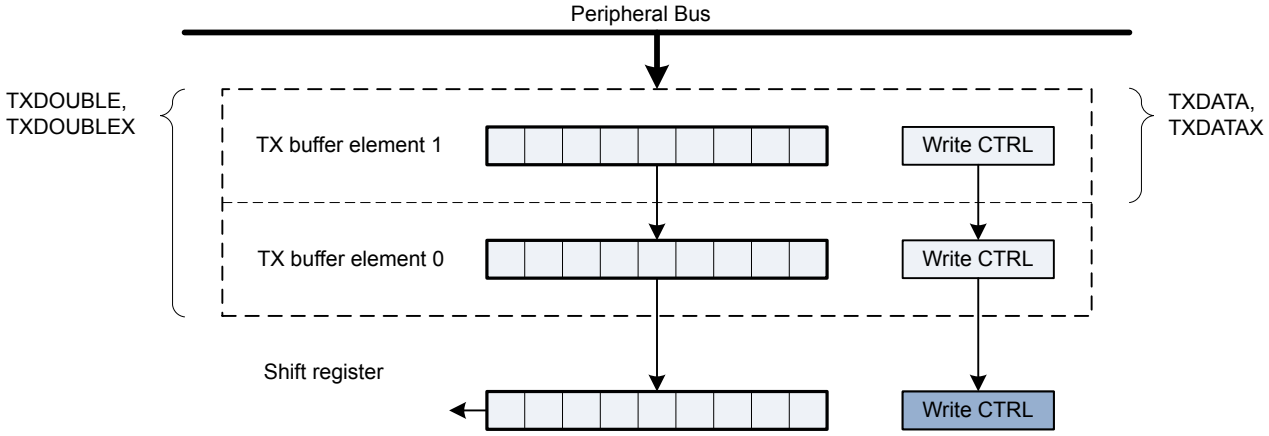


Figure 19.5. USART Transmit Buffer Operation

When writing more frames to the transmit buffer than there is free space for, the TXOF interrupt flag in USARTn_IF will be set, indicating the overflow. The data already in the transmit buffer is preserved in this case, and no data is written.

In addition to the interrupt flag TXC in USARTn_IF and status flag TXC in USARTn_STATUS which are set when the transmission is complete, TXBL in USARTn_STATUS and the TXBL interrupt flag in USARTn_IF are used to indicate the level of the transmit buffer. TXBIL in USARTn_CTRL controls the level at which these bits are set. If TXBIL is cleared, they are set whenever the transmit buffer becomes empty, and if TXBIL is set, they are set whenever the transmit buffer goes from full to half-full or empty. Both the TXBL status flag and the TXBL interrupt flag are cleared automatically when their condition becomes false.

The transmit buffer, including the transmit shift register can be cleared by setting CLEARTX in USARTn_CMD. This will prevent the USART from transmitting the data in the buffer and shift register, and will make them available for new data. Any frame currently being transmitted will not be aborted. Transmission of this frame will be completed.

19.3.2.7 Frame Transmission Control

The transmission control bits, which can be written using USARTn_TXDATAX and USARTn_TXDOUBLEX, affect the transmission of the written frame. The following options are available:

- Generate break: By setting TXBREAK, the output will be held low during the stop-bit period to generate a framing error. A receiver that supports break detection detects this state, allowing it to be used e.g. for framing of larger data packets. The line is driven high before the next frame is transmitted so the next start condition can be identified correctly by the recipient. Continuous breaks lasting longer than a USART frame are thus not supported by the USART. GPIO can be used for this.
- Disable transmitter after transmission: If TXDISAT is set, the transmitter is disabled after the frame has been fully transmitted.
- Enable receiver after transmission: If RXENAT is set, the receiver is enabled after the frame has been fully transmitted. It is enabled in time to detect a start-bit directly after the last stop-bit has been transmitted.
- Unblock receiver after transmission: If UBRXAT is set, the receiver is unblocked and RXBLOCK is cleared after the frame has been fully transmitted.
- Tristate transmitter after transmission: If TXTRIAT is set, TXTRI is set after the frame has been fully transmitted, tristating the transmitter output. Tristating of the output can also be performed automatically by setting AUTOTRI. If AUTOTRI is set TXTRI is always read as 0.

Note: When in SmartCard mode with repeat enabled, none of the actions, except generate break, will be performed until the frame is transmitted without failure. Generation of a break in SmartCard mode with repeat enabled will cause the USART to detect a NACK on every frame.

19.3.2.8 Data Reception

Data reception is enabled by setting RXEN in USARTn_CMD. When the receiver is enabled, it actively samples the input looking for a transition from high to low indicating the start baud of a new frame. When a start baud is found, reception of the new frame begins if the receive shift register is empty and ready for new data. When the frame has been received, it is pushed into the receive buffer, making the shift register ready for another frame of data, and the receiver starts looking for another start baud. If the receive buffer is full, the received frame remains in the shift register until more space in the receive buffer is available. If an incoming frame is detected while both the receive buffer and the receive shift register are full, the data in the shift register is overwritten, and the RXOF interrupt flag in USARTn_IF is set to indicate the buffer overflow.

The receiver can be disabled by setting the command bit RXDIS in USARTn_CMD. Any frame currently being received when the receiver is disabled is discarded. Whether or not the receiver is enabled at a given time can be read out from RXENS in USARTn_STATUS.

19.3.2.9 Receive Buffer Operation

When data becomes available in the receive buffer, the RXDATAV flag in USARTn_STATUS, and the RXDATAV interrupt flag in USARTn_IF are set, and when the buffer becomes full, RXFULL in USARTn_STATUS and the RXFULL interrupt flag in USARTn_IF are set. The status flags RXDATAV and RXFULL are automatically cleared by hardware when their condition is no longer true. This also goes for the RXDATAV interrupt flag, but the RXFULL interrupt flag must be cleared by software. When the RXFULL flag is set, notifying that the buffer is full, space is still available in the receive shift register for one more frame.

Data can be read from the receive buffer in a number of ways. USARTn_RXDATA gives access to the 8 least significant bits of the received frame, and USARTn_RXDOUBLE makes it possible to read the 8 least significant bits of two frames at once, pulling two frames from the buffer. To get access to the 9th, most significant bit, USARTn_RXDATAX must be used. This register also contains status information regarding the frame. USARTn_RXDOUBLEX can be used to get two frames complete with the 9th bits and status bits.

When a frame is read from the receive buffer using USARTn_RXDATA or USARTn_RXDATAX, the frame is pulled out of the buffer, making room for a new frame. USARTn_RXDOUBLE and USARTn_RXDOUBLEX pull two frames out of the buffer. If an attempt is done to read more frames from the buffer than what is available, the RXUF interrupt flag in USARTn_IF is set to signal the underflow, and the data read from the buffer is undefined.

Frames can be read from the receive buffer without removing the data by using USARTn_RXDATAXP and USARTn_RXDOUBLEXP. USARTn_RXDATAXP gives access the first frame in the buffer with status bits, while USARTn_RXDOUBLEXP gives access to both frames with status bits. The data read from these registers when the receive buffer is empty is undefined. If the receive buffer contains one valid frame, the first frame in USARTn_RXDOUBLEXP will be valid. No underflow interrupt is generated by a read using these registers, i.e. RXUF in USARTn_IF is never set as a result of reading from USARTn_RXDATAXP or USARTn_RXDOUBLEXP.

The basic operation of the receive buffer when DATABITS in USARTn_FRAME is configured to less than 10 bits is shown in Figure 19.6 USART Receive Buffer Operation on page 542.

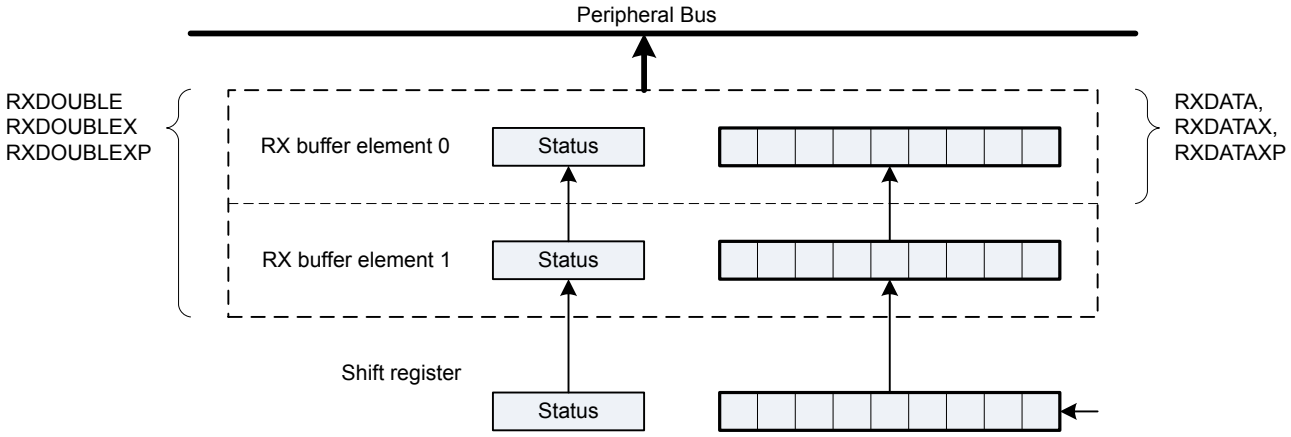


Figure 19.6. USART Receive Buffer Operation

The receive buffer, including the receive shift register can be cleared by setting CLEARRX in USARTn_CMD. Any frame currently being received will not be discarded.

19.3.2.10 Blocking Incoming Data

When using hardware frame recognition, as detailed in 19.3.2.20 Multi-Processor Mode and 19.3.2.21 Collision Detection, it is necessary to be able to let the receiver sample incoming frames without passing the frames to software by loading them into the receive buffer. This is accomplished by blocking incoming data.

Incoming data is blocked as long as RXBLOCK in USARTn_STATUS is set. When blocked, frames received by the receiver will not be loaded into the receive buffer, and software is not notified by the RXDATAV flag in USARTn_STATUS or the RXDATAV interrupt flag in USARTn_IF at their arrival. For data to be loaded into the receive buffer, RXBLOCK must be cleared in the instant a frame is fully received by the receiver. RXBLOCK is set by setting RXBLOCKEN in USARTn_CMD and disabled by setting RXBLOCKDIS also in USARTn_CMD. There is one exception where data is loaded into the receive buffer even when RXBLOCK is set. This is when an address frame is received when operating in multi-processor mode. See 19.3.2.20 Multi-Processor Mode for more information.

Frames received containing framing or parity errors will not result in the FERR and PERR interrupt flags in USARTn_IF being set while RXBLOCK in USARTn_STATUS is set. Hardware recognition is not applied to these erroneous frames, and they are silently discarded.

Note: If a frame is received while RXBLOCK in USARTn_STATUS is cleared, but stays in the receive shift register because the receive buffer is full, the received frame will be loaded into the receive buffer when space becomes available even if RXBLOCK is set at that time. The overflow interrupt flag RXOF in USARTn_IF will be set if a frame in the receive shift register, waiting to be loaded into the receive buffer is overwritten by an incoming frame even though RXBLOCK in USARTn_STATUS is set.

19.3.2.11 Clock Recovery and Filtering

The receiver samples the incoming signal at a rate 16, 8, 6 or 4 times higher than the given baud rate, depending on the oversampling mode given by OVS in USARTn_CTRL. Lower oversampling rates make higher baud rates possible, but give less room for errors.

When a high-to-low transition is registered on the input while the receiver is idle, this is recognized as a start-bit, and the baud rate generator is synchronized with the incoming frame.

For oversampling modes 16, 8 and 6, every bit in the incoming frame is sampled three times to gain a level of noise immunity. These samples are aimed at the middle of the bit-periods, as visualized in Figure 19.7 USART Sampling of Start and Data Bits on page 544. With OVS=0 in USARTn_CTRL, the start and data bits are thus sampled at locations 8, 9 and 10 in the figure, locations 4, 5 and 6 for OVS=1 and locations 3, 4, and 5 for OVS=2. The value of a sampled bit is determined by majority vote. If two or more of the three bit-samples are high, the resulting bit value is high. If the majority is low, the resulting bit value is low.

Majority vote is used for all oversampling modes except 4x oversampling. In this mode, a single sample is taken at position 3 as shown in Figure 19.7 USART Sampling of Start and Data Bits on page 544.

Majority vote can be disabled by setting MVDIS in USARTn_CTRL.

If the value of the start bit is found to be high, the reception of the frame is aborted, filtering out false start bits possibly generated by noise on the input.

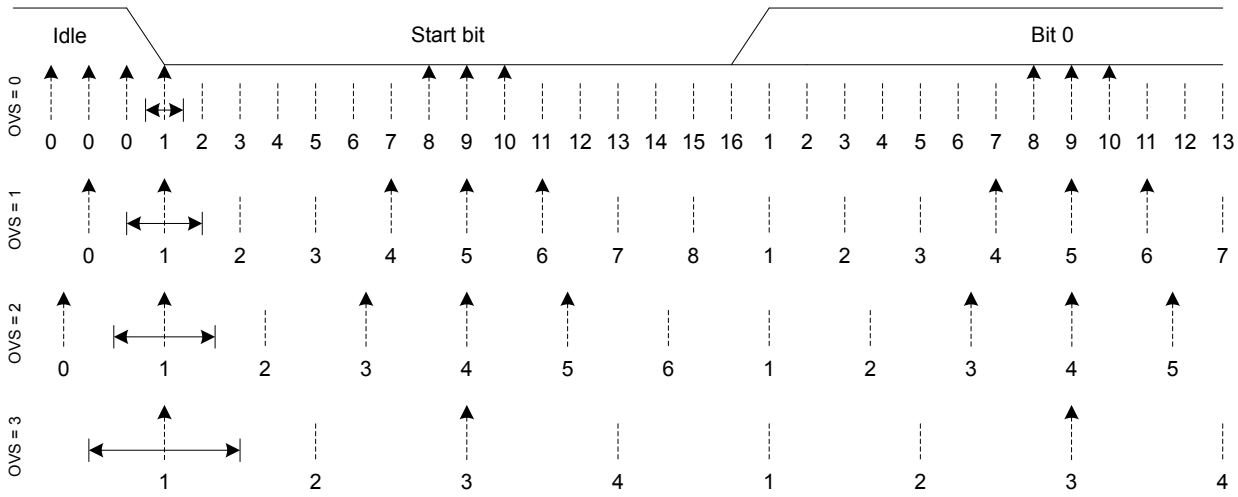


Figure 19.7. USART Sampling of Start and Data Bits

If the baud rate of the transmitter and receiver differ, the location each bit is sampled will be shifted towards the previous or next bit in the frame. This is acceptable for small errors in the baud rate, but for larger errors, it will result in transmission errors.

When the number of stop bits is 1 or more, stop bits are sampled like the start and data bits as seen in Figure 19.8 USART Sampling of Stop Bits when Number of Stop Bits are 1 or More on page 545. When a stop bit has been detected by sampling at positions 8, 9 and 10 for normal mode, or 4, 5 and 6 for smart mode, the USART is ready for a new start bit. As seen in Figure 19.8 USART Sampling of Stop Bits when Number of Stop Bits are 1 or More on page 545, a stop-bit of length 1 normally ends at c, but the next frame will be received correctly as long as the start-bit comes after position a for OVS=0 and OVS=3, and b for OVS=1 and OVS=2.

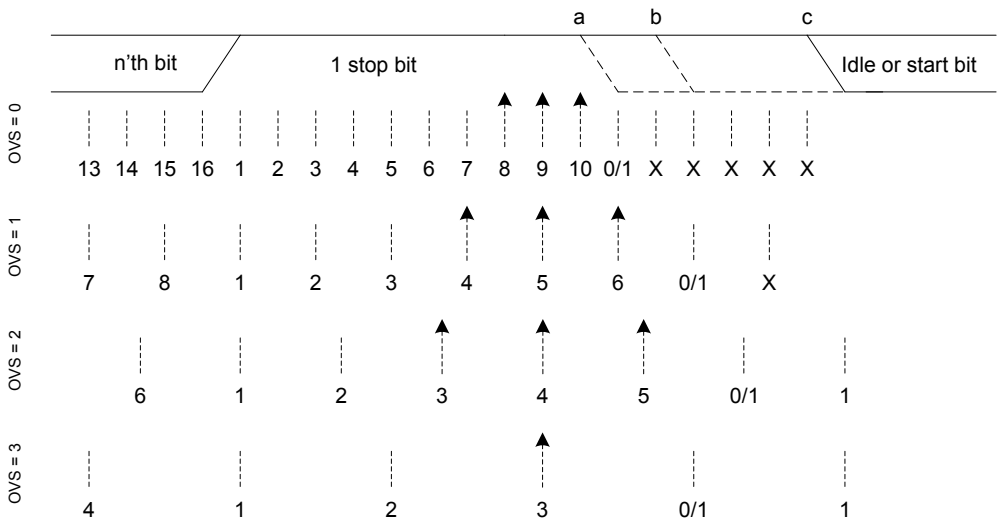


Figure 19.8. USART Sampling of Stop Bits when Number of Stop Bits are 1 or More

When working with stop bit lengths of half a baud period, the above sampling scheme no longer suffices. In this case, the stop-bit is not sampled, and no framing error is generated in the receiver if the stop-bit is not generated. The line must still be driven high before the next start bit however for the USART to successfully identify the start bit.

19.3.2.12 Parity Error

When parity bits are enabled, a parity check is automatically performed on incoming frames. When a parity error is detected in an incoming frame, the data parity error bit PERR in the frame is set, as well as the interrupt flag PERR in USARTn_IF. Frames with parity errors are loaded into the receive buffer like regular frames.

PERR can be accessed by reading the frame from the receive buffer using the USARTn_RXDATAX, USARTn_RXDATAXP, USARTn_RXDOUBLEX or USARTn_RXDOUBLEXP registers.

If ERRSTX in USARTn_CTRL is set, the transmitter is disabled on received parity and framing errors. If ERRSRX in USARTn_CTRL is set, the receiver is disabled on parity and framing errors.

19.3.2.13 Framing Error and Break Detection

A framing error is the result of an asynchronous frame where the stop bit was sampled to a value of 0. This can be the result of noise and baud rate errors, but can also be the result of a break generated by the transmitter on purpose.

When a framing error is detected in an incoming frame, the framing error bit FERR in the frame is set. The interrupt flag FERR in USARTn_IF is also set. Frames with framing errors are loaded into the receive buffer like regular frames.

FERR can be accessed by reading the frame from the receive buffer using the USARTn_RXDATAX, USARTn_RXDATAXP, USARTn_RXDOUBLEX or USARTn_RXDOUBLEXP registers.

If ERRSTX in USARTn_CTRL is set, the transmitter is disabled on parity and framing errors. If ERRSRX in USARTn_CTRL is set, the receiver is disabled on parity and framing errors.

19.3.2.14 Local Loopback

The USART receiver samples U(S)n_RX by default, and the transmitter drives U(S)n_TX by default. This is not the only option however. When LOOPBK in USARTn_CTRL is set, the receiver is connected to the U(S)n_TX pin as shown in Figure 19.9 USART Local Loopback on page 546. This is useful for debugging, as the USART can receive the data it transmits, but it is also used to allow the USART to read and write to the same pin, which is required for some half duplex communication modes. In this mode, the U(S)n_TX pin must be enabled as an output in the GPIO.

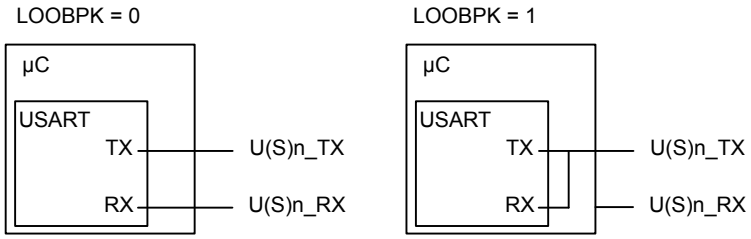


Figure 19.9. USART Local Loopback

19.3.2.15 Asynchronous Half Duplex Communication

When doing full duplex communication, two data links are provided, making it possible for data to be sent and received at the same time. In half duplex mode, data is only sent in one direction at a time. There are several possible half duplex setups, as described in the following sections.

19.3.2.16 Single Data-link

In this setup, the USART both receives and transmits data on the same pin. This is enabled by setting LOOPBK in USARTn_CTRL, which connects the receiver to the transmitter output. Because they are both connected to the same line, it is important that the USART transmitter does not drive the line when receiving data, as this would corrupt the data on the line.

When communicating over a single data-link, the transmitter must thus be tristated whenever not transmitting data. This is done by setting the command bit TXTRIEN in USARTn_CMD, which tristates the transmitter. Before transmitting data, the command bit TXTRIDIS, also in USARTn_CMD, must be set to enable transmitter output again. Whether or not the output is tristated at a given time can be read from TXTRI in USARTn_STATUS. If TXTRI is set when transmitting data, the data is shifted out of the shift register, but is not put out on U(S)n_TX.

When operating a half duplex data bus, it is common to have a main bus controller, which first transmits a request to one of the secondary devices on the bus, then receives a reply. In this case, the frame transmission control bits, which can be set by writing to USARTn_TXDATAX, can be used to make the USART automatically disable transmission, tristate the transmitter and enable reception when the request has been transmitted, making it ready to receive a response from the secondary device.

The timer, 19.3.10 Timer, can also be used to add delay between the RX and TX frames so that the interrupt service routine has time to process data that was just received before transmitting more data. Also hardware flow control is another method to insert time for processing the frame. RTS and CTS can be used to halt either the link partner's transmitter or the local transmitter. See the section on hardware flow control, 19.3.4 Hardware Flow Control, for more details.

Tristating the transmitter can also be performed automatically by the USART by using AUTOTRI in USARTn_CTRL. When AUTOTRI is set, the USART automatically tristates U(S)n_TX whenever the transmitter is idle, and enables transmitter output when the transmitter goes active. If AUTOTRI is set TXTRI is always read as 0.

Note: Another way to tristate the transmitter is to enable wired-and or wired-or mode in GPIO. For wired-and mode, outputting a 1 will be the same as tristating the output, and for wired-or mode, outputting a 0 will be the same as tristating the output. This can only be done on buses with a pull-up or pull-down resistor respectively.

19.3.2.17 Single Data-link with External Driver

Some communication schemes, such as RS-485 rely on an external driver. Here, the driver has an extra input which enables it, and instead of tristating the transmitter when receiving data, the external driver must be disabled.

This can be done manually by assigning a GPIO to turn the driver on or off, or it can be handled automatically by the USART. If AUTOCS in USARTn_CTRL is set, the USn_CS output is automatically activated a configurable number of baud periods before the transmitter starts transmitting data, and deactivated a configurable number of baud periods after the last bit has been transmitted and there is no more data in the transmit buffer to transmit. The number of baud periods are controlled by CSSETUP and CSHOLD in USARTn_TIMING. This feature can be used to turn the external driver on when transmitting data, and turn it off when the data has been transmitted.

The timer, 19.3.10 Timer, can also be used to configure CSSETUP and CSHOLD values between 1 to 256 baud-times by using TCMPVAL0, TCMPVAL1, or TCMPVAL2 for the TX sequencer.

USn_CS is immediately deasserted when the transmitter becomes disabled.

Note: When using CSSETUP in asynchronous mode with AUTOCS (USARTn_CTRL.SYNC = 0, USARTn_CTRL.AUTOCS = 1), TXDELAY in USARTn_TIMING should be set to 1.

Figure 19.10 USART Half Duplex Communication with External Driver on page 547 shows an example configuration where USn_CS is used to automatically enable and disable an external driver.

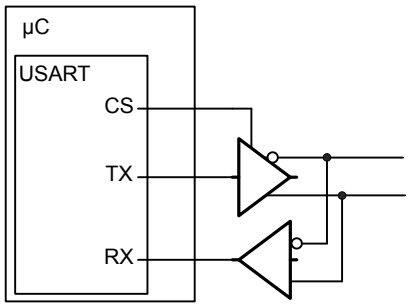


Figure 19.10. USART Half Duplex Communication with External Driver

The USn_CS output is active low by default, but its polarity can be changed with CSINV in USARTn_CTRL. AUTOCS works regardless of which mode the USART is in, so this functionality can also be used for automatic chip select when in synchronous mode (e.g. SPI).

19.3.2.18 Two Data-links

Some limited devices only support half duplex communication even though two data links are available. In this case software is responsible for making sure data is not transmitted when incoming data is expected.

TXARXnEN in USARTn_TRIGCTRL may be used to automatically start transmission after the end of the RX frame plus any TXSTDELAY and CSSETUP delay in USARTn_TIMING. For enabling the receiver either use RXENAT in USARTn_TXDATAX or RXATXnEN in USARTn_TRIGCTRL.

19.3.2.19 Large Frames

As each frame in the transmit and receive buffers holds a maximum of 9 bits, both the elements in the buffers are combined when working with USART-frames of 10 or more data bits.

To transmit such a frame, at least two elements must be available in the transmit buffer. If only one element is available, the USART will wait for the second element before transmitting the combined frame. Both the elements making up the frame are consumed when transmitting such a frame.

When using large frames, the 9th bits in the buffers are unused. For an 11 bit frame, the 8 least significant bits are thus taken from the first element in the buffer, and the 3 remaining bits are taken from the second element as shown in Figure 19.11 USART Transmission of Large Frames on page 548. The first element in the transmit buffer, i.e. element 0 in Figure 19.11 USART Transmission of Large Frames on page 548 is the first element written to the FIFO, or the least significant byte when writing two bytes at a time using USARTn_TXDOUBLE.

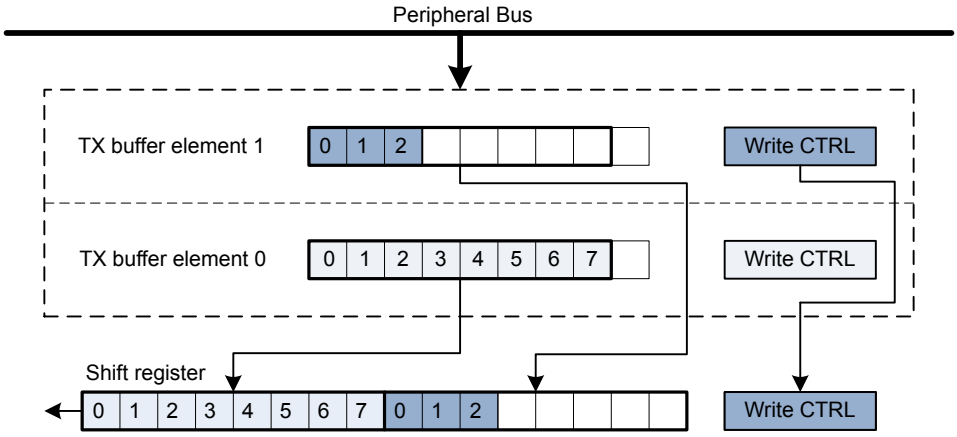


Figure 19.11. USART Transmission of Large Frames

As shown in Figure 19.11 USART Transmission of Large Frames on page 548, frame transmission control bits are taken from the second element in FIFO.

The two buffer elements can be written at the same time using the USARTn_TXDOUBLE or USARTn_TXDOUBLEX register. The TXDATAX0 bitfield then refers to buffer element 0, and TXDATAX1 refers to buffer element 1.

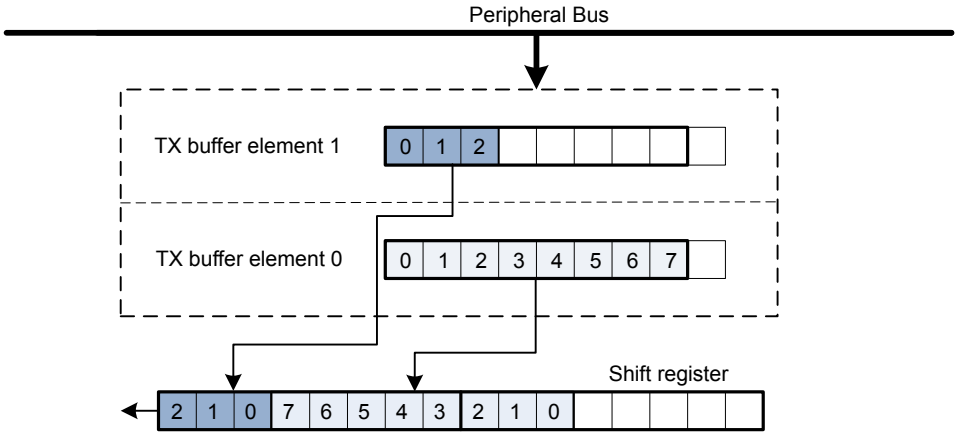


Figure 19.12. USART Transmission of Large Frames, MSBF

Figure 19.12 USART Transmission of Large Frames, MSBF on page 548 illustrates the order of the transmitted bits when an 11 bit frame is transmitted with MSBF set. If MSBF is set and the frame is smaller than 10 bits, only the contents of transmit buffer 0 will be transmitted.

When receiving a large frame, BYTESWAP in USARTn_CTRL determines the order the way the large frame is split into the two buffer elements. If BYTESWAP is cleared, the least significant 8 bits of the received frame are loaded into the first element of the receive buffer, and the remaining bits are loaded into the second element, as shown in Figure 19.13 USART Reception of Large Frames on page 549. The first byte read from the buffer thus contains the 8 least significant bits. Set BYTESWAP to reverse the order.

The status bits are loaded into both elements of the receive buffer. The frame is not moved from the receive shift register before there are two free spaces in the receive buffer.

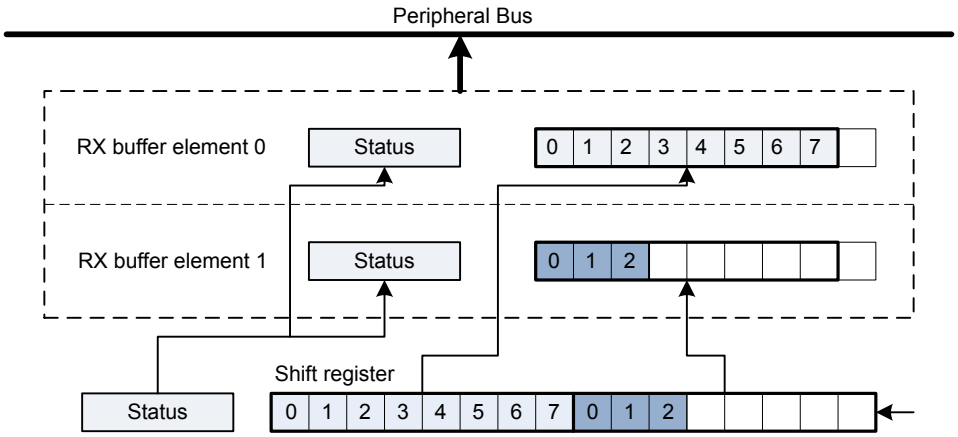


Figure 19.13. USART Reception of Large Frames

The two buffer elements can be read at the same time using the USARTn_RXDOUBLE or USARTn_RXDOUBLEX register. RXDATA0 then refers to buffer element 0 and RXDATA1 refers to buffer element 1.

Large frames can be used in both asynchronous and synchronous modes.

19.3.2.20 Multi-Processor Mode

To simplify communication between multiple processors, the USART supports a special multi-processor mode. In this mode the 9th data bit in each frame is used to indicate whether the content of the remaining 8 bits is data or an address.

When multi-processor mode is enabled, an incoming 9-bit frame with the 9th bit equal to the value of MPAB in USARTn_CTRL is identified as an address frame. When an address frame is detected, the MPAF interrupt flag in USARTn_IF is set, and the address frame is loaded into the receive register. This happens regardless of the value of RXBLOCK in USARTn_STATUS.

Multi-processor mode is enabled by setting MPM in USARTn_CTRL, and the value of the 9th bit in address frames can be set in MPAB. Note that the receiver must be enabled for address frames to be detected. The receiver can be blocked however, preventing data from being loaded into the receive buffer while looking for address frames.

When a device has received an address frame and wants to receive the following data, it must make sure the receiver is unblocked before the next frame has been completely received in order to prevent data loss.

BIT8DV in USARTn_CTRL can be used to specify the value of the 9th bit without writing to the transmit buffer with USARTn_TXDATAX or USARTn_TXDOUBLEX, giving higher efficiency in multi-processor mode, as the 9th bit is only set when writing address frames, and 8-bit writes to the USART can be used when writing the data frames.

19.3.2.21 Collision Detection

The USART supports a basic form of collision detection. When the receiver is connected to the output of the transmitter, either by using the LOOPBK bit in USARTn_CTRL or through an external connection, this feature can be used to detect whether data transmitted on the bus by the USART did get corrupted by a simultaneous transmission by another device on the bus.

For collision detection to be enabled, CCEN in USARTn_CTRL must be set, and the receiver enabled. The data sampled by the receiver is then continuously compared with the data output by the transmitter. If they differ, the CCF interrupt flag in USARTn_IF is set. The collision check includes all bits of the transmitted frames. The CCF interrupt flag is set once for each bit sampled by the receiver that differs from the bit output by the transmitter. When the transmitter output is disabled, i.e. the transmitter is tristated, collisions are not registered.

19.3.2.22 SmartCard Mode

In SmartCard mode, the USART supports the ISO 7816 I/O line T0 mode. With exception of the stop-bits (guard time), the 7816 data frame is equal to the regular asynchronous frame. In this mode, the receiver pulls the line low for one baud, half a baud into the guard time to indicate a parity error. This NAK can for instance be used by the transmitter to re-transmit the frame. SmartCard mode is a half duplex asynchronous mode, so the transmitter must be tristated whenever not transmitting data.

To enable SmartCard mode, set SCMODE in USARTn_CTRL, set the number of databits in a frame to 8, and configure the number of stopbits to 1.5 by writing to STOPBITS in USARTn_FRAME.

The SmartCard mode relies on half duplex communication on a single line, so for it to work, both the receiver and transmitter must work on the same line. This can be achieved by setting LOOPBK in USARTn_CTRL or through an external connection. The TX output should be configured as open-drain in the GPIO module.

When no parity error is identified by the receiver, the data frame is as shown in Figure 19.14 USART ISO 7816 Data Frame Without Error on page 550. The frame consists of 8 data bits, a parity bit, and 2 stop bits. The transmitter does not drive the output line during the guard time.

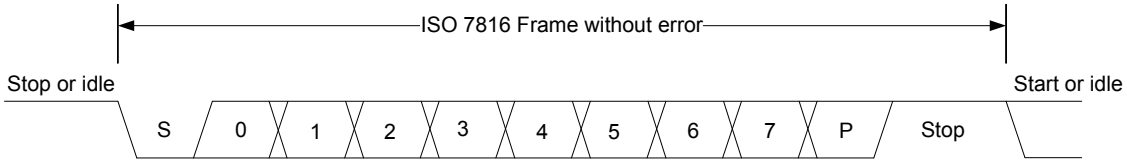


Figure 19.14. USART ISO 7816 Data Frame Without Error

If a parity error is detected by the receiver, it pulls the line I/O line low after half a stop bit, see Figure 19.15 USART ISO 7816 Data Frame With Error on page 550. It holds the line low for one bit-period before it releases the line. In this case, the guard time is extended by one bit period before a new transmission can start, resulting in a total of 3 stop bits.

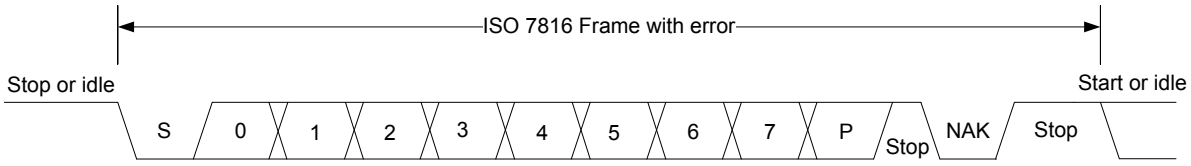


Figure 19.15. USART ISO 7816 Data Frame With Error

On a parity error, the NAK is generated by hardware. The NAK generated by the receiver is sampled as the stop-bit of the frame. Because of this, parity errors when in SmartCard mode are reported with both a parity error and a framing error.

When transmitting a T0 frame, the USART receiver on the transmitting side samples position 16, 17 and 18 in the stop-bit to detect the error signal when in 16x oversampling mode as shown in Figure 19.16 USART SmartCard Stop Bit Sampling on page 551. Sampling at this location places the stop-bit sample in the middle of the bit-period used for the error signal (NAK).

If a NAK is transmitted by the receiver, it will thus appear as a framing error at the transmitter, and the FERR interrupt flag in USARTn_IF will be set. If SCRETRANS USARTn_CTRL is set, the transmitter will automatically retransmit a NACK'ed frame. The transmitter will retransmit the frame until it is ACK'ed by the receiver. This only works when the number of databits in a frame is configured to 8.

Set SKIPPERRF in USARTn_CTRL to make the receiver discard frames with parity errors. The PERR interrupt flag in USARTn_IF is set when a frame is discarded because of a parity error.

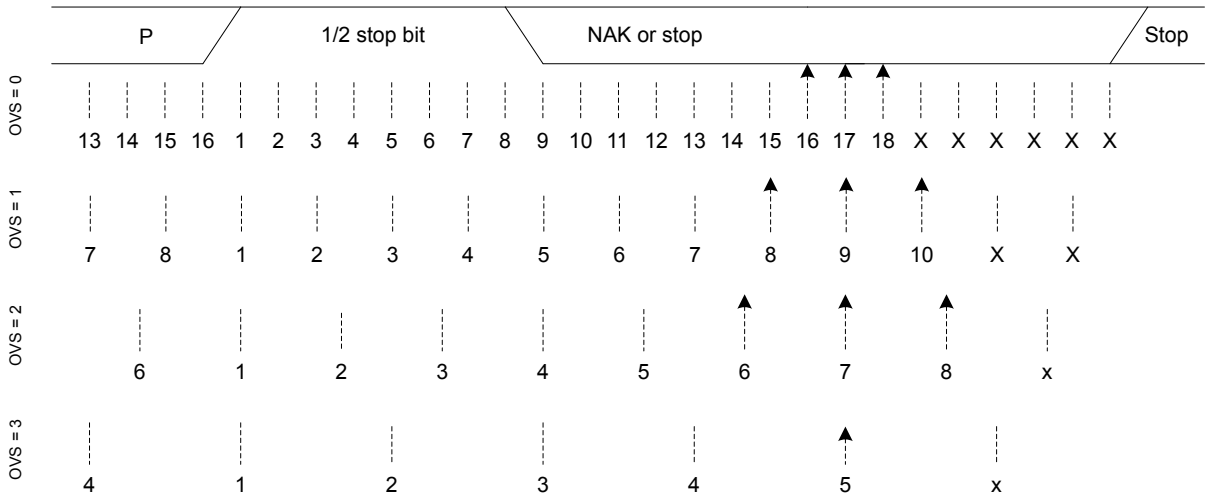


Figure 19.16. USART SmartCard Stop Bit Sampling

For communication with a SmartCard, a clock signal needs to be generated for the card. This clock output can be generated using one of the timers. See the ISO 7816 specification for more info on this clock signal.

SmartCard T1 mode is also supported. The T1 frame format used is the same as the asynchronous frame format with parity bit enabled and one stop bit. The USART must then be configured to operate in asynchronous half duplex mode.

19.3.3 Synchronous Operation

Most of the features in asynchronous mode are available in synchronous mode. Multi-processor mode can be enabled for 9-bit frames, loopback is available and collision detection can be performed.

19.3.3.1 Frame Format

The frames used in synchronous mode need no start and stop bits since a single clock is available to all parts participating in the communication. Parity bits cannot be used in synchronous mode.

The USART supports frame lengths of 4 to 16 bits per frame. Larger frames can be simulated by transmitting multiple smaller frames, i.e. a 22 bit frame can be sent using two 11-bit frames, and a 21 bit frame can be generated by transmitting three 7-bit frames. The number of bits in a frame is set using DATABITS in USARTn_FRAME.

The frames in synchronous mode are by default transmitted with the least significant bit first like in asynchronous mode. The bit-order can be reversed by setting MSBF in USARTn_CTRL.

The frame format used by the transmitter can be inverted by setting TXINV in USARTn_CTRL, and the format expected by the receiver can be inverted by setting RXINV, also in USARTn_CTRL.

19.3.3.2 Clock Generation

The bit-rate in synchronous mode is given by Figure 19.17 USART Synchronous Mode Bit Rate on page 552. As in the case of asynchronous operation, the clock division factor have a 15-bit integral part and a 5-bit fractional part.

$$br = f_{PCLK}/(2 \times (1 + USARTn\_CLKDIV/256))$$

Figure 19.17. USART Synchronous Mode Bit Rate

Given a desired baud rate brdesired, the clock divider USARTn_CLKDIV can be calculated using Figure 19.18 USART Synchronous Mode Clock Division Factor on page 552

$$USARTn\_CLKDIV = 256 \times (f_{PCLK}/(2 \times brdesired) - 1)$$

Figure 19.18. USART Synchronous Mode Clock Division Factor

When the USART operates as a synchronous main interface, the highest possible bit rate is half the peripheral clock rate. When operating as a secondary interface however, the highest bit rate is one sixth of the peripheral clock:

- Main interface mode: $br_{max} = f_{PCLK}/2$
- Secondary interface mode: $br_{max} = f_{PCLK}/6$

On every clock edge data on the data lines, MOSI and MISO, is either set up or sampled. When CLKPHA in USARTn_CTRL is cleared, data is sampled on the leading clock edge and set-up is done on the trailing edge. If CLKPHA is set however, data is set-up on the leading clock edge, and sampled on the trailing edge. In addition to this, the polarity of the clock signal can be changed by setting CLKPOL in USARTn_CTRL, which also defines the idle state of the clock. This results in four different modes which are summarized in Table 19.8 USART SPI Modes on page 552. Figure 19.19 USART SPI Timing on page 552 shows the resulting timing of data set-up and sampling relative to the bus clock.

Table 19.8. USART SPI Modes

SPI mode	CLKPOL	CLKPHA	Leading edge	Trailing edge
0	0	0	Rising, sample	Falling, set-up
1	0	1	Rising, set-up	Falling, sample
2	1	0	Falling, sample	Rising, set-up
3	1	1	Falling, set-up	Rising, sample

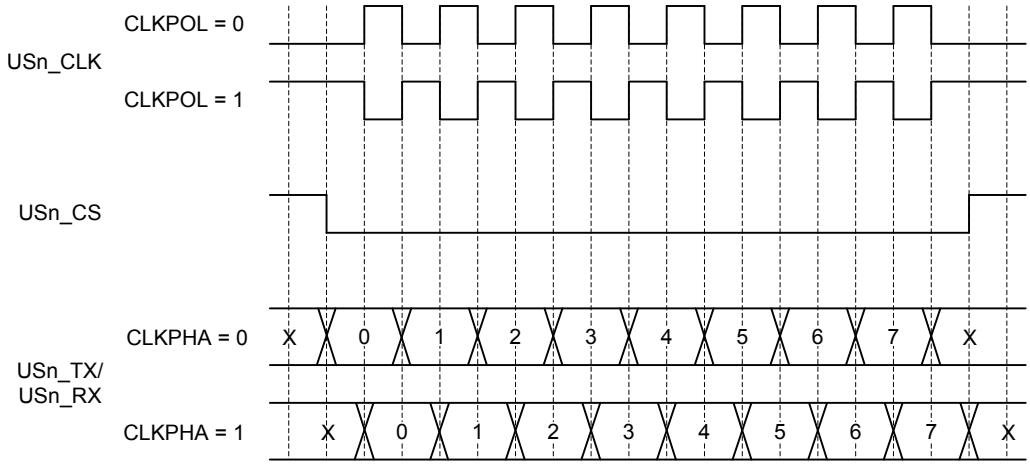


Figure 19.19. USART SPI Timing

If CPHA=1, the TX underflow flag, TXUF, will be set on the first setup clock edge of a frame in secondary mode if TX data is not available. If CPHA=0, TXUF is set if data is not available in the transmit buffer three PCLK cycles prior to the first sample clock edge. The RXDATAV flag is updated on the last sample clock edge of a transfer, while the RX overflow interrupt flag, RXOF, is set on the first

sample clock edge if the receive buffer overflows. When a transfer has been performed, interrupt flags TXBL and TXC are updated on the first setup clock edge of the succeeding frame, or when CS is deasserted.

19.3.3.3 Synchronous Main Interface Mode

When configured as a main interface, the USART is in full control of the data flow on the synchronous bus. When operating in full duplex mode, the secondary devices cannot transmit data to the main device without the main device transmitting to the secondary. The main device outputs the bus clock on USn_CLK.

Communication starts whenever there is data in the transmit buffer and the transmitter is enabled. The USART clock then starts, and the main device shifts bits out from the transmit shift register using the internal clock.

When there are no more frames in the transmit buffer and the transmit shift register is empty, the clock stops, and communication ends. When the receiver is enabled, it samples data using the internal clock when the transmitter transmits data. Operation of the RX and TX buffers is as in asynchronous mode.

19.3.3.4 Operation of USn_CS Pin

When operating as a synchronous main interface, the USn_CS pin can have one of two functions, or it can be disabled.

If USn_CS is configured as an output, it can be used to automatically generate a chip select for a secondary device by setting AUTOCS in USARTn_CTRL. If AUTOCS is set, USn_CS is activated before a transmission begins, and deactivated after the last bit has been transmitted and there is no more data in the transmit buffer.

The time between when CS is asserted and the first bit is transmitted can be controlled using the USART Timer and with CSSETUP in USARTn_TIMING. Any of the three comparators can be used to set this delay. If new data is ready for transmission before CS is deasserted, the data is sent without deasserting CS in between. CSHOLD in USARTn_TIMING keeps CS asserted after the end of frame for the number of baud-times specified.

By default, USn_CS is active low, but its polarity can be inverted by setting CSINV in USARTn_CTRL.

When USn_CS is configured as an input, it can be used by another synchronous main device that wants control of the bus to make the USART release it. When USn_CS is driven low, or high if CSINV is set, the interrupt flag SSM in USARTn_IF is set, and if CSMA in USARTn_CTRL is set, the USART goes to secondary mode.

19.3.3.5 AUTOTX

The main device on a synchronous bus is required to transmit data to a secondary device in order to receive data from that device. In some cases, only a few words are transmitted and a lot of data is then received from the secondary device. In that case, one solution is to keep feeding the TX with data to transmit, but that consumes system bandwidth. Instead AUTOTX can be used.

When AUTOTX in USARTn_CTRL is set, the USART transmits data as long as there is available space in the RX shift register for the chosen frame size. This happens even though there is no data in the TX buffer. The TX underflow interrupt flag TXUF in USARTn_IF is set on the first word that is transmitted which does not contain valid data.

During AUTOTX the USART will always send the previous sent bit, thus reducing the number of transitions on the TX output. So if the last bit sent was a 0, 0's will be sent during AUTOTX and if the last bit sent was a 1, 1's will be sent during AUTOTX.

19.3.3.6 Synchronous Secondary Interface Mode

When the USART is in synchronous secondary interface mode, data transmission is not controlled by the USART, but by an external synchronous main device. The USART is therefore not able to initiate a transmission, and has no control over the number of bytes written to the external main device.

The output and input to the USART are also swapped when in secondary mode, making the receiver take its input from USn_TX (MOSI) and the transmitter drive USn_RX (MISO).

To transmit data when in secondary mode, the device must load data into the transmit buffer and enable the transmitter. The data will remain in the USART until the main device starts a transmission by pulling the USn_CS input low and transmitting data. For every frame transmitted from main to secondary device, a frame is transferred from secondary to main as well. After a transmission, MISO remains in the same state as the last bit transmitted. This also applies if the main transmits to the secondary and the secondary TX buffer is empty.

If the transmitter is enabled in secondary synchronous mode and the main device starts transmission of a frame, the underflow interrupt flag TXUF in USARTn_IF will be set if no data is available for transmission.

If the secondary device needs to control its own chip select signal, this can be achieved by clearing CSPEN in the GPIO_USARTn_ROUTEEN register. The internal chip select signal can then be controlled through CSINV in the CTRL register. The chip select signal will be CSINV inverted, i.e. if CSINV is cleared, the chip select is active and vice versa.

19.3.3.7 Synchronous Half Duplex Communication

Half duplex communication in synchronous mode is very similar to half duplex communication in asynchronous mode as detailed in 19.3.2.15 Asynchronous Half Duplex Communication. The main difference is that in this mode, the main interface must generate the bus clock even when it is not transmitting data, i.e. it must provide the secondary device with a clock to receive data. To generate the bus clock, the main device should transmit data with the transmitter tristated, i.e. TXTRI in USARTn_STATUS set, when receiving data. If 2 bytes are expected from the secondary device, then transmit 2 bytes with the transmitter tristated, and the secondary uses the generated bus clock to transmit data to the main. TXTRI can be set by setting the TXTRIEN command bit in USARTn_CMD.

Note: When operating as SPI secondary interface in half duplex mode, TX has to be tristated (not disabled) during data reception if the device is to transmit data in the current transfer.

19.3.3.8 I2S

I2S is a synchronous format for transmission of audio data. The frame format is 32-bit, but since data is always transmitted with MSB first, an I2S device operating with 16-bit audio may choose to only process the 16 msb of the frame, and only transmit data in the 16 msb of the frame.

In addition to the bit clock used for regular synchronous transfers, I2S mode uses a separate word clock. When operating in mono mode, with only one channel of data, the word clock pulses once at the start of each new word. In stereo mode, the word clock toggles at the start of new words, and also gives away whether the transmitted word is for the left or right audio channel; A word transmitted while the word clock is low is for the left channel, and a word transmitted while the word clock is high is for the right.

When operating in I2S mode, the CS pin is used as a the word clock. In main mode, this is automatically driven by the USART, and in secondary mode, the word clock is expected from an external main device.

19.3.3.9 Word Format

The general I2S word format is 32 bits wide, but the USART also supports 16-bit and 8-bit words. In addition to this, it can be specified how many bits of the word should actually be used by the USART. These parameters are given by FORMAT in USARTn_I2SCTRL.

As an example, configuring FORMAT to using a 32-bit word with 16-bit data will make each word on the I2S bus 32-bits wide, but when receiving data through the USART, only the 16 most significant bits of each word can be read out of the USART. Similarly, only the 16 most significant bits have to be written to the USART when transmitting. The rest of the bits will be transmitted as zeroes.

19.3.3.10 Major Modes

The USART supports a set of different I2S formats as shown in Table 19.9 USART I2S Modes on page 555, but it is not limited to these modes. MONO, JUSTIFY and DELAY in USARTn_I2SCTRL can be mixed and matched to create an appropriate format. MONO enables mono mode, i.e. one data stream instead of two which is the default. JUSTIFY aligns data within a word on the I2S bus, either left or right which can bee seen in figures Figure 19.22 USART Left-Justified I2S Waveform on page 556 and Figure 19.23 USART Right-Justified I2S Waveform on page 556. Finally, DELAY specifies whether a new I2S word should be started directly on the edge of the word-select signal, or one bit-period after the edge.

Table 19.9. USART I2S Modes

Mode	MONO	JUSTIFY	DELAY	CLKPOL
Regular I2S	0	0	1	0
Left-Justified	0	0	0	1
Right-Justified	0	1	0	1
Mono	1	0	0	0

The regular I2S waveform is shown in Figure 19.20 USART Standard I2S Waveform on page 555 and Figure 19.21 USART Standard I2S Waveform (Reduced Accuracy) on page 555. The first figure shows a waveform transmitted with full accuracy. The wordlength can be configured to 32-bit, 16-bit or 8-bit using FORMAT in USARTn_I2SCTRL. In the second figure, I2S data is transmitted with reduced accuracy, i.e. the data transmitted has less bits than what is possible in the bus format.

Note that the msb of a word transmitted in regular I2S mode is delayed by one cycle with respect to word select

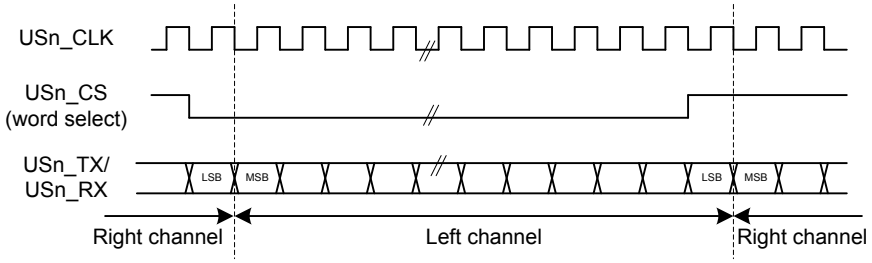


Figure 19.20. USART Standard I2S Waveform

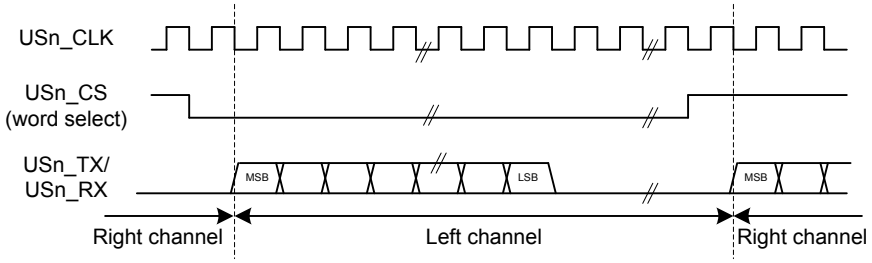


Figure 19.21. USART Standard I2S Waveform (Reduced Accuracy)

A left-justified stream is shown in Figure 19.22 USART Left-Justified I2S Waveform on page 556. Note that the MSB comes directly after the edge on the word-select signal in contradiction to the regular I2S waveform where it comes one bit-period after.

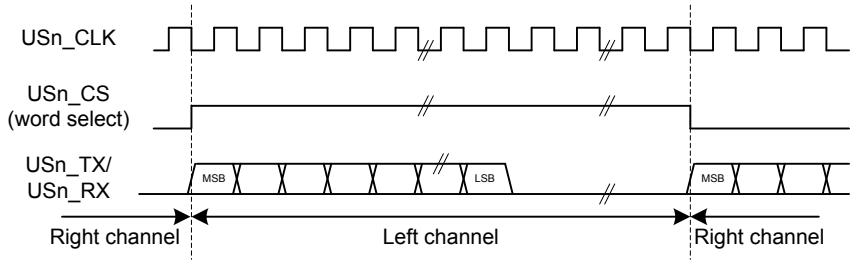


Figure 19.22. USART Left-Justified I2S Waveform

A right-justified stream is shown in Figure 19.23 USART Right-Justified I2S Waveform on page 556. The left and right justified streams are equal when the data-size is equal to the word-width.

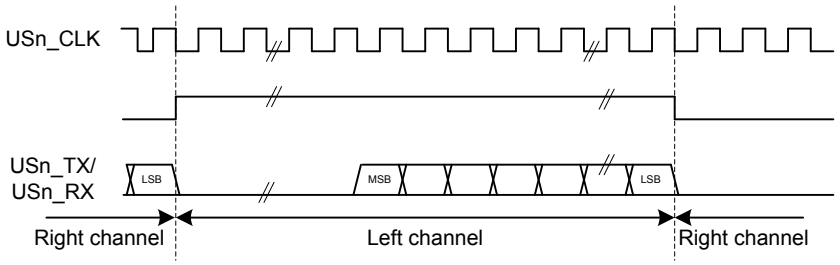


Figure 19.23. USART Right-Justified I2S Waveform

In mono-mode, the word-select signal pulses at the beginning of each word instead of toggling for each word. Mono I2S waveform is shown in Figure 19.24 USART Mono I2S Waveform on page 556.

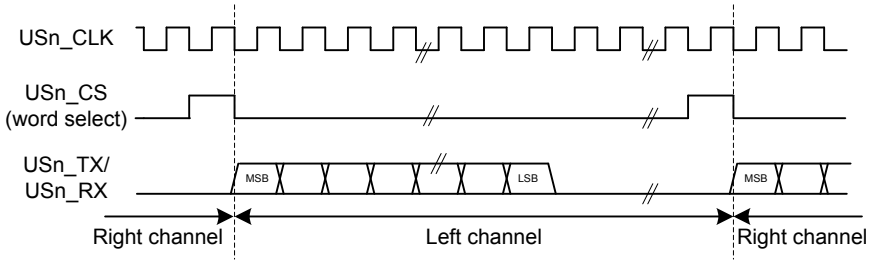


Figure 19.24. USART Mono I2S Waveform

19.3.3.11 Using I2S Mode

When using the USART in I2S mode, DATABITS in USARTn_FRAME must be set to 8 or 16 data-bits. 8 databits can be used in all modes, and 16 can be used in the modes where the number of bytes in the I2S word is even. In addition to this, MSBF in USARTn_CTRL should be set, and CLKPOL and CLKPHA in USARTn_CTRL should be cleared.

The USART does not have separate TX and RX buffers for left and right data, so when using I2S in stereo mode, the application must keep track of whether the buffers contain left or right data. This can be done by observing TXBLRIGHT, RXDATAVRIGHT and RXFULLRIGHT in USARTn_STATUS. TXBLRIGHT tells whether TX is expecting data for the left or right channel. It will be set with TXBL if right data is expected. The receiver will set RXDATAVRIGHT if there is at least one right element in the buffer, and RXFULLRIGHT if the buffer is full of right elements.

When using I2S with DMA, separate DMA requests can be used for left and right data by setting DMASPLIT in USARTn_I2SCTRL.

In both main and secondary mode the USART always starts transmitting on the LEFT channel after being enabled. In main mode, the transmission will stop if TX becomes empty. In that case, TXC is set. Continuing the transmission in this case will make the data-stream continue where it left off. To make the USART start on the LEFT channel after going empty, disable and re-enable TX.

19.3.4 Hardware Flow Control

Hardware flow control can be used to hold off the link partner's transmission until RX buffer space is available. The RTS and CTS signals are enabled and configured using the GPIO_USARTn_ROUTEEN, GPIO_USARTn_RTSROUTEx and GPIO_USARTn_CTSROUTE registers. RTS is an out going signal which indicates that RX buffer space is available to receive a frame. The link partner is being requested to send its data when RTS is asserted. CTS is an incoming signal to stop the next TX data from going out. When CTS is negated, the frame currently being transmitted is completed before stopping. CTS indicates that the link partner has RX buffer space available, and the local transmitter is clear to send. Also use CTSEN in USARTn_CTLX to enable the CTS input into the TX sequencer. For debug use set DBGHALT in USARTn_CTRLX which will force the RTS to request one frame from the link partner when the CPU core single steps.

19.3.5 Debug Halt

When DBGHALT in USART_CTRLX is clear, RTS is only dependent on the RX buffer having space available to receive data. Incoming data is always received until both the RX buffer is full and the RX shift register is full regardless of the state of DBGHALT or chip halt. Additional incoming data is discarded. When DBGHALT is set, RTS deasserts on RX buffer full or when chip halt is high. However, a low pulse detected on chip halt will keep RTS asserted when no frame is being received. At the start of frame reception, RTS will deassert if chip halt is high and DBGHALT is set. This behavior allows single stepping to pulse the chip halt low for a cycle, and receive the next frame. The link partner must stop transmitting when RTS is deasserted, or the RX buffer could overflow. All data in the transmit buffer is sent out even when chip halt is asserted; therefore, the DMA will need to be set to stop sending the USART TX data during chip halt.

19.3.6 PRS-triggered Transmissions

If a transmission must be started on an event with very little delay, the PRS system can be used to trigger the transmission. The PRS channel to use as a trigger can be selected using PRSSEL in PRS_USARTn_TRIGGER. When a positive edge is detected on this signal, the receiver is enabled if RXTEN in USARTn_TRIGCTRL is set, and the transmitter is enabled if TXTEN in USARTn_TRIGCTRL is set. Only one signal input is supported by the USART.

The AUTOTX feature can also be enabled via PRS. If an external SPI device sets a pin high when there is data to be read from the device, this signal can be routed to the USART through the PRS system and be used to make the USART clock data out of the external device. If AUTOTXTEN in USARTn_TRIGCTRL is set, the USART will transmit data whenever the PRS signal selected by PRS_USARTn_TRIGGER is high given that there is enough room in the RX buffer for the chosen frame size. Note that if there is no data in the TX buffer when using AUTOTX, the TX underflow interrupt will be set.

AUTOTXTEN can also be combined with TXTEN to make the USART transmit a command to the external device prior to clocking out data. To do this, disable TX using the TXDIS command, load the TX buffer with the command and enable AUTOTXTEN and TXTEN. When the selected PRS input goes high, the USART will now transmit the loaded command, and then continue clocking out while both the PRS input is high and there is room in the RX buffer

19.3.7 PRS RX Input

The USART can be configured to receive data directly from a PRS channel by setting RXPRSEN in USARTn_CTRLX. The PRS channel used is selected using PRSSEL in PRS_USARTn_RX.

19.3.8 PRS CLK Input

The USART can be configured to receive clock directly from a PRS channel by setting CLKPRSEN in USARTn_CTRLX. The PRS channel used is selected using PRSSEL in PRS_USARTn_CLK. This is useful in synchronous secondary mode and can together with RX PRS input be used to input data from PRS.

19.3.9 DMA Support

The USART has full DMA support. The DMA controller can write to the transmit buffer using the registers USARTn_TXDATA, USARTn_TXDATAX, USARTn_TXDOUBLE and USARTn_TXDOUBLEX, and it can read from the receive buffer using the registers USARTn_RXDATA, USARTn_RXDATAX, USARTn_RXDOUBLE and USARTn_RXDOUBLEX. This enables single byte transfers, 9 bit data + control/status bits, double byte and double byte + control/status transfers both to and from the USART.

A request for the DMA controller to read from the USART receive buffer can come from the following source:

- Data available in the receive buffer
- Data available in the receive buffer and data is for the RIGHT I2S channel. Only used in I2S mode.

A write request can come from one of the following sources:

- Transmit buffer and shift register empty. No data to send.
- Transmit buffer has room for more data. This does not check the TXBIL for half full. For DMA use, it is either full or empty.
- Transmit buffer has room for RIGHT I2S data. Only used in I2S mode

Even though there are two sources for write requests to the DMA, only one should be used at a time, since the requests from both sources are cleared even though only one of the requests are used.

In some cases, it may be sensible to temporarily stop DMA access to the USART when an error such as a framing error has occurred. This is enabled by setting ERRSDMA in USARTn_CTRL.

Note: For Synchronous mode full duplex operation, if both receive buffer and transmit buffer are served by DMA, to make sure receive buffer is not overflowed the settings below should be followed.

- The DMA channel that serves receive buffer should have higher priority than the DMA channel that serves transmit buffer.
- TXBL should be used as write request for transmit buffer DMA channel.
- IGNORESREQ should be set for both DMA channel.

19.3.10 Timer

In addition to the TX sequence timer, there is a versatile 8 bit timer that can generate up to three event pulses. These pulses can be used to create timing for a variety of uses such as RX timeout, break detection, response timeout, and RX enable delay. Transmission delay, CS setup, inter-character spacing, and CS hold use the TX sequence counter. The TX sequencer counter can use the three 8 bit compare values or preset values for delays. There is one general counter with three comparators. Each comparator has a start source, a stop source, a restart enable, and a timer compare value. The start source enables the comparator, resets the counter, and starts the counter. If the counter is already running, the start source will reset the counter and restart it.

Any comparator could start the counter using the same start source but have different timing events programmed into TCMPVALn in USARTn_TIMECMPn. The TCMP0, TCMP1, or TCMP2 events can be preempted by using the comparator stop source to disable the comparator before the counter reaches TCMPVAL0, TCMPVAL1, or TCMPVAL2. If one comparator gets disabled while the other comparator is still enabled, the counter continues counting. By default the counter will count up to 256 and stop unless a RESTARTEN is set in one of the USARTn_TIMECMPn registers. By using RESTARTEN and an interval programmed into TCMPVAL, an interval timer can be set up. The TSTART field needs to be changed to DISABLE to stop the interval timer. The timer stops running once all of the comparators are disabled. If a comparator's start and stop sources both trigger the same cycle, the TCMPn event triggers, the comparator stays enabled, and the counter begins counting from zero.

The TXDELAY, CSSETUP, ICS, and CSHOLD in USARTn_TIMING are used to program start of transmission delay, chip select setup delay, inter-character space, and chip select hold delay. Either a preset value of 0, 1, 2, 3, or 7 can be used for any of these delays; or the value in TCMPVALn may be used to set the delay. Using the preset values leaves the TCMPVALn free for other uses. The same TCMPVALn may be used for multiple events that require the same timing. The transmit sequencer's counter can run in parallel with the timer's counter. The counters and controls are shown in Figure 19.25 USART Timer Block Diagram on page 560.

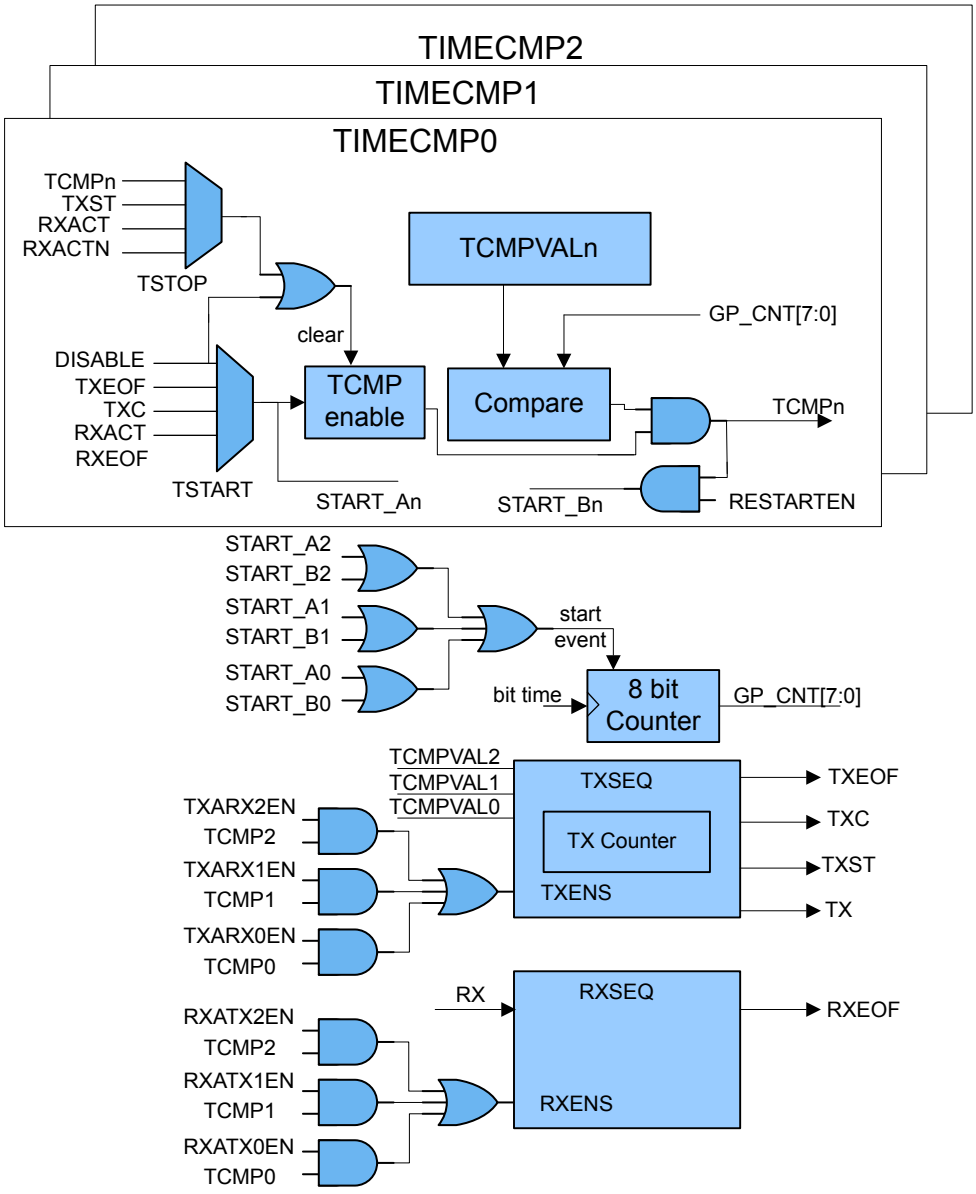


Figure 19.25. USART Timer Block Diagram

The following sections will go into more details on programming the various usage cases.

Table 19.10. USART Application Settings for USARTn_TIMING and USARTn_TIMECMPn

Application	TSTARTn	TSTOPn	TCMPVALn	Other
Response Timeout	TSTART0 = TXEOF	TSTOP0 = RXACT	TCMPVAL0 = 0x08	TCMP0 in USARTn_IEN
Receiver Timeout	TSTART1 = RXEOF	TSTOP1 = RXACT	TCMPVAL1 = 0x08	TCMP1 in USARTn_IEN
Large Receiver Timeout	TSTART1 = RXEOF, TCMP1	TSTOP1 = RXACT	TCMPVAL1 = 0xFF	TCMP1 in USARTn_IEN; TIMERRESTARTED in USARTn_STATUS; RESTART1EN in USARTn_TIMECMP1

Application	TSTARTn	TSTOPn	TCMPVALn	Other
Break Detect	TSTART1 = RXACT	TSTOP1 = RXACTN	TCMPVAL1 = 0x0C	TCMP1 in USARTn_IEN
TX delayed start of transmission and CS setup	TSTART0 = DISABLE, TSTART1 = DISABLE	TSTOP0 = TCMP0, TSTOP1 = TCMP1	TCMPVAL0 = 0x04, TCMPVAL1 = 0x02	TXDELAY = TCMP0, CSSETUP = TCMP1 in USARTn_TIMING; AUTOCS in USARTn_CTRL
TX inter-character spacing	TSTART2 = DISABLE	TSTOP2 = TCMP2	TCMPVAL2 = 0x03	ICS = TCMP2 in USARTn_TIMING; AUTOCS in USARTn_CTRL
TX Chip Select End Delay	TSTART1 = DISABLE	TSTOP1 = TCMP1	TCMPVAL1 = 0x04	CSHOLD = TCMP1 in USARTn_TIMING; AUTOCS in USARTn_CTRL
Response Delay	TSTART1 = RXEOF	TSTOP1 = TCMP1	TCMPVAL1 = 0x08	TXARX1EN in USARTn_TRIGCTRL
Combined TX and RX Example	TSTART1 = RXEOF, TSTART0 = TXEOF	TSTOP1 = TCMP1, TSTOP0 = TCMP0	TCMPVAL1 = 0x1C, TCMPVAL0 = 0x10	TXARX1EN, RXATX0EN in USARTn_TRIGCTRL; CSSETUP = 0x7, CSHOLD = 0x3 in USARTn_TIMING
Combined Delayed TX and Receiver Timeout Example	TSTART0 = TCMPVAL0, TSTART1 = RXEOF	TSTOP0 = RXACTN, TSTOP1 = RXACT	TCMPVAL0 = 0x20, TCMPVAL1 = 0x0C	TXARX0EN in USARTn_TRIGCTRL; TCMP0 in USARTn_IEN

Table 19.10 USART Application Settings for USARTn_TIMING and USARTn_TIMECMPn on page 560 shows some examples of how the USART timer can be programmed for various applications. The following sections will describe more details for each applications shown in the table.

19.3.10.1 Response Timeout

Response Timeout is when a UART transmitter sends a frame and expects another device to respond within a certain number of baud-times. Refer to Table 19.10 USART Application Settings for USARTn_TIMING and USARTn_TIMECMPn on page 560 for specific register settings. Comparator 0 will be looking for TX end of frame to use as the timer start source. For this example, a receiver start of frame RXACT has not been detected for 8 baud-times, and the TCMP0 interrupt in USARTn_IF is set. If an RX start bit is detected before the 8 baud-times, comparator 0 is disabled before the TCMP0 event can trigger.

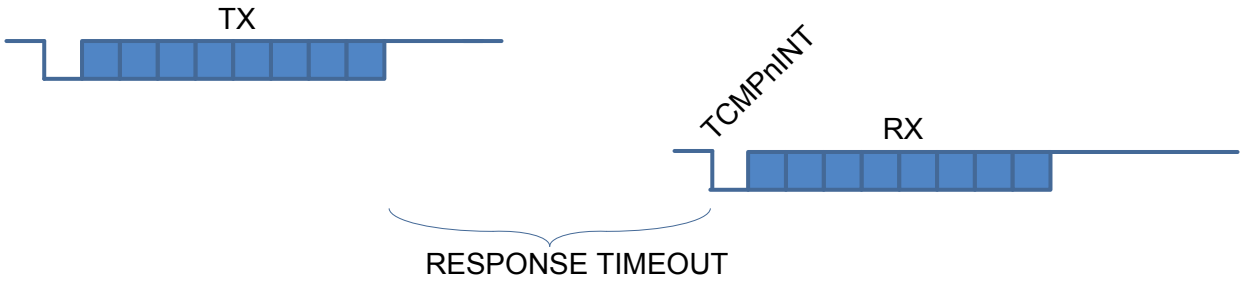


Figure 19.26. USART Response Timeout

19.3.10.2 RX Timeout

A receiver timeout function can be implemented by using the RX end of frame to start comparator 1 and look for the RX start bit RXACT to disable the comparator. See Table 19.10 USART Application Settings for USARTn_TIMING and USARTn_TIMECMPn on page 560 for details on setting up this example. As long as the next RX start bit occurs before the counter reaches the comparator 1 value TCMPVAL1, the interrupt will not get set. In this example the RX Timeout was set to 8 baud-times. To get an RX timeout larger than 256 baud-times, RESTART1EN in USARTn_TIMER can used to restart the counter when it reaches TCMPVAL1. By setting TCMPVAL1 in USARTn_TIMING to 0xFF, an interrupt will be generated after 256 baud-times. An interrupt service routine can then increment a memory location until the desired timeout is reached. Once the RX start bit is detected, comparator 1 will be disabled. If TIMERRESTARTED in USARTn_STATUS is clear, the TCMP1 interrupt is the first interrupt after RXEOF.

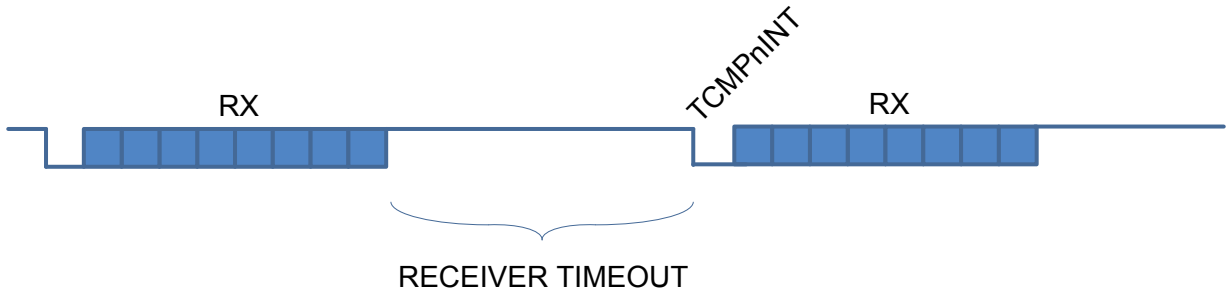


Figure 19.27. USART RX Timeout

19.3.10.3 Break Detect

LIN bus and half-duplex UARTs can take advantage of the timer configured for break detection where RX is held low for a number of baud-times to indicate a break condition. Table 19.10 USART Application Settings for USARTn_TIMING and USARTn_TIMECMPn on page 560 shows the settings for this mode. Each time RX is active (default of low) such as for a start bit, the timer begins counting. If the counter reaches 12 baud-times before RX goes to inactive RXACTN (default of high), an interrupt is asserted.

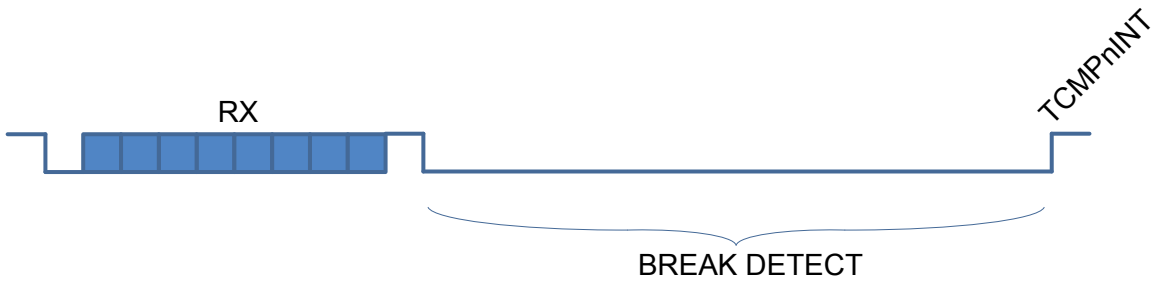


Figure 19.28. USART Break Detection

19.3.10.4 TX Start Delay

Some applications may require a delay before the start of transmission. This example in Figure 19.29 USART TXSEQ Timing on page 563 shows the TXSEQ timer used to delay the start of transmission by 4 baud times before the start of CS, and by 2 baud times with CS asserted. See Table 19.10 USART Application Settings for USARTn_TIMING and USARTn_TIMECMPn on page 560 for details on how to configure this mode. The TX sequencer could be enabled on PRS and start the TXSEQ counter running for 4 baud times as programmed in TCMPVAL0. Then CS is asserted for 2 baud times before the transmitter begins sending TX data. TXDELAY in USARTn_TIMING is the initial delay before any CS assertion, and CSSETUP is the delay during CS assertion. There are several small preset timing values such as 1, 2, 3, or 7 that can be used for some of the TX sequencer timing which leaves TCMPVAL0, TCMPVAL1, and TCMPVAL2 free for other uses.

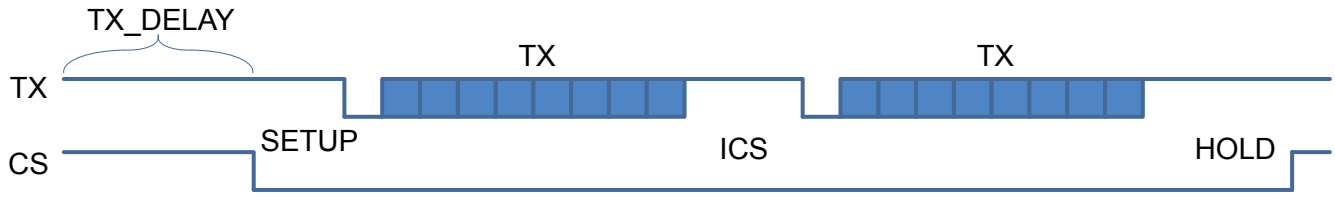


Figure 19.29. USART TXSEQ Timing

19.3.10.5 Inter-Character Space

In addition to delaying the start of frame transmission, it is sometimes necessary to also delay the time between each transmit character (inter-character space). After the first transmission, the inter-character space will delay the start of all subsequent transmissions until the transmit buffer is empty. See Table 19.10 USART Application Settings for USARTn_TIMING and USARTn_TIMECMPn on page 560 for details on setting up this example. For this example in Figure 19.29 USART TXSEQ Timing on page 563 ICS is set to TCMP2 in USARTn_TIMING. To keep CS asserted during the inter-character space, set AUTOCS in USARTn_CTRL. There are a few small pre-set timing values provided for TX sequence timing. Using these preset timing values can free up the TCMPVALn for other uses. For this example, the inter-character space is set to 0x03 and a preset value could be used.

19.3.10.6 TX Chip Select End Delay

The assertion of CS can be extended after the final character of the frame by using CSHOLD in USARTn_TIMING. See Table 19.10 USART Application Settings for USARTn_TIMING and USARTn_TIMECMPn on page 560 for details on setting up this example. AUTOCS in USARTn_CTRL needs to be set to extend the CS assertion after the last TX character is transmitted as shown in Figure 19.29 USART TXSEQ Timing on page 563.

19.3.10.7 Response Delay

A response delay can be used to hold off the transmitter until a certain number of baud-times after the RX frame. See Table 19.10 USART Application Settings for USARTn_TIMING and USARTn_TIMECMPn on page 560 for details on setting up this example. TXARX1EN in USARTn_TRIGCTRL tells the TX sequencer to trigger after RX EOF plus tcmp1val baud times.

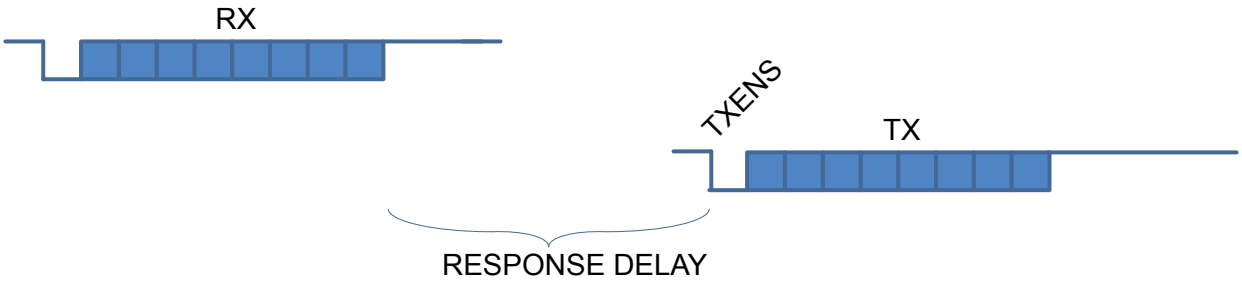


Figure 19.30. USART Response Delay

19.3.10.8 Combined TX and RX Example

This example describes how to alternate between TX and RX frames. This has a 28 baud-time space after RX and a 16 baud-time space after TX. The TSTART1 in USARTn_TIMECMP1 is set to RXEOF which uses the the receiver end of frame to start the timer. The TSTOP1 is set to TCMP1 to generate an event after 28 baud times. Set TXARX1EN in USARTn_TRIGCTRL, and the transmitter is held off until 28 baud times. TCMPVAL in USARTn_TIMECMP1 is set to 0x1C for 28 baud times. By setting TSTART0 in USARTn_TIMECMP0 to TXEOF, the timer will be started after the transmission has completed. RXATX0EN in USARTn_TRIGCTRL is used to delay enabling of the receiver until 16 baud times after the transmitter has completed. Write 0x10 into TCMPVAL of USARTn_TIMECMP0 for a 16 baud time delay. CS is also asserted 7 baud-times before start of transmission by setting CSSETUP to 0x7 in USARTn_TIMING. To keep CS asserted for 3 baud-times after transmission completes, CSHOLD is set to 0x3 in USARTn_TIMING. See Table 19.10 USART Application Settings for USARTn_TIMING and USARTn_TIMECMPn on page 560 for details on setting up this example.

19.3.10.9 Combined TX Delay and RX Break Detect

This example describes how to delay TX transmission after an RX frame and how to have a break condition signal an interrupt. See Table 19.10 USART Application Settings for USARTn_TIMING and USARTn_TIMECMPn on page 560 for details on setting up this example. The TX delay is set up by using transmit after RX, TXARX0EN in USARTn_TRIGCTRL to start the timer. TSTART0 in USARTn_TIMECMP0 is set to RXEOF which enables the transitter of the timer delay. For this example TCMPVAL in USARTn_TIMECMP0 is set to 0x20 to create a 32 baud-time delay between the end of the RX frame and the start of the TX frame. The break detect is configured by setting TSTART1 to RXACT to detect the start bit, and setting TSTOP1 to RXACTN to detect RX going high. In this case the interrupt asserts after RX stays low for 12 baud-times, so TCMPVAL1 is set to 0x0C.

19.3.10.10 Other Stop Conditions

There is also a timer stop on TX start using the TXST setting in TSTOP of USARTn_TIMECMPn. This can be used to see that the DMA has not written to the TXBUFFER for a given time.

19.3.11 Interrupts

The interrupts generated by the USART are combined into two interrupt vectors. Interrupts related to reception are assigned to one interrupt vector, and interrupts related to transmission are assigned to the other. Separating the interrupts in this way allows different priorities to be set for transmission and reception interrupts.

The transmission interrupt vector groups the transmission-related interrupts generated by the following interrupt flags:

- TXC
- TXBL
- TXOF
- CCF
- TXIDLE

The reception interrupt on the other hand groups the reception-related interrupts, triggered by the following interrupt flags:

- RXDATAV
- RXFULL
- RXOF
- RXUF
- PERR
- FERR
- MPAF
- SSM
- TCMPn

If USART interrupts are enabled, an interrupt will be made if one or more of the interrupt flags in USART_IF and their corresponding bits in USART_IEN are set.

19.3.12 IrDA Modulator/ Demodulator

The IrDA modulator implements the physical layer of the IrDA specification, which is necessary for communication over IrDA. The modulator takes the signal output from the USART module, and modulates it before it leaves the USART. In the same way, the input signal is demodulated before it enters the actual USART module. The modulator implements the original Rev. 1.0 physical layer and one high speed extension which supports speeds from 2.4 kbps to 1.152 Mbps.

The data from and to the USART is represented in a NRZ (Non Return to Zero) format, where the signal value is at the same level through the entire bit period. For IrDA, the required format is RZI (Return to Zero Inverted), a format where a “1” is signalled by holding the line low, and a “0” is signalled by a short high pulse. An example is given in Figure 19.31 USART Example RZI Signal for a given Asynchronous USART Frame on page 565.

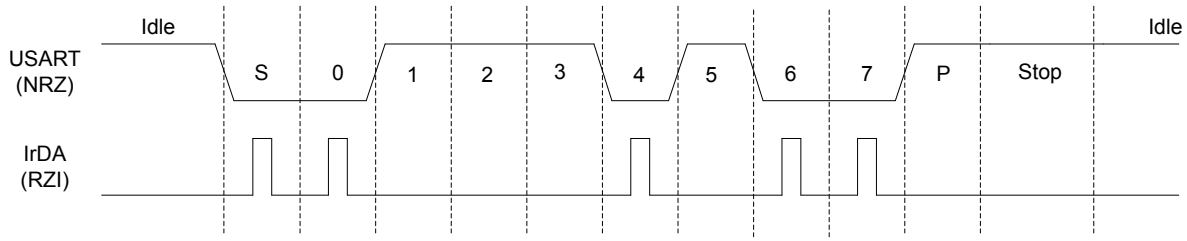


Figure 19.31. USART Example RZI Signal for a given Asynchronous USART Frame

The IrDA module is enabled by setting IREN. The USART transmitter output and receiver input is then routed through the IrDA modulator.

The width of the pulses generated by the IrDA modulator is set by configuring IRPW in USARTn_IRCTRL. Four pulse widths are available, each defined relative to the configured bit period as listed in Table 19.11 USART IrDA Pulse Widths on page 565.

Table 19.11. USART IrDA Pulse Widths

IRPW	Pulse width OVS=0	Pulse width OVS=1	Pulse width OVS=2	Pulse width OVS=3
00	1/16	1/8	1/6	1/4
01	2/16	2/8	2/6	N/A
10	3/16	3/8	N/A	N/A
11	4/16	N/A	N/A	N/A

By default, no filter is enabled in the IrDA demodulator. A filter can be enabled by setting IRFILT in USARTn_IRCTRL. When the filter is enabled, an incoming pulse has to last for 4 consecutive clock cycles to be detected by the IrDA demodulator.

Note that by default, the idle value of the USART data signal is high. This means that the IrDA modulator generates negative pulses, and the IrDA demodulator expects negative pulses. To make the IrDA module use RZI signalling, both TXINV and RXINV in USARTn_CTRL must be set.

19.4 USART Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	USART_IPVERSION	R	IPVERSION
0x004	USART_EN	RW	USART Enable
0x008	USART_CTRL	RW	Control Register
0x00C	USART_FRAME	RW	USART Frame Format Register
0x010	USART_TRIGCTRL	RW	USART Trigger Control Register
0x014	USART_CMD	W	Command Register
0x018	USART_STATUS	RH	USART Status Register
0x01C	USART_CLKDIV	RWH	Clock Control Register
0x020	USART_RXDATAX	RH	RX Buffer Data Extended Register
0x024	USART_RXDATA	RH	RX Buffer Data Register
0x028	USART_RXDOUBLEX	RH	RX Buffer Double Data Extended Register
0x02C	USART_RXDOUBLE	RH	RX FIFO Double Data Register
0x030	USART_RXDATAXP	RH	RX Buffer Data Extended Peek Register
0x034	USART_RXDOUBLEXP	RH	RX Buffer Double Data Extended Peek R...
0x038	USART_TXDATAX	W	TX Buffer Data Extended Register
0x03C	USART_TXDATA	W	TX Buffer Data Register
0x040	USART_TXDOUBLEX	W	TX Buffer Double Data Extended Register
0x044	USART_TXDOUBLE	W	TX Buffer Double Data Register
0x048	USART_IF	RWH INTFLAG	Interrupt Flag Register
0x04C	USART_IEN	RW	Interrupt Enable Register
0x050	USART_IRCTRL	RW	IrDA Control Register
0x054	USART_I2SCTRL	RW	I2S Control Register
0x058	USART_TIMING	RW	Timing Register
0x05C	USART_CTRLX	RW	Control Register Extended
0x060	USART_TIMECMP0	RW	Timer Compare 0
0x064	USART_TIMECMP1	RW	Timer Compare 1
0x068	USART_TIMECMP2	RW	Timer Compare 2
0x1000	USART_IPVERSION_SET	R	IPVERSION
0x1004	USART_EN_SET	RW	USART Enable
0x1008	USART_CTRL_SET	RW	Control Register
0x100C	USART_FRAME_SET	RW	USART Frame Format Register
0x1010	USART_TRIGCTRL_SET	RW	USART Trigger Control Register
0x1014	USART_CMD_SET	W	Command Register
0x1018	USART_STATUS_SET	RH	USART Status Register
0x101C	USART_CLKDIV_SET	RWH	Clock Control Register

Offset	Name	Type	Description
0x1020	USART_RXDATAX_SET	RH	RX Buffer Data Extended Register
0x1024	USART_RXDATA_SET	RH	RX Buffer Data Register
0x1028	USART_RXDOUBLEX_SET	RH	RX Buffer Double Data Extended Register
0x102C	USART_RXDOUBLE_SET	RH	RX FIFO Double Data Register
0x1030	USART_RXDATAXP_SET	RH	RX Buffer Data Extended Peek Register
0x1034	USART_RXDOUBLEXP_SET	RH	RX Buffer Double Data Extended Peek R...
0x1038	USART_TXDATAX_SET	W	TX Buffer Data Extended Register
0x103C	USART_TXDATA_SET	W	TX Buffer Data Register
0x1040	USART_TXDOUBLEX_SET	W	TX Buffer Double Data Extended Register
0x1044	USART_TXDOUBLE_SET	W	TX Buffer Double Data Register
0x1048	USART_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x104C	USART_IEN_SET	RW	Interrupt Enable Register
0x1050	USART_IRCTRL_SET	RW	IrDA Control Register
0x1054	USART_I2SCTRL_SET	RW	I2S Control Register
0x1058	USART_TIMING_SET	RW	Timing Register
0x105C	USART_CTRLX_SET	RW	Control Register Extended
0x1060	USART_TIMECMP0_SET	RW	Timer Compare 0
0x1064	USART_TIMECMP1_SET	RW	Timer Compare 1
0x1068	USART_TIMECMP2_SET	RW	Timer Compare 2
0x2000	USART_IPVERSION_CLR	R	IPVERSION
0x2004	USART_EN_CLR	RW	USART Enable
0x2008	USART_CTRL_CLR	RW	Control Register
0x200C	USART_FRAME_CLR	RW	USART Frame Format Register
0x2010	USART_TRIGCTRL_CLR	RW	USART Trigger Control Register
0x2014	USART_CMD_CLR	W	Command Register
0x2018	USART_STATUS_CLR	RH	USART Status Register
0x201C	USART_CLKDIV_CLR	RWH	Clock Control Register
0x2020	USART_RXDATAX_CLR	RH	RX Buffer Data Extended Register
0x2024	USART_RXDATA_CLR	RH	RX Buffer Data Register
0x2028	USART_RXDOUBLEX_CLR	RH	RX Buffer Double Data Extended Register
0x202C	USART_RXDOUBLE_CLR	RH	RX FIFO Double Data Register
0x2030	USART_RXDATAXP_CLR	RH	RX Buffer Data Extended Peek Register
0x2034	USART_RXDOUBLEXP_CLR	RH	RX Buffer Double Data Extended Peek R...
0x2038	USART_TXDATAX_CLR	W	TX Buffer Data Extended Register
0x203C	USART_TXDATA_CLR	W	TX Buffer Data Register
0x2040	USART_TXDOUBLEX_CLR	W	TX Buffer Double Data Extended Register
0x2044	USART_TXDOUBLE_CLR	W	TX Buffer Double Data Register

Offset	Name	Type	Description
0x2048	USART_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x204C	USART_IEN_CLR	RW	Interrupt Enable Register
0x2050	USART_IRCTRL_CLR	RW	IrDA Control Register
0x2054	USART_I2SCTRL_CLR	RW	I2S Control Register
0x2058	USART_TIMING_CLR	RW	Timing Register
0x205C	USART_CTRLX_CLR	RW	Control Register Extended
0x2060	USART_TIMECMP0_CLR	RW	Timer Compare 0
0x2064	USART_TIMECMP1_CLR	RW	Timer Compare 1
0x2068	USART_TIMECMP2_CLR	RW	Timer Compare 2
0x3000	USART_IPVERSION_TGL	R	IPVERSION
0x3004	USART_EN_TGL	RW	USART Enable
0x3008	USART_CTRL_TGL	RW	Control Register
0x300C	USART_FRAME_TGL	RW	USART Frame Format Register
0x3010	USART_TRIGCTRL_TGL	RW	USART Trigger Control Register
0x3014	USART_CMD_TGL	W	Command Register
0x3018	USART_STATUS_TGL	RH	USART Status Register
0x301C	USART_CLKDIV_TGL	RWH	Clock Control Register
0x3020	USART_RXDATAX_TGL	RH	RX Buffer Data Extended Register
0x3024	USART_RXDATA_TGL	RH	RX Buffer Data Register
0x3028	USART_RXDOUBLEX_TGL	RH	RX Buffer Double Data Extended Register
0x302C	USART_RXDOUBLE_TGL	RH	RX FIFO Double Data Register
0x3030	USART_RXDATAXP_TGL	RH	RX Buffer Data Extended Peek Register
0x3034	USART_RXDOUBLEXP_TGL	RH	RX Buffer Double Data Extended Peek R...
0x3038	USART_TXDATAX_TGL	W	TX Buffer Data Extended Register
0x303C	USART_TXDATA_TGL	W	TX Buffer Data Register
0x3040	USART_TXDOUBLEX_TGL	W	TX Buffer Double Data Extended Register
0x3044	USART_TXDOUBLE_TGL	W	TX Buffer Double Data Register
0x3048	USART_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x304C	USART_IEN_TGL	RW	Interrupt Enable Register
0x3050	USART_IRCTRL_TGL	RW	IrDA Control Register
0x3054	USART_I2SCTRL_TGL	RW	I2S Control Register
0x3058	USART_TIMING_TGL	RW	Timing Register
0x305C	USART_CTRLX_TGL	RW	Control Register Extended
0x3060	USART_TIMECMP0_TGL	RW	Timer Compare 0
0x3064	USART_TIMECMP1_TGL	RW	Timer Compare 1
0x3068	USART_TIMECMP2_TGL	RW	Timer Compare 2

19.5 USART Register Description

19.5.1 USART_IPVERSION - IPVERSION

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x0	R	**IPVERSION**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

19.5.2 USART_EN - USART Enable

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																EN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	EN	0x0	RW	**USART Enable**
	The ENABLE bit enables the module.			

19.5.3 USART_CTRL - Control Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0	0x0			0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0		0x0		0x0	0x0	0x0	0x0	0x0
Access	RW	RW	RW	RW			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW		RW		RW	RW	RW	RW	RW
Name	SMSDELAY	MVDIS	AUTOTX	BYTESWAP			SSSEARLY	ERRSTX	ERRSRX	ERRSDMA	BIT8DV	SKIPPERRF	SCRETRANS	SCMODE	AUTOTRI	AUTOCS	CSINV	TXINV	RXINV	TXBIL	CSMA	MSBF	CLKPHA	CLKPOL		OVS		MPAB	MPM	CCEN	LOOPBK	SYNC

Bit	Name	Reset	Access	Description
31	SMSDELAY	0x0	RW	**Synchronous Main Sample Delay**
	Delay Synchronous Main interface sample point to the next setup edge to improve timing and allow communication at higher speeds			
30	MVDIS	0x0	RW	**Majority Vote Disable**
	Disable majority vote for 16x, 8x and 6x oversampling modes.			
29	AUTOTX	0x0	RW	**Always Transmit When RX Not Full**
	Transmits as long as RX is not full. If TX is empty, underflows are generated.			
28	BYTESWAP	0x0	RW	**Byteswap In Double Accesses**
	Set to switch the order of the bytes in double accesses.			

Value	Mode	Description
0	DISABLE	Normal byte order
1	ENABLE	Byte order swapped

27:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*

Bit	Name	Reset	Access	Description
25	SSSEARLY	0x0	RW	**Synchronous Secondary Setup Early**
	Setup data on sample edge in synchronous secondary interface mode to improve MOSI setup time			
24	ERRSTX	0x0	RW	**Disable TX On Error**
	When set, the transmitter is disabled on framing and parity errors (asynchronous mode only) in the receiver.			

Value	Mode	Description
0	DISABLE	Received framing and parity errors have no effect on transmitter
1	ENABLE	Received framing and parity errors disable the transmitter

Bit	Name	Reset	Access	Description
23	ERRSRX	0x0	RW	**Disable RX On Error**
	When set, the receiver is disabled on framing and parity errors (asynchronous mode only).			

Value	Mode	Description
0	DISABLE	Framing and parity errors have no effect on receiver
1	ENABLE	Framing and parity errors disable the receiver

Bit	Name	Reset	Access	Description
22	ERRSDMA	0x0	RW	**Halt DMA On Error**

When set, DMA requests will be cleared on framing and parity errors (asynchronous mode only).

Value	Mode	Description
0	DISABLE	Framing and parity errors have no effect on DMA requests from the USART
1	ENABLE	DMA requests from the USART are blocked while the PERR or FERR interrupt flags are set

Bit	Name	Reset	Access	Description
21	BIT8DV	0x0	RW	**Bit 8 Default Value**

The default value of the 9th bit. If 9-bit frames are used, and an 8-bit write operation is done, leaving the 9th bit unspecified, the 9th bit is set to the value of BIT8DV.

Bit	Name	Reset	Access	Description
20	SKIPPERRF	0x0	RW	**Skip Parity Error Frames**

When set, the receiver discards frames with parity errors (asynchronous mode only). The PERR interrupt flag is still set.

Bit	Name	Reset	Access	Description
19	SCRETRANS	0x0	RW	**SmartCard Retransmit**

When in SmartCard mode, a NACK'ed frame will be kept in the shift register and retransmitted if the transmitter is still enabled.

Bit	Name	Reset	Access	Description
18	SCMODE	0x0	RW	**SmartCard Mode**

Use this bit to enable or disable SmartCard mode.

Bit	Name	Reset	Access	Description
17	AUTOTRI	0x0	RW	**Automatic TX Tristate**

When enabled, TXTRI is set by hardware whenever the transmitter is idle, and TXTRI is cleared by hardware when transmission starts.

Value	Mode	Description
0	DISABLE	The output on U(S)n_TX when the transmitter is idle is defined by TXINV
1	ENABLE	U(S)n_TX is tristated whenever the transmitter is idle

Bit	Name	Reset	Access	Description
16	AUTOCS	0x0	RW	**Automatic Chip Select**

When enabled, the output on USn_CS will be activated one baud-period before transmission starts, and deactivated when transmission ends.

Bit	Name	Reset	Access	Description
15	CSINV	0x0	RW	**Chip Select Invert**

Default value is active low. This affects both the selection of external secondaries, as well as the selection of the microcontroller as a secondary interface.

Value	Mode	Description
0	DISABLE	Chip select is active low
1	ENABLE	Chip select is active high

Bit	Name	Reset	Access	Description
14	TXINV	0x0	RW	**Transmitter output Invert**

The output from the USART transmitter can optionally be inverted by setting this bit.

Value	Mode	Description
0	DISABLE	Output from the transmitter is passed unchanged to U(S)n_TX
1	ENABLE	Output from the transmitter is inverted before it is passed to U(S)n_TX

Bit	Name	Reset	Access	Description
13	RXINV	0x0	RW	**Receiver Input Invert**

Setting this bit will invert the input to the USART receiver.

Value	Mode	Description
0	DISABLE	Input is passed directly to the receiver
1	ENABLE	Input is inverted before it is passed to the receiver

Bit	Name	Reset	Access	Description
12	TXBIL	0x0	RW	**TX Buffer Interrupt Level**

Determines the interrupt and status level of the transmit buffer.

Value	Mode	Description
0	EMPTY	TXBL and the TXBL interrupt flag are set when the transmit buffer becomes empty. TXBL is cleared when the buffer becomes nonempty.
1	HALFFULL	TXBL and TXBLIF are set when the transmit buffer goes from full to half-full or empty. TXBL is cleared when the buffer becomes full.

Bit	Name	Reset	Access	Description
11	CSMA	0x0	RW	**Action On Chip Select In Main Mode**

This register determines the action to be performed when chip select is configured as an input and driven low while in main interface mode.

Value	Mode	Description
0	NOACTION	No action taken
1	GOTOSLAVEMODE	Go to secondary mode

Bit	Name	Reset	Access	Description
10	MSBF	0x0	RW	**Most Significant Bit First**

Decides whether data is sent with the least significant bit first, or the most significant bit first.

Value	Mode	Description
0	DISABLE	Data is sent with the least significant bit first
1	ENABLE	Data is sent with the most significant bit first

Bit	Name	Reset	Access	Description
9	CLKPHA	0x0	RW	**Clock Edge For Setup/Sample**

Determines where data is set-up and sampled according to the bus clock when in synchronous mode.

Value	Mode	Description
0	SAMPLELEADING	Data is sampled on the leading edge and set-up on the trailing edge of the bus clock in synchronous mode
1	SAMPLETRAILING	Data is set-up on the leading edge and sampled on the trailing edge of the bus clock in synchronous mode

Bit	Name	Reset	Access	Description
8	CLKPOL	0x0	RW	**Clock Polarity**

Determines the clock polarity of the bus clock used in synchronous mode.

Value	Mode	Description
0	IDLELOW	The bus clock used in synchronous mode has a low base value

Bit	Name	Reset	Access	Description
	1	IDLEHIGH		The bus clock used in synchronous mode has a high base value
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:5	OVS	0x0	RW	**Oversampling**
	Sets the number of clock periods in a UART bit-period. More clock cycles gives better robustness, while less clock cycles gives better performance.			
	Value	Mode		Description
	0	X16		Regular UART mode with 16X oversampling in asynchronous mode
	1	X8		Double speed with 8X oversampling in asynchronous mode
	2	X6		6X oversampling in asynchronous mode
	3	X4		Quadruple speed with 4X oversampling in asynchronous mode
4	MPAB	0x0	RW	**Multi-Processor Address-Bit**
	Defines the value of the multi-processor address bit. An incoming frame with its 9th bit equal to the value of this bit marks the frame as a multi-processor address frame.			
3	MPM	0x0	RW	**Multi-Processor Mode**
	Multi-processor mode uses the 9th bit of the USART frames to tell whether the frame is an address frame or a data frame.			
	Value	Mode		Description
	0	DISABLE		The 9th bit of incoming frames has no special function
	1	ENABLE		An incoming frame with the 9th bit equal to MPAB will be loaded into the receive buffer regardless of RXBLOCK and will result in the MPAB interrupt flag being set
2	CCEN	0x0	RW	**Collision Check Enable**
	Enables collision checking on data when operating in half duplex modus.			
	Value	Mode		Description
	0	DISABLE		Collision check is disabled
	1	ENABLE		Collision check is enabled. The receiver must be enabled for the check to be performed
1	LOOPBK	0x0	RW	**Loopback Enable**
	Allows the receiver to be connected directly to the USART transmitter for loopback and half duplex communication.			
	Value	Mode		Description
	0	DISABLE		The receiver is connected to and receives data from U(S)n_RX
	1	ENABLE		The receiver is connected to and receives data from U(S)n_TX
0	SYNC	0x0	RW	**USART Synchronous Mode**
	Determines whether the USART is operating in asynchronous or synchronous mode.			
	Value	Mode		Description

Bit	Name	Reset	Access	Description
	0	DISABLE		The USART operates in asynchronous mode
	1	ENABLE		The USART operates in synchronous mode

19.5.4 USART_FRAME - USART Frame Format Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																			0x1				0x0						0x5			
Access																			RW				RW						RW			
Name																			STOPBITS				PARITY						DATABITS			

Bit	Name	Reset	Access	Description
31:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:12	STOPBITS	0x1	RW	**Stop-Bit Mode**

Determines the number of stop-bits used.

Value	Mode	Description
0	HALF	The transmitter generates a half stop bit. Stop-bits are not verified by receiver
1	ONE	One stop bit is generated and verified
2	ONEANDAHALF	The transmitter generates one and a half stop bit. The receiver verifies the first stop bit
3	TWO	The transmitter generates two stop bits. The receiver checks the first stop-bit only

11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	PARITY	0x0	RW	**Parity-Bit Mode**

Determines whether parity bits are enabled, and whether even or odd parity should be used. Only available in asynchronous mode.

Value	Mode	Description
0	NONE	Parity bits are not used
2	EVEN	Even parity are used. Parity bits are automatically generated and checked by hardware.
3	ODD	Odd parity is used. Parity bits are automatically generated and checked by hardware.

7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	DATABITS	0x5	RW	**Data-Bit Mode**

This register sets the number of data bits in a USART frame.

Value	Mode	Description
1	FOUR	Each frame contains 4 data bits
2	FIVE	Each frame contains 5 data bits

Bit	Name	Reset	Access	Description
	3	SIX		Each frame contains 6 data bits
	4	SEVEN		Each frame contains 7 data bits
	5	EIGHT		Each frame contains 8 data bits
	6	NINE		Each frame contains 9 data bits
	7	TEN		Each frame contains 10 data bits
	8	ELEVEN		Each frame contains 11 data bits
	9	TWELVE		Each frame contains 12 data bits
	10	THIRTEEN		Each frame contains 13 data bits
	11	FOURTEEN		Each frame contains 14 data bits
	12	FIFTEEN		Each frame contains 15 data bits
	13	SIXTEEN		Each frame contains 16 data bits

19.5.5 USART_TRIGCTRL - USART Trigger Control Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																				0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0				
Access																				RW	RW	RW	RW	RW	RW	RW	RW	RW				
Name																				RXATX2EN	RXATX1EN	RXATX0EN	TXARX2EN	TXARX1EN	TXARX0EN	AUTOTXTEN	TXTEN	RXTEN				

Bit	Name	Reset	Access	Description
31:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	RXATX2EN	0x0	RW	**Enable Receive Trigger after TX end of f**
	When set, a TX end of frame will trigger the receiver after a TCMPVAL2 baud-time delay			
11	RXATX1EN	0x0	RW	**Enable Receive Trigger after TX end of f**
	When set, a TX end of frame will trigger the receiver after a TCMPVAL1 baud-time delay			
10	RXATX0EN	0x0	RW	**Enable Receive Trigger after TX end of f**
	When set, a TX end of frame will trigger the receiver after a TCMPVAL0 baud-time delay			
9	TXARX2EN	0x0	RW	**Enable Transmit Trigger after RX End of**
	When set, an RX end of frame will trigger the transmitter after TCMP2VAL bit times to force a minimum response delay			
8	TXARX1EN	0x0	RW	**Enable Transmit Trigger after RX End of**
	When set, an RX end of frame will trigger the transmitter after TCMP1VAL bit times to force a minimum response delay			
7	TXARX0EN	0x0	RW	**Enable Transmit Trigger after RX End of**
	When set, an RX end of frame will trigger the transmitter after TCMP0VAL bit times to force a minimum response delay			
6	AUTOTXTEN	0x0	RW	**AUTOTX Trigger Enable**
	When set, AUTOTX is enabled as long as the PRS channel selected by TSEL has a high value			
5	TXTEN	0x0	RW	**Transmit Trigger Enable**
	When set, the PRS channel selected by TSEL sets TXEN, enabling the transmitter on positive trigger edges.			
4	RXTEN	0x0	RW	**Receive Trigger Enable**
	When set, the PRS channel selected by TSEL sets RXEN, enabling the receiver on positive trigger edges.			
3:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

19.5.6 USART_CMD - Command Register

Offset	Bit Position																											
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																					W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)
Name																					CLEARRX	CLEARTX	TXTRIDIS	TXTRIEN	RXBLOCKDIS	RXBLOCKEN	MASTERDIS	MASTEREN	TXDIS	TXEN	RXDIS	RXEN

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	CLEARRX	0x0	W(nB)	**Clear RX**
	Set to clear receive buffer and the RX shift register.			
10	CLEARTX	0x0	W(nB)	**Clear TX**
	Set to clear transmit buffer and the TX shift register.			
9	TXTRIDIS	0x0	W(nB)	**Transmitter Tristate Disable**
	Disables tristating of the transmitter output.			
8	TXTRIEN	0x0	W(nB)	**Transmitter Tristate Enable**
	Tristates the transmitter output.			
7	RXBLOCKDIS	0x0	W(nB)	**Receiver Block Disable**
	Set to clear RXBLOCK, resulting in all incoming frames being loaded into the receive buffer.			
6	RXBLOCKEN	0x0	W(nB)	**Receiver Block Enable**
	Set to set RXBLOCK, resulting in all incoming frames being discarded.			
5	MASTERDIS	0x0	W(nB)	**Main Mode Disable**
	Set to disable main interface mode, clearing the MASTER status bit and putting the USART in secondary interface mode.			
4	MASTEREN	0x0	W(nB)	**Main Mode Enable**
	Set to enable main interface mode, setting the MASTER status bit. Main mode should not be enabled while TXENS is set to 1. To enable both main interface and TX mode, write MASTEREN before TXEN, or enable them both in the same write operation.			
3	TXDIS	0x0	W(nB)	**Transmitter Disable**
	Set to disable transmission.			
2	TXEN	0x0	W(nB)	**Transmitter Enable**
	Set to enable data transmission.			
1	RXDIS	0x0	W(nB)	**Receiver Disable**
	Set to disable data reception. If a frame is under reception when the receiver is disabled, the incoming frame is discarded.			
0	RXEN	0x0	W(nB)	**Receiver Enable**

Bit	Name	Reset	Access	Description
	Set to activate data reception on U(S)n_RX.			

19.5.7 USART_STATUS - USART Status Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0			0x0	0x1	0x0	0x0	0x0	0x0	0x0	0x0	0x1	0x0	0x0	0x0	0x0	0x0	0x0
Access															R			R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Name															TXBUFCNT			TIMERRESTARTED	TXIDLE	RXFULLRIGHT	RXDATAVRIGHT	TXBSRIGHT	TXBDRIGHT	RXFULL	RXDATAV	TXBL	TXC	TXTRI	RXBLOCK	MASTER	TXENS	RXENS

Bit	Name	Reset	Access	Description
31:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17:16	TXBUFCNT	0x0	R	**TX Buffer Count**
	Count of TX buffer entry 0, entry 1, and TX shift register. For large frames, the count is only of TX buffer entry 0 and the TX shifter register.			
15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14	TIMERRESTARTED	0x0	R	**The USART Timer restarted itself**
	When the timer is restarting itself on each TCMP event, a TIMERRESTARTED value of 0x0 indicates the first TCMP event in the sequence of multiple TCMP events. Any non TCMP timer start events will clear TIMERRESTARTED. When there is a TCMP interrupt and TIMERRESTARTED is 0x0, an interrupt service routine can set a TCMP event counter variable in memory to 0x1 to indicate the first TCMP interrupt of the sequence.			
13	TXIDLE	0x1	R	**TX Idle**
	Set when TX idle			
12	RXFULLRIGHT	0x0	R	**RX Full of Right Data**
	When set, the entire RX buffer contains right data. Only used in I2S mode			
11	RXDATAVRIGHT	0x0	R	**RX Data Right**
	When set, reading RXDATA or RXDATAX gives right data. Else left data is read. Only used in I2S mode			
10	TXBSRIGHT	0x0	R	**TX Buffer Expects Single Right Data**
	When set, the TX buffer expects at least a single right data. Else it expects left data. Only used in I2S mode			
9	TXBDRIGHT	0x0	R	**TX Buffer Expects Double Right Data**
	When set, the TX buffer expects double right data. Else it may expect a single right data or left data. Only used in I2S mode			
8	RXFULL	0x0	R	**RX FIFO Full**
	Set when the RXFIFO is full. Cleared when the receive buffer is no longer full. When this bit is set, there is still room for one more frame in the receive shift register.			
7	RXDATAV	0x0	R	**RX Data Valid**
	Set when data is available in the receive buffer. Cleared when the receive buffer is empty.			
6	TXBL	0x1	R	**TX Buffer Level**

Bit	Name	Reset	Access	Description
	Indicates the level of the transmit buffer. If TXBIL is 0x0, TXBL is set whenever the transmit buffer is completely empty. Otherwise TXBL is set whenever the TX Buffer becomes half full.			
5	TXC	0x0	R	**TX Complete**
	Set when a transmission has completed and no more data is available in the transmit buffer and shift register. Cleared when data is written to the transmit buffer.			
4	TXTRI	0x0	R	**Transmitter Tristated**
	Set when the transmitter is tristated, and cleared when transmitter output is enabled. If AUTOTRI in USARTn_CTRL is set this bit is always read as 0.			
3	RXBLOCK	0x0	R	**Block Incoming Data**
	When set, the receiver discards incoming frames. An incoming frame will not be loaded into the receive buffer if this bit is set at the instant the frame has been completely received.			
2	MASTER	0x0	R	**SPI Main Mode**
	Set when the USART operates as a main interface. Set using the MASTEREN command and clear using the MASTERDIS command.			
1	TXENS	0x0	R	**Transmitter Enable Status**
	Set when the transmitter is enabled.			
0	RXENS	0x0	R	**Receiver Enable Status**
	Set when the receiver is enabled.			

19.5.8 USART_CLKDIV - Clock Control Register

Offset	Bit Position																															
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																		0x0													
Access	RW																		RW													
Name	AUTOBAUDEN																		DIV													

Bit	Name	Reset	Access	Description
31	AUTOBAUDEN	0x0	RW	**AUTOBAUD detection enable**
	Detects the baud rate based on receiving a 0x55 frame (0x00 for IrDA). This is used in Asynchronous mode.			
30:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:3	DIV	0x0	RW	**Fractional Clock Divider**
	Specifies the fractional clock divider for the USART. Setting AUTOBAUDEN in USARTn_CLKDIV will overwrite the DIV field.			
2:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

19.5.9 USART_RXDATAX - RX Buffer Data Extended Register

Offset	Bit Position																															
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0	0x0						0x0								
Access																	R	R						R								
Name																	FERR	PERR						RXDATA								

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15	FERR	0x0	R	**Data Framing Error**
	Set if data in buffer has a framing error. Can be the result of a break condition.			
14	PERR	0x0	R	**Data Parity Error**
	Set if data in buffer has a parity error (asynchronous mode only).			
13:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	RXDATA	0x0	R	**RX Data**
	Use this register to access data read from the USART. Buffer is cleared on read access.			

19.5.10 USART_RXDATA - RX Buffer Data Register

Offset	Bit Position																															
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									RXDATA							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	RXDATA	0x0	R	**RX Data**
	Use this register to access data read from USART. Buffer is cleared on read access. Only the 8 LSB can be read using this register.			

19.5.11 USART_RXDOUBLEX - RX Buffer Double Data Extended Register

Offset	Bit Position																														
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0										0x0					0x0	0x0										0x0				
Access	R	R										R					R	R										R				
Name	FERR1	PERR1										RXDATA1					FERR0	PERR0										RXDATA0				

Bit	Name	Reset	Access	Description
31	FERR1	0x0	R	**Data Framing Error 1**
	Set if data in buffer has a framing error. Can be the result of a break condition.			
30	PERR1	0x0	R	**Data Parity Error 1**
	Set if data in buffer has a parity error (asynchronous mode only).			
29:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24:16	RXDATA1	0x0	R	**RX Data 1**
	Second frame read from buffer.			
15	FERR0	0x0	R	**Data Framing Error 0**
	Set if data in buffer has a framing error. Can be the result of a break condition.			
14	PERR0	0x0	R	**Data Parity Error 0**
	Set if data in buffer has a parity error (asynchronous mode only).			
13:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	RXDATA0	0x0	R	**RX Data 0**
	First frame read from buffer.			

19.5.12 USART_RXDOUBLE - RX FIFO Double Data Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0								0x0							
Access																	R								R							
Name																	RXDATA1								RXDATA0							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:8	RXDATA1	0x0	R	**RX Data 1**
	Second frame read from buffer.			
7:0	RXDATA0	0x0	R	**RX Data 0**
	First frame read from buffer.			

19.5.13 USART_RXDATAXP - RX Buffer Data Extended Peek Register

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0	0x0							0x0							
Access																	R	R							R							
Name																	FERRP	PERRP							RXDATAP							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15	FERRP	0x0	R	**Data Framing Error Peek**
	Set if data in buffer has a framing error. Can be the result of a break condition.			
14	PERRP	0x0	R	**Data Parity Error Peek**
	Set if data in buffer has a parity error (asynchronous mode only).			
13:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	RXDATAP	0x0	R	**RX Data Peek**
	Use this register to access data read from the USART.			

19.5.14 USART_RXDOUBLEXP - RX Buffer Double Data Extended Peek R...

Offset	Bit Position																														
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0						0x0									0x0	0x0						0x0								
Access	R	R						R									R	R						R								
Name	FERRP1	PERRP1						RXDATAP1									FERRP0	PERRP0						RXDATAP0								

Bit	Name	Reset	Access	Description
31	FERRP1	0x0	R	**Data Framing Error 1 Peek**
	Set if data in buffer has a framing error. Can be the result of a break condition.			
30	PERRP1	0x0	R	**Data Parity Error 1 Peek**
	Set if data in buffer has a parity error (asynchronous mode only).			
29:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24:16	RXDATAP1	0x0	R	**RX Data 1 Peek**
	Second frame read from FIFO.			
15	FERRP0	0x0	R	**Data Framing Error 0 Peek**
	Set if data in buffer has a framing error. Can be the result of a break condition.			
14	PERRP0	0x0	R	**Data Parity Error 0 Peek**
	Set if data in buffer has a parity error (asynchronous mode only).			
13:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	RXDATAP0	0x0	R	**RX Data 0 Peek**
	First frame read from FIFO.			

19.5.15 USART_TXDATAX - TX Buffer Data Extended Register

Offset	Bit Position																															
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0	0x0	0x0	0x0	0x0			0x0								
Access																	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)			W(nB)								
Name																	RXENAT	TXDISAT	TXBREAK	TXTRIAT	UBRXAT			TXDATAX								

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15	RXENAT	0x0	W(nB)	**Enable RX After Transmission**
	Set to enable reception after transmission.			
14	TXDISAT	0x0	W(nB)	**Clear TXEN After Transmission**
	Set to disable transmitter and release data bus directly after transmission.			
13	TXBREAK	0x0	W(nB)	**Transmit Data As Break**
	Set to send data as a break. Recipient will see a framing error or a break condition depending on its configuration and the value of TXDATA.			
12	TXTRIAT	0x0	W(nB)	**Set TXTRI After Transmission**
	Set to tristate transmitter by setting TXTRI after transmission.			
11	UBRXAT	0x0	W(nB)	**Unblock RX After Transmission**
	Set to clear RXBLOCK after transmission, unblocking the receiver.			
10:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	TXDATAX	0x0	W(nB)	**TX Data**
	Use this register to write data to the USART. If TXEN is set, a transfer will be initiated at the first opportunity.			

19.5.16 USART_TXDATA - TX Buffer Data Register

Offset	Bit Position																													
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W(nB)							
Name																									TXDATA							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	TXDATA	0x0	W(nB)	**TX Data**
	This frame will be added to TX buffer. Only 8 LSB can be written using this register. 9th bit and control bits will be cleared.			

19.5.17 USART_TXDOUBLEX - TX Buffer Double Data Extended Register

Offset	Bit Position																														
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0	0x0	0x0			0x0									0x0	0x0	0x0	0x0	0x0			0x0								
Access	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)			W(nB)									W(nB)	W(nB)	W(nB)	W(nB)	W(nB)			W(nB)								
Name	RXENAT1	TXDISAT1	TXBREAK1	TXTRIAT1	UBRXAT1			TXDATA1									RXENAT0	TXDISAT0	TXBREAK0	TXTRIAT0	UBRXAT0			TXDATA0								

Bit	Name	Reset	Access	Description
31	RXENAT1	0x0	W(nB)	**Enable RX After Transmission**
	Set to enable reception after transmission.			
30	TXDISAT1	0x0	W(nB)	**Clear TXEN After Transmission**
	Set to disable transmitter and release data bus directly after transmission.			
29	TXBREAK1	0x0	W(nB)	**Transmit Data As Break**
	Set to send data as a break. Recipient will see a framing error or a break condition depending on its configuration and the value of USARTn_TXDATA.			
28	TXTRIAT1	0x0	W(nB)	**Set TXTRI After Transmission**
	Set to tristate transmitter by setting TXTRI after transmission.			
27	UBRXAT1	0x0	W(nB)	**Unblock RX After Transmission**
	Set clear RXBLOCK after transmission, unblocking the receiver.			
26:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24:16	TXDATA1	0x0	W(nB)	**TX Data**
	Second frame to write to FIFO.			
15	RXENAT0	0x0	W(nB)	**Enable RX After Transmission**
	Set to enable reception after transmission.			
14	TXDISAT0	0x0	W(nB)	**Clear TXEN After Transmission**
	Set to disable transmitter and release data bus directly after transmission.			
13	TXBREAK0	0x0	W(nB)	**Transmit Data As Break**
	Set to send data as a break. Recipient will see a framing error or a break condition depending on its configuration and the value of TXDATA.			
12	TXTRIAT0	0x0	W(nB)	**Set TXTRI After Transmission**
	Set to tristate transmitter by setting TXTRI after transmission.			
11	UBRXAT0	0x0	W(nB)	**Unblock RX After Transmission**
	Set clear RXBLOCK after transmission, unblocking the receiver.			
10:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	TXDATA0	0x0	W(nB)	**TX Data**

Bit	Name	Reset	Access	Description
	First frame to write to buffer.			

19.5.18 USART_TXDOUBLE - TX Buffer Double Data Register

Offset	Bit Position																													
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0								0x0							
Access																	W								W							
Name																	TXDATA1								TXDATA0							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:8	TXDATA1	0x0	W	**TX Data**
	Second frame to write to buffer.			
7:0	TXDATA0	0x0	W	**TX Data**
	First frame to write to buffer.			

19.5.19 USART_IF - Interrupt Flag Register

Offset	Bit Position																															
0x048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x1	0x0
Access																RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name																TCMP2	TCMP1	TCMP0	TXIDLE	CCF	SSM	MPAF	FERR	PERR	TXUF	TXOF	RXUF	RXOF	RXFULL	RXDATAV	TXBL	TXC

Bit	Name	Reset	Access	Description
31:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	TCMP2	0x0	RW	**Timer comparator 2 Interrupt Flag**
	Set when the timer reaches the comparator 2 value, TCMP2.			
15	TCMP1	0x0	RW	**Timer comparator 1 Interrupt Flag**
	Set when the timer reaches the comparator 1 value, TCMP1.			
14	TCMP0	0x0	RW	**Timer comparator 0 Interrupt Flag**
	Set when the Timer reaches the comparator 0 value, TCMP0.			
13	TXIDLE	0x0	RW	**TX Idle Interrupt Flag**
	Set when TX goes idle. At this point, transmission has ended			
12	CCF	0x0	RW	**Collision Check Fail Interrupt Flag**
	Set when a collision check notices an error in the transmitted data.			
11	SSM	0x0	RW	**Chip-Select In Main Mode Interrupt Flag**
	Set when the chip select is pulled active when in main interface mode.			
10	MPAF	0x0	RW	**Multi-Processor Address Frame Interrupt**
	Set when a multi-processor address frame is detected.			
9	FERR	0x0	RW	**Framing Error Interrupt Flag**
	Set when a frame with a framing error is received while RXBLOCK is cleared.			
8	PERR	0x0	RW	**Parity Error Interrupt Flag**
	Set when a frame with a parity error (asynchronous mode only) is received while RXBLOCK is cleared.			
7	TXUF	0x0	RW	**TX Underflow Interrupt Flag**
	Set when operating as a synchronous secondary, no data is available in the transmit buffer when the main interface starts transmission of a new frame.			
6	TXOF	0x0	RW	**TX Overflow Interrupt Flag**
	Set when a write is done to the transmit buffer while it is full. The data already in the transmit buffer is preserved.			
5	RXUF	0x0	RW	**RX Underflow Interrupt Flag**
	Set when trying to read from the receive buffer when it is empty.			
4	RXOF	0x0	RW	**RX Overflow Interrupt Flag**
	Set when data is incoming while the receive shift register is full. The data previously in the shift register is lost.			

Bit	Name	Reset	Access	Description
3	RXFULL	0x0	RW	**RX Buffer Full Interrupt Flag**
	Set when the receive buffer becomes full.			
2	RXDATAV	0x0	RW	**RX Data Valid Interrupt Flag**
	Set when data becomes available in the receive buffer.			
1	TXBL	0x1	RW	**TX Buffer Level Interrupt Flag**
	Set when buffer becomes empty if buffer level is set to 0x0, or when the number of empty TX buffer elements equals specified buffer level.			
0	TXC	0x0	RW	**TX Complete Interrupt Flag**
	This interrupt is set after a transmission when both the TX buffer and shift register are empty.			

19.5.20 USART_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x04C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name																TCMP2	TCMP1	TCMP0	TXIDLE	CCF	SSM	MPAF	FERR	PERR	TXUF	TXOF	RXUF	RXOF	RXFULL	RXDATAV	TXBL	TXC

Bit	Name	Reset	Access	Description
31:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	TCMP2	0x0	RW	**Timer comparator 2 Interrupt Enable**
	Set when the timer reaches the comparator 2 value, TCMP2.			
15	TCMP1	0x0	RW	**Timer comparator 1 Interrupt Enable**
	Set when the timer reaches the comparator 1 value, TCMP1.			
14	TCMP0	0x0	RW	**Timer comparator 0 Interrupt Enable**
	Set when the Timer reaches the comparator 0 value, TCMP0.			
13	TXIDLE	0x0	RW	**TX Idle Interrupt Enable**
	Set when TX goes idle. At this point, transmission has ended			
12	CCF	0x0	RW	**Collision Check Fail Interrupt Enable**
	Set when a collision check notices an error in the transmitted data.			
11	SSM	0x0	RW	**Chip-Select In Main Mode Interrupt Flag**
	Set when the chip select is pulled active when in main interface mode.			
10	MPAF	0x0	RW	**Multi-Processor Address Frame Interrupt**
	Set when a multi-processor address frame is detected.			
9	FERR	0x0	RW	**Framing Error Interrupt Enable**
	Set when a frame with a framing error is received while RXBLOCK is cleared.			
8	PERR	0x0	RW	**Parity Error Interrupt Enable**
	Set when a frame with a parity error (asynchronous mode only) is received while RXBLOCK is cleared.			
7	TXUF	0x0	RW	**TX Underflow Interrupt Enable**
	Set when operating as a synchronous secondary, no data is available in the transmit buffer when the main interface starts transmission of a new frame.			
6	TXOF	0x0	RW	**TX Overflow Interrupt Enable**
	Set when a write is done to the transmit buffer while it is full. The data already in the transmit buffer is preserved.			
5	RXUF	0x0	RW	**RX Underflow Interrupt Enable**
	Set when trying to read from the receive buffer when it is empty.			
4	RXOF	0x0	RW	**RX Overflow Interrupt Enable**
	Set when data is incoming while the receive shift register is full. The data previously in the shift register is lost.			

Bit	Name	Reset	Access	Description
3	RXFULL	0x0	RW	**RX Buffer Full Interrupt Enable**
	Set when the receive buffer becomes full.			
2	RXDATAV	0x0	RW	**RX Data Valid Interrupt Enable**
	Set when data becomes available in the receive buffer.			
1	TXBL	0x0	RW	**TX Buffer Level Interrupt Enable**
	Set when buffer becomes empty if buffer level is set to 0x0, or when the number of empty TX buffer elements equals specified buffer level.			
0	TXC	0x0	RW	**TX Complete Interrupt Enable**
	This interrupt is set after a transmission when both the TX buffer and shift register are empty.			

19.5.21 USART_IRCTRL - IrDA Control Register

Offset	Bit Position																															
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0		0x0
Access																													RW	RW		RW
Name																													IRFILT	IRPW		IREN

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	IRFILT	0x0	RW	**IrDA RX Filter**
	Set to enable filter on IrDA demodulator.			

Value	Mode	Description
0	DISABLE	No filter enabled
1	ENABLE	Filter enabled. IrDA pulse must be high for at least 5 consecutive clock cycles to be detected

Bit	Name	Reset	Access	Description
2:1	IRPW	0x0	RW	**IrDA TX Pulse Width**
	Configure the pulse width generated by the IrDA modulator as a fraction of the configured USART bit period.			

Value	Mode	Description
0	ONE	IrDA pulse width is 1/16 for OVS=0 and 1/8 for OVS=1
1	TWO	IrDA pulse width is 2/16 for OVS=0 and 2/8 for OVS=1
2	THREE	IrDA pulse width is 3/16 for OVS=0 and 3/8 for OVS=1
3	FOUR	IrDA pulse width is 4/16 for OVS=0 and 4/8 for OVS=1

Bit	Name	Reset	Access	Description
0	IREN	0x0	RW	**Enable IrDA Module**
	Enable IrDA module and rout USART signals through it.			

19.5.22 USART_I2SCTRL - I2S Control Register

Offset	Bit Position																													
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0					0x0	0x0	0x0	0x0	0x0
Access																							RW					RW	RW	RW	RW	RW
Name																							FORMAT					DELAY	DMASPLIT	JUSTIFY	MONO	EN

Bit	Name	Reset	Access	Description
31:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10:8	FORMAT	0x0	RW	**I2S Word Format**
	Configure the data-width used internally for I2S data			

Value	Mode	Description
0	W32D32	32-bit word, 32-bit data
1	W32D24M	32-bit word, 32-bit data with 8 lsb masked
2	W32D24	32-bit word, 24-bit data
3	W32D16	32-bit word, 16-bit data
4	W32D8	32-bit word, 8-bit data
5	W16D16	16-bit word, 16-bit data
6	W16D8	16-bit word, 8-bit data
7	W8D8	8-bit word, 8-bit data

7:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	DELAY	0x0	RW	**Delay on I2S data**
	Set to add a one-cycle delay between a transition on the word-clock and the start of the I2S word. Should be set for standard I2S format			
3	DMASPLIT	0x0	RW	**Separate DMA Request For Left/Right Data**
	When set DMA requests for right-channel data are put on the TXBLRIGHT and RXDATAVRIGHT DMA requests.			
2	JUSTIFY	0x0	RW	**Justification of I2S Data**
	Determines whether the I2S data is left or right justified			

Value	Mode	Description
0	LEFT	Data is left-justified
1	RIGHT	Data is right-justified

1	MONO	0x0	RW	**Stero or Mono**
	Switch between stereo and mono mode. Set for mono			
0	EN	0x0	RW	**Enable I2S Mode**

Bit	Name	Reset	Access	Description
	Set the U(S)ART in I2S mode.			

19.5.23 USART_TIMING - Timing Register

Offset	Bit Position																															
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0x0				0x0				0x0				0x0																		
Access		RW				RW				RW				RW																		
Name		CSHOLD				ICS				CSSETUP				TXDELAY																		

Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
30:28	CSHOLD	0x0	RW	**Chip Select Hold**

Chip Select will be asserted after the end of frame transmission. When using TCMPn, normally set TIMECMPn_TSTART to DISABLE to stop general timer and to prevent unwanted interrupts.

Value	Mode	Description
0	ZERO	Disable CS being asserted after the end of transmission
1	ONE	CS is asserted for 1 baud-times after the end of transmission
2	TWO	CS is asserted for 2 baud-times after the end of transmission
3	THREE	CS is asserted for 3 baud-times after the end of transmission
4	SEVEN	CS is asserted for 7 baud-times after the end of transmission
5	TCMP0	CS is asserted after the end of transmission for TCMPVAL0 baud-times
6	TCMP1	CS is asserted after the end of transmission for TCMPVAL1 baud-times
7	TCMP2	CS is asserted after the end of transmission for TCMPVAL2 baud-times

27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26:24	ICS	0x0	RW	**Inter-character spacing**

Inter-character spacing after each TX frame while the TX buffer is not empty. When using USART_TIMECMPn, normally set TSTART to DISABLE to stop general timer and to prevent unwanted interrupts.

Value	Mode	Description
0	ZERO	There is no space between charcters
1	ONE	Create a space of 1 baud-times before start of transmission
2	TWO	Create a space of 2 baud-times before start of transmission
3	THREE	Create a space of 3 baud-times before start of transmission
4	SEVEN	Create a space of 7 baud-times before start of transmission
5	TCMP0	Create a space of before the start of transmission for TCMPVAL0 baud-times

Bit	Name	Reset	Access	Description
	6	TCMP1		Create a space of before the start of transmission for TCMPVAL1 baud-times
	7	TCMP2		Create a space of before the start of transmission for TCMPVAL2 baud-times
23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:20	CSSETUP	0x0	RW	**Chip Select Setup**
	Chip Select will be asserted before the start of frame transmission. When using USART_TIMECMPn, normally set TSTART to DISABLE to stop general timer and to prevent unwanted interrupts.			
	Value	Mode		Description
	0	ZERO		CS is not asserted before start of transmission
	1	ONE		CS is asserted for 1 baud-times before start of transmission
	2	TWO		CS is asserted for 2 baud-times before start of transmission
	3	THREE		CS is asserted for 3 baud-times before start of transmission
	4	SEVEN		CS is asserted for 7 baud-times before start of transmission
	5	TCMP0		CS is asserted before the start of transmission for TCMPVAL0 baud-times
	6	TCMP1		CS is asserted before the start of transmission for TCMPVAL1 baud-times
	7	TCMP2		CS is asserted before the start of transmission for TCMPVAL2 baud-times
19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18:16	TXDELAY	0x0	RW	**TX frame start delay**
	Number of baud-times to delay the start of frame transmission. When using USART_TIMECMPn, normally set TSTART to DISABLE to stop general timer and to prevent unwanted interrupts.			
	Value	Mode		Description
	0	DISABLE		Disable - TXDELAY in USARTn_CTRL can be used for legacy
	1	ONE		Start of transmission is delayed for 1 baud-times
	2	TWO		Start of transmission is delayed for 2 baud-times
	3	THREE		Start of transmission is delayed for 3 baud-times
	4	SEVEN		Start of transmission is delayed for 7 baud-times
	5	TCMP0		Start of transmission is delayed for TCMPVAL0 baud-times
	6	TCMP1		Start of transmission is delayed for TCMPVAL1 baud-times
	7	TCMP2		Start of transmission is delayed for TCMPVAL2 baud-times
15:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

19.5.24 USART_CTRLX - Control Register Extended

Offset	Bit Position																													
0x05C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0								0x0				0x0	0x0	0x0	0x0
Access																	RW								RW				RW	RW	RW	RW
Name																	CLKPRSEN								RXPRSEN				RTSINV	CTSEN	CTSINV	DBGHALT

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15	CLKPRSEN	0x0	RW	**PRS CLK Enable**
	When set, the PRS channel selected as input to CLK.			
14:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	RXPRSEN	0x0	RW	**PRS RX Enable**
	When set, the PRS channel selected as input to RX.			
6:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	RTSINV	0x0	RW	**RTS Pin Inversion**
	When set, the RTS pin polarity is inverted.			

Value	Mode	Description
0	DISABLE	The USn_RTS pin is low true
1	ENABLE	The USn_RTS pin is high true

Bit	Name	Reset	Access	Description
2	CTSEN	0x0	RW	**CTS Function enabled**
	When set, frames in the TXBUFn will not be sent until link partner asserts CTS. Any data in the TX shift register will continue transmitting, the next TXBUFn data will not load into the TX shift register			

Value	Mode	Description
0	DISABLE	Ingore CTS
1	ENABLE	Stop transmitting when CTS is negated

Bit	Name	Reset	Access	Description
1	CTSINV	0x0	RW	**CTS Pin Inversion**
	When set, the CTS pin polarity is inverted.			

Value	Mode	Description
0	DISABLE	The USn_CTS pin is low true
1	ENABLE	The USn_CTS pin is high true

Bit	Name	Reset	Access	Description
0	DBGHALT	0x0	RW	**Debug halt**

Bit	Name	Reset	Access	Description
	.			

Value	Mode	Description
0	DISABLE	Continue to transmit until TX buffer is empty
1	ENABLE	Negate RTS to stop link partner's transmission during debug HALT. NOTE** The core clock should be equal to or faster than the peripheral clock; otherwise, each single step could transmit multiple frames instead of just transmitting one frame.

19.5.25 USART_TIMECMP0 - Timer Compare 0

Offset	Bit Position																													
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset								0x0		0x0				0x0											0x0							
Access								RW		RW				RW											RW							
Name								RESTARTEN		TSTOP				TSTART											TCMPVAL							

Bit	Name	Reset	Access	Description
31:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24	RESTARTEN	0x0	RW	**Restart Timer on TCMP0**
	Each TCMP0 event will reset and restart the timer			
	Value	Mode		Description
	0	DISABLE		Disable the timer restarting on TCMP0
	1	ENABLE		Enable the timer restarting on TCMP0
23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:20	TSTOP	0x0	RW	**Source used to disable comparator 0**
	Select the source which disables comparator 0			
	Value	Mode		Description
	0	TCMP0		Comparator 0 is disabled when the counter equals TCMPVAL and triggers a TCMP0 event
	1	TXST		Comparator 0 is disabled at TX start TX Engine
	2	RXACT		Comparator 0 is disabled on RX going going Active (default: low)
	3	RXACTN		Comparator 0 is disabled on RX going Inactive
19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18:16	TSTART	0x0	RW	**Timer start source**
	Source used to start comparator 0 and timer			
	Value	Mode		Description
	0	DISABLE		Comparator 0 is disabled
	1	TXEOF		Comparator 0 and timer are started at TX end of frame
	2	TXC		Comparator 0 and timer are started at TX Complete
	3	RXACT		Comparator 0 and timer are started at RX going going Active (default: low)

Bit	Name	Reset	Access	Description
	4	RXEOF		Comparator 0 and timer are started at RX end of frame
15:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	TCMPVAL	0x0	RW	**Timer comparator 0.** When the timer equals TCMPVAL, this signals a TCMP0 event and sets the TCMP0 flag. This event can also be used to enable various USART functionality. A value of 0x00 represents 256 baud times.

19.5.26 USART_TIMECMP1 - Timer Compare 1

Offset	Bit Position																															
0x064	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset								0x0		0x0				0x0											0x0							
Access								RW		RW				RW											RW							
Name								RESTARTEN		TSTOP				TSTART											TCMPVAL							

Bit	Name	Reset	Access	Description
31:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24	RESTARTEN	0x0	RW	**Restart Timer on TCMP1**
	Each TCMP1 event will reset and restart the timer			
	Value	Mode		Description
	0	DISABLE		Disable the timer restarting on TCMP1
	1	ENABLE		Enable the timer restarting on TCMP1
23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:20	TSTOP	0x0	RW	**Source used to disable comparator 1**
	Select the source which disables comparator 1			
	Value	Mode		Description
	0	TCMP1		Comparator 1 is disabled when the counter equals TCMPVAL and triggers a TCMP1 event
	1	TXST		Comparator 1 is disabled at TX start TX Engine
	2	RXACT		Comparator 1 is disabled on RX going going Active (default: low)
	3	RXACTN		Comparator 1 is disabled on RX going Inactive
19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18:16	TSTART	0x0	RW	**Timer start source**
	Source used to start comparator 1 and timer			
	Value	Mode		Description
	0	DISABLE		Comparator 1 is disabled
	1	TXEOF		Comparator 1 and timer are started at TX end of frame
	2	TXC		Comparator 1 and timer are started at TX Complete
	3	RXACT		Comparator 1 and timer are started at RX going going Active (default: low)

Bit	Name	Reset	Access	Description
	4	RXEOF		Comparator 1 and timer are started at RX end of frame
15:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	TCMPVAL	0x0	RW	**Timer comparator 1.** When the timer equals TCMPVAL, this signals a TCMP1 event and sets the TCMP1 flag. This event can also be used to enable various USART functionality. A value of 0x00 represents 256 baud times.

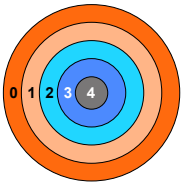
19.5.27 USART_TIMECMP2 - Timer Compare 2

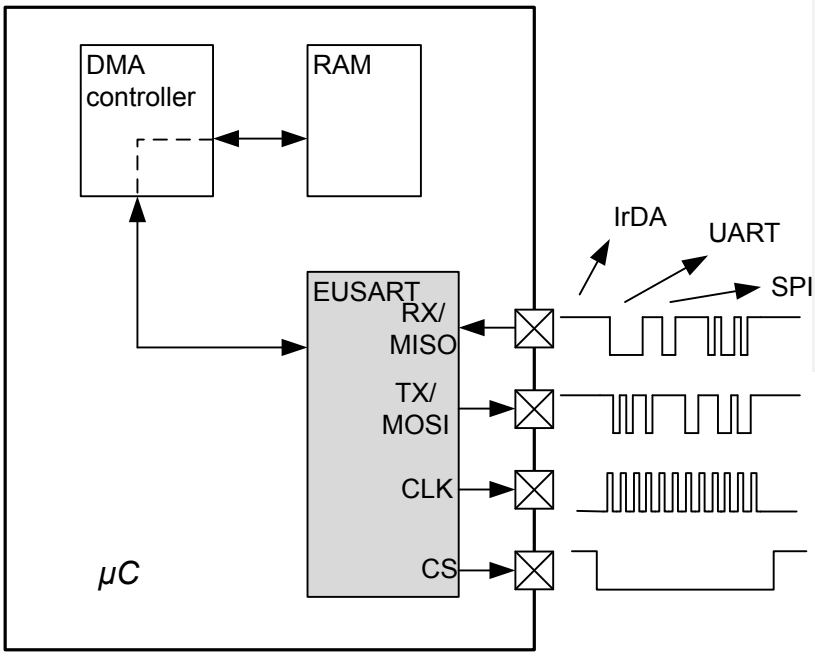
Offset	Bit Position																															
0x068	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset								0x0		0x0				0x0											0x0							
Access								RW		RW				RW											RW							
Name								RESTARTEN		TSTOP				TSTART											TCMPVAL							

Bit	Name	Reset	Access	Description
31:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24	RESTARTEN	0x0	RW	**Restart Timer on TCMP2**
	Each TCMP2 event will reset and restart the timer			
	Value	Mode		Description
	0	DISABLE		Disable the timer restarting on TCMP2
	1	ENABLE		Enable the timer restarting on TCMP2
23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:20	TSTOP	0x0	RW	**Source used to disable comparator 2**
	Select the source which disables comparator 2			
	Value	Mode		Description
	0	TCMP2		Comparator 2 is disabled when the counter equals TCMPVAL and triggers a TCMP2 event
	1	TXST		Comparator 2 is disabled at TX start TX Engine
	2	RXACT		Comparator 2 is disabled on RX going going Active (default: low)
	3	RXACTN		Comparator 2 is disabled on RX going Inactive
19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18:16	TSTART	0x0	RW	**Timer start source**
	Source used to start comparator 2 and timer			
	Value	Mode		Description
	0	DISABLE		Comparator 2 is disabled
	1	TXEOF		Comparator 2 and timer are started at TX end of frame
	2	TXC		Comparator 2 and timer are started at TX Complete
	3	RXACT		Comparator 2 and timer are started at RX going going Active (default: low)

Bit	Name	Reset	Access	Description
	4	RXEOF		Comparator 2 and timer are started at RX end of frame
15:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	TCMPVAL	0x0	RW	**Timer comparator 2.** When the timer equals TCMPVAL, this signals a TCMP2 event and sets the TCMP2 flag. This event can also be used to enable various USART functionality. A value of 0x00 represents 256 baud times.

20. EUSART - Universal Synchronous Asynchronous Receiver/Transmitter





Quick Facts

What?

The EUSART handles high-speed UART, SPI-bus, and IrDA communication.

Why?

Serial communication is frequently used in embedded systems and the EUSART allows efficient communication with a wide range of external devices.

How?

The EUSART has a wide selection of operating modes, frame formats and baud rates. The multiprocessor mode allows the EUSART to remain idle when not addressed. 16-deep FIFOs and DMA support makes high data-rates possible with minimal CPU intervention and it is possible to transmit and receive large frames while the MCU remains in EM1 Sleep.

20.1 Introduction

The Enhanced Universal Synchronous Asynchronous serial Receiver and Transmitter (EUSART) is a very flexible serial I/O module. It supports full duplex asynchronous UART communication as well as SPI, MicroWire. It can also interface with IrDA devices.

20.2 Features

- Asynchronous (UART) and synchronous (SPI) communication
- Full duplex and half duplex
- Separate TX/RX enable
- Separate receive / transmit 16 deep FIFOs, with additional separate shift registers
- Can operate using HF clock or LF clock (UART only)
- Programmable baud rate, generated as an fractional division from the peripheral clock ($HFPERCLK_{EUSARTn}$)
- Max bit-rate
  - Main SPI mode: 20 MHz
  - Secondary SPI mode: 10 MHz
  - UART mode, HF peripheral clock rate/16, 8, 6, or 4
  - UART mode, 32 KHz LF peripheral clock: 9600 (oversampling not supported with LF operation)
- Supports transmission and reception in EM0 (also EM1 and EM2 in certain modes) with
  - Full DMA support
  - UART mode, specified start-frame can start reception automatically
- UART mode, capable of sleep-mode wake-up on received frame (EM2 Capable instance only)
  - Either wake-up on any received byte or
  - Wake up only on specified start and signal frames
- SPI mode, capable of sleep-mode wake-up on received frame (EM2/3 Capable instance only)
  - Either wake-up on CS active or
  - Wake up on Receive FIFO level matching Watermark Level Setting
- Asynchronous mode supports
  - Configurable number of data bits 7-9 (plus the parity bit, if enabled)
  - HW parity bit generation and check
  - Majority vote baud-reception
  - False start-bit detection
  - Break generation/detection
  - Multi-processor mode
  - Configurable number of stop bits in asynchronous mode:
    - HF clock EM0/1 operation: 0.5, 1, 1.5, 2
    - LF clock operation: 1, 2
  - HW collision detection
  - IrDA support
    - HF clock EM0/1 operation: IrDA modulator
    - LF clock operation: Pulse extender, RX-only
  - Hardware Flow Control
  - Automatic Baud Rate Detection
- Synchronous mode supports
  - Configurable number of data bits 8-16
  - All 4 SPI clock polarity/phase configurations
  - Main and Secondary interface modes
- Data can be transmitted LSB first or MSB first
- Separate interrupt vectors for receive and transmit interrupts
- Loopback mode
  - Half duplex communication
  - Communication debugging
- PRS RX input
- DMA Support
- EM2 operation with LF clock (EM2/3 Capable instance only), wakeup to EM1 for DMA interaction
- Async mode Automatic Baud Rate Detection operating with HF clock in EM0/1

20.3 Functional Description

An overview of the EUSART module is shown in Figure 20.1 EUSART Overview on page 608.

This section describes all posible EUSART features. Please refer to the Device Datasheet to see what features a specific EUSART instance supports.

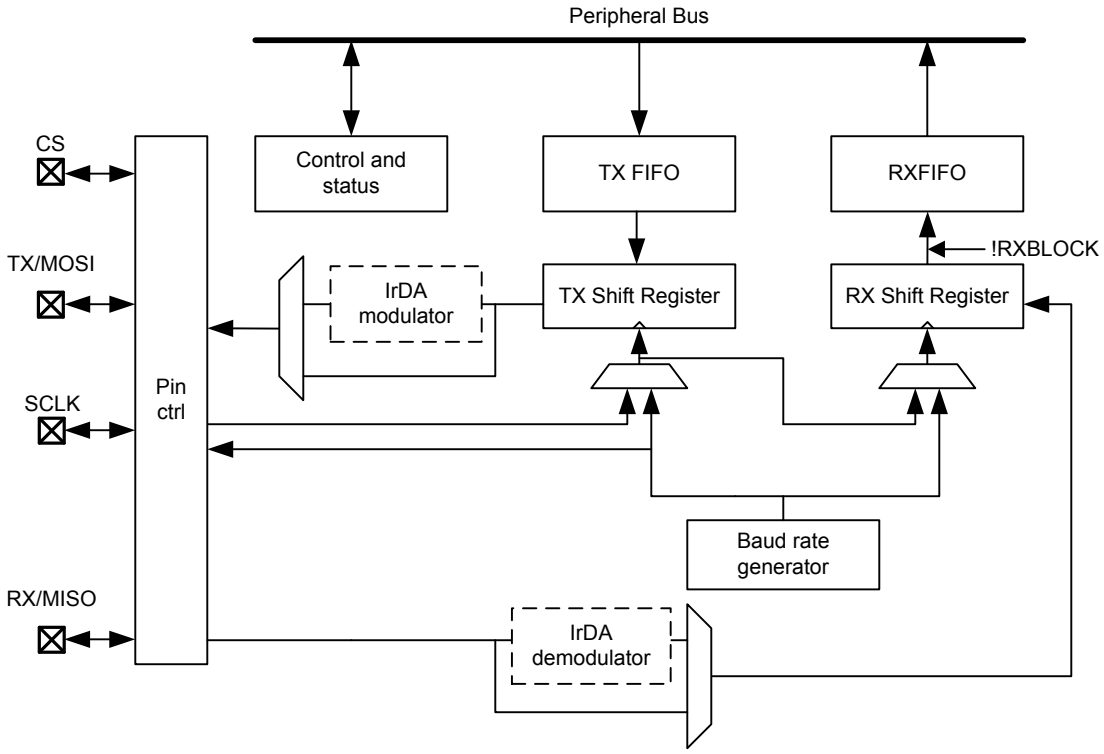


Figure 20.1. EUSART Overview

20.3.1 Modes of Operation

The EUSART operates in either asynchronous or synchronous mode.

In synchronous mode, a separate clock signal is transmitted with the data. This clock signal is generated by the main interface on the bus, and both the main and secondary devices sample and transmit data according to this clock. Both main and secondary interface modes are supported by the EUSART. The synchronous communication mode is compatible with the Serial Peripheral Interface Bus (SPI) standard.

In asynchronous mode, no separate clock signal is transmitted with the data on the bus. The EUSART receiver thus has to determine where to sample the data on the bus from the actual data. To make this possible, additional synchronization bits are added to the data when operating in asynchronous mode, resulting in a slight overhead.

Asynchronous or synchronous mode can be selected by configuring SYNC in EUSARTn_CFG0. The options are listed with supported protocols in Table 20.1 EUSART Asynchronous vs. Synchronous Mode on page 609. Full duplex and half duplex communication is supported in both asynchronous and synchronous mode.

Table 20.1. EUSART Asynchronous vs. Synchronous Mode

SYNC	Communication Mode	Supported Protocols
0	Asynchronous	RS-232, IrDA
1	Synchronous	SPI, MicroWire

Table 20.2 EUSART Pin Usage on page 609 explains the functionality of the different EUSART pins when the EUSART operates in different modes. Pin functionality enclosed in square brackets is optional, and depends on additional configuration parameters. LOOPBK and MASTER are discussed in 20.3.2.7 Local Loopback and 20.3.3.3 Main SPI Interface Mode (Clock Driver) respectively.

Table 20.2. EUSART Pin Usage

SYNC	LOOPBK	MASTER	Pin Functionality			
			TX (MOSI)	RX (MISO)	CLK	CS
0	0	x	Data out	Data in	-	-
0	1	x	Data out/in	-	-	-
1	0	0	Data in	Data out	Clock in	Secondary select
1	0	1	Data out	Data in	Clock out	[Auto secondary select]
1	1	0	Data out/in	-	Clock in	Secondary select
1	1	1	Data out/in	-	Clock out	[Auto secondary select]

20.3.2 Asynchronous Operation

EUSART can operate in asynchronous mode when EUSARTn_CFG0.SYNC is set to 0.

EUSART0 may operate as either a high-speed peripheral running from a high-frequency clock source (HF mode, available in EM0 an EM1), or as a low-energy peripheral operating from a low-frequency clock source (LF mode, available in EM0, EM1, or EM2). EUSART0 operates in HF mode when the EUSART0CLK clock selected in CMU_EUSART0CLKCTRL_CLKSEL is EM01GRPACLK or HFRCOEM23. EUSART0 operates in LF mode when the selected clock is LFXO or LFRCO. Baud rate generation differs between these two modes, and there are certain operational restrictions in LF mode discussed in this chapter. It is not generally useful to switch between modes on-the-fly in a single application.

Other EUSART instances operate only as a high-speed peripheral running from the EM01GRPCCLK selected using CMU_EM01GRPCCLKCTRL_CLKSEL.

20.3.2.1 Frame Format

The frame format used in asynchronous mode consists of a set of data bits in addition to bits for synchronization and optionally a parity bit for error checking. A frame starts with one start-bit (S), where the line is driven low for one bit-period. This signals the start of a frame, and is used for synchronization. Following the start bit are 7 to 9 data bits and an optional parity bit. Finally, a number of stop-bits, where the line is driven high that indicates the end of the frame. An example frame is shown in Figure 20.2 Frame Format on page 610.

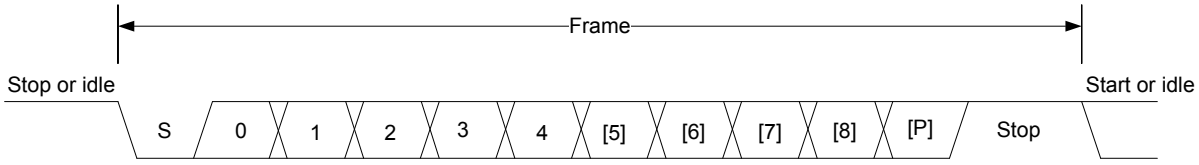


Figure 20.2. Frame Format

The number of data bits in a frame is set by DATABITS in EUSARTn_FRAMECFG, see Table 20.3 Data Bits on page 610, and the number of stop-bits is set by STOPBITS in EUSARTn_FRAMECFG, see Table 20.4 Stop Bits on page 610. Whether or not a parity bit should be included, and whether it should be even or odd is defined by PARITY, also in EUSARTn_FRAMECFG. For proper and reliable communication, all parties of a transfer must agree on the frame format prior to the start of the transfer.

Table 20.3. Data Bits

EUSARTn_FRAMECFG_DATABITS[3:0]	Number of Data bits
0001	7
0010	8 (Default)
0011	9

Table 20.4. Stop Bits

EUSARTn_FRAMECFG_STOPBITS[1:0]	Number of Stop bits
00	0.5 (HF clock operation only)
01	1 (Default)
10	1.5 (HF clock operation only)
11	2

The order in which the data bits are transmitted and received is defined by MSBF in EUSARTn_CFG0. When MSBF is cleared, data in a frame is sent and received with the least significant bit first. When it is set, the most significant bit comes first.

The frame format used by the transmitter can be inverted by setting TXINV in EUSARTn_CFG0, and the format expected by the receiver can be inverted by setting RXINV in EUSARTn_CFG0. These bits affect the entire frame, not only the data bits. An inverted frame has a low idle state, a high start-bit, inverted data and parity bits, and low stop-bits.

20.3.2.2 Parity Bit Calculation and Handling

When parity bit is enabled, hardware automatically calculates and inserts a parity bit into outgoing frames, and verifies the received parity bit in incoming frames. The possible parity modes are defined in Table 20.5 Parity Bits on page 611. When even parity is chosen, a parity bit is inserted to make the number of high bits (data + parity) even. If odd parity is chosen, the parity bit makes the total number of high bits odd. When parity bit is disabled, which is the default configuration, the parity bit is omitted.

Table 20.5. Parity Bits

EUSARTn_FRAMECFG_PARITY[1:0]	Description
00	No parity bit (Default)
01	Reserved
10	Even parity
11	Odd parity

20.3.2.3 Clock Generation

The EUSART clock defines the transmission and reception data rate. The baud rate is given by Figure 20.3 EUSART Baud Rate on page 612.

$$br = f_{EUSARTn}/(oversample \times (1 + EUSARTn\_CLKDIV/256))$$

Figure 20.3. EUSART Baud Rate

where $f_{EUSARTn}$ is the peripheral clock frequency and oversample is the oversampling rate as defined by OVS in EUSARTn_CFG0, see Table 20.6 Oversampling (EUSARTn_CFG0_OVS) on page 612. Note that different instances of the EUSART inside a device may use different peripheral clocks. The peripheral clock may be generically referred to in this chapter as clk_per.

Note: Please note that high frequency clocks should not be selected as UART clock source when nominal voltage is 0.9V.

Note:

Please note that when EUSARTn_CFG0_OVS is set to OVS_DISABLE (0x4), the peripheral clock frequency must be at least three times higher than the chosen baud rate. This condition is given in Figure 20.4 Requirement for EUSARTn_CFG0_OVS = OVS_DISABLE on page 612.

$$f_{EUSARTn} / br >= 3.0$$

Figure 20.4. Requirement for EUSARTn_CFG0_OVS = OVS_DISABLE

Table 20.6. Oversampling (EUSARTn_CFG0_OVS)

OVS[2:0]	Oversample
000	16 (HF clock operation only)
001	8 (HF clock operation only)
010	6 (HF clock operation only)
011	4 (HF clock operation only)
100	1 (OVS disabled - LF clock operation only)

Note: Please note that EUSARTn_CFG0_OVS must not be set to OVS_DISABLE (0x4) when one of the high frequency clocks is selected as EUSARTn peripheral clock source. When one of the low frequency clocks (LFXO/LFRCO) is selected as EUSARTn peripheral clock source, EUSARTn_CFG0_OVS must be set to OVS_DISABLE (0x4).

The EUSART has a fractional clock divider to allow the EUSART clock to be controlled more accurately than what is possible with a standard integral divider. The clock divider used in the EUSART is a 20-bit value, with a 15-bit integral part and an 5-bit fractional part. The fractional part is configured in the lower 5 bits of DIV in EUSARTn_CLKDIV. Fractional clock division is implemented by distributing the selected fraction over thirty two baud periods. The fractional part of the divider tells how many of these periods should be extended by one peripheral clock cycle.

Given a desired baud rate *brdesired*, the clock divider EUSARTn_CLKDIV can be calculated by using Figure 20.5 EUSART Desired Baud Rate on page 612:

$$EUSARTn\_CLKDIV = 256 \times (f_{EUSARTn}/(oversample \times brdesired) - 1)$$

Figure 20.5. EUSART Desired Baud Rate

Table 20.7 EUSART Baud Rates @ 4 MHz Peripheral Clock with 20 Bit CLKDIV on page 613 shows a set of desired baud rates and how accurately the EUSART is able to generate these baud rates when running at a 4 MHz peripheral clock, using 16x or 8x oversampling.

Table 20.7. EUSART Baud Rates @ 4 MHz Peripheral Clock with 20 Bit CLKDIV

Desired baud rate [baud/s]	EUSARTn_CFG0_OVS =00			EUSARTn_CFG0_OVS =01		
	EUSARTn_CLKDIV/256 (to 32nd position)	Actual baud rate [baud/s]	Error %	EUSARTn_CLKDIV/256 (to 32nd position)	Actual baud rate [baud/s]	Error[1] [%]
600	415,6563	600,015	0,003	832,3438	599,9925	-0,001
1200	207,3438	1199,94	-0,005	415,6563	1200,03	0,003
2400	103,1563	2400,24	0,010	207,3438	2399,88	-0,005
4800	51,09375	4799,04	-0,020	103,1563	4800,48	0,010
9600	25,03125	9603,842	0,040	51,09375	9598,08	-0,020
14400	16,375	14388,49	-0,080	33,71875	14401,44	0,010
19200	12,03125	19184,65	-0,080	25,03125	19207,68	0,040
28800	7,6875	28776,98	-0,080	16,375	28776,98	-0,080
38400	5,5	38461,54	0,160	12,03125	38369,3	-0,080
57600	3,34375	57553,96	-0,080	7,6875	57553,96	-0,080
76800	2,25	76923,08	0,160	5,5	76923,08	0,160
115200	1,15625	115942	0,644	3,34375	115107,9	-0,080
230400	0,09375	228571,4	-0,794	1,15625	231884,1	0,644

1 In addition to error introduced by oscillator frequency variation

Table 20.8 EUSART0 Baud Rates in LF mode @ 32.768 kHz Peripheral Clock with 20 bit CLKDIV on page 613 shows a set of desired baud rates and how accurately the EUSART0 is able to generate these baud rates when running from a 32.768 kHz peripheral clock in LF mode.

Table 20.8. EUSART0 Baud Rates in LF mode @ 32.768 kHz Peripheral Clock with 20 bit CLKDIV

Desired baud rate [baud/s]	EUSART0_CLKDIV/256	Actual baud rate [baud/s]	Error[1] [%]
300	108,21875	300,0217	0.01
600	53,625	599,8719	-0.02
1200	26,3125	1199,744	-0.02
2400	12,65625	2399,487	-0.02
4800	5,8125	4809,982	0.21
9600	2,40625	9619,963	0.21

1 In addition to error introduced by oscillator frequency variation

20.3.2.4 Auto Baud Detection (HF clock only)

Setting AUTOBAUDEN in EUSARTn_CFG0 uses the first frame received to automatically set the baud rate provided that it contains 0x55 (IrDA uses 0x00) and is sent out as LSB first and there is no break in the frame. The receiver will measure the number of local clock cycles between the beginning of the START bit and the beginning of the 8th data bit. The DIV field in EUSARTn_CLKDIV will be overwritten with the new value. The OVS in EUSARTn_CFG0 and the +1 count of the Baud Rate equation are already factored into the result that gets written into the DIV field. To restart autobaud detection, clear AUTOBAUDEN and set it high again. Since the auto baud rate detection is done over 8 baud times, only the upper 3 bits of the fractional part of the clock divider are populated.

Auto baud detection has associated status bit EUSARTn_STATUS_AUTOBAUDDONE and interrupt flag EUSARTn_IF_AUTOBAUDDONE. Both the status and the interrupt flag get set after auto baud detection is complete and DIV field in EUSARTn_CLKDIV is overwritten with the new value.

Note:

- If autobaud detection is enabled, software must wait for autobaud detection to complete before transmitting any data.
- Autobaud should be used only during times when it is known that the transmitter will be sending the required data word.
- Autobaud detection is not available when operating with LF clock source.
- For autobaud to work with IrDA, there should be odd parity or no parity in the received data frame.

20.3.2.5 Data Transmission

Asynchronous data transmission is initiated by writing data to the transmit FIFO using one of the methods described in 20.3.2.5.1 Transmit FIFO Operation. When the transmission shift register is empty and if the transmitter is enabled, a frame from the transmit FIFO is loaded into the shift register and transmission begins. Note that the frame loading in to the shift register can also be blocked if CTSEN is set in EUSARTn_CFG0 but the CTS input is inactive. When the frame has been transmitted, a new frame is loaded into the shift register if available, and transmission continues. If the transmit FIFO is empty, the transmitter goes to an idle state, waiting for a new frame to become available.

Transmission is enabled through the command register EUSARTn_CMD by setting TXEN, and disabled by setting TXDIS in the same command register. When the transmitter is disabled using TXDIS, any ongoing transmission is aborted, and any frame currently being transmitted is discarded. When disabled, the TX output goes to an idle state, which by default is a high value. Whether or not the transmitter is enabled at a given time can be read from TXENS in EUSARTn_STATUS.

When the EUSART transmitter is enabled and there is no data in the transmit shift register or transmit FIFO, the TXC status flag in EUSARTn_STATUS and the TXC interrupt flag in EUSARTn_IF are set, signaling that the transmission is complete. The TXC status flag is cleared when a new frame becomes available for transmission, but the TXC interrupt flag must be cleared by software.

Note: (1) Condition for TX to send data out: TX is enabled and there is data available in the TX FIFO and CTS is either not enabled, or if it is enabled, then it is active. (2) If TX output is tri-stated using TXDIS command or TXDISAT setting, then the TX module will still send out data (emptying the FIFO) if condition in point 1 is satisfied, even though the pad is tristated. Restriction: User should not set TXEN when the output is tristated.

20.3.2.5.1 Transmit FIFO Operation

The transmit FIFO is a 16 deep FIFO. A frame can be loaded into the FIFO by writing to EUSARTn_TXDATA. Using EUSARTn_TXDATA allows 9 bits to be written to the FIFO, as well as a set of control bits regarding the transmission of the written frame. Every frame in the FIFO is stored with 9 data bits and additional transmission control bits. A frame is loaded from the FIFO in to the shift register if the transmitter is enabled.

Figure 20.6 Transmit FIFO Operation on page 615 shows the basics of the transmit FIFO.

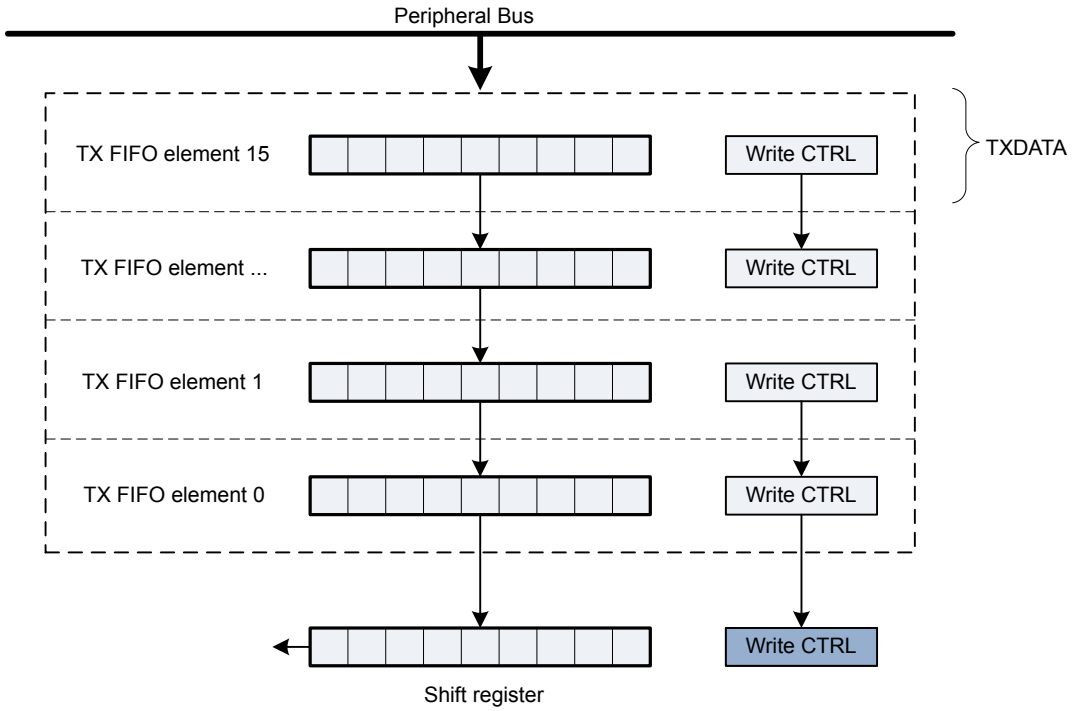


Figure 20.6. Transmit FIFO Operation

In addition to the interrupt flag TXC in EUSARTn_IF and status flag TXC in EUSARTn_STATUS which are set when the transmission is complete, TXFL in EUSARTn_STATUS and the TXFL interrupt flag in EUSARTn_IF are used to indicate the level of the transmit FIFO. TXFIW in EUSARTn_CFG1 controls the level at which these bits are set. The user can choose a transmit FIFO watermark level in TXFIW field of EUSARTn_CFG1 register in order to program when the TXFL interrupt should be triggered.

There is a TXIDLE status bit in EUSARTn_STATUS to provide an indication of when the transmitter is idle. The count of TX valid FIFO entries is called TXFCNT in EUSARTn_STATUS.

The transmit FIFO can be cleared by setting CLEARTX in EUSARTn_CMD. Since this is an Async FIFO, the user needs to first issue the CLEARTX command and then wait on the CLEARTXBUSY status flag until it goes low. EUSART must be enabled (EUSARTn_EN should be set) for the flush to work. EUSART should not be transmitting when CLEARTX command gets issued (can be achieved by disabling the trasmitter). Any frame present in the transmit shift register currently being transmitted will not be aborted due to the flush. Transmission of this frame will be completed. Note that the transmit shift register is never used to store a transmit frame, i.e., if the transmitter is not enabled then the data stays in the transmit FIFO until the transmitter gets enabled. Whenever a frame is loaded in to the transmit shift register, the transmission starts immediately.

Transmit FIFO Status Flags

TX FIFO has two associated status flags: TXFL (set when there is space for data in the TX FIFO, depends on the setting in the TXFIW in CFG1 register) and TXFCNT (count of number of TX frames in the FIFO). These status flags remain set as long as the underlying condition is true, even if the EUSART is disabled (i.e., EUSARTn_EN is 0). It is possible to write to the TX FIFO while EUSART is disabled, this will impact the two flags mentioned above.

Transmit FIFO Interrupt Flags

TX FIFO has two associated interrupt flags: TXFL and TXOF. TXFL is set when space becomes available in the FIFO, depends on TXFIW in CFG1 register. Reset value for TXFIW is such that TXFL gets set as soon the chip comes out of reset (so the interrupt should

be cleared once TXFIW is updated). TXFL remains set as long as the underlying condition is true even if the EUSART is disabled. This means that if a software clear is done for TXFL, then the interrupt will get set again after the clear if the underlying condition is still true (this will happen even if the EUSART is disabled). Writing more data to the FIFO or disabling the TXFL will make the TXFL go away (even if the EUSART is disabled). When writing more frames to the transmit FIFO than there is free space for, the TXOF interrupt flag in EUSARTn_IF will be set, indicating the overflow. The data already in the transmit FIFO is preserved in this case, and no data is written. Note that TXOF can also trigger if DMA tries to write to the FIFO while a TX Flush operation is going on (Flush is activated by setting CMD.CLEARTX). No data will be written to the FIFO in this case. TXOF triggers only once every time an overflow occurs.

Note: (EUSART0 only) In LF clock mode, the TXFL interrupt flag and the TXFL wakeup flag differ slightly in their behavior. IF.TXFL is always set out of reset since there is space available in the FIFO. However, the TXFL wakeup flag is only set after a FIFO read happens and the space that becomes available in the FIFO is the same as programmed in TXFIW in EUSART0_CFG1.

20.3.2.5.2 Frame Transmission Control

The transmission control bits, which can be written using EUSARTn_TXDATA, affect the transmission of the written frame. The following options are available:

- Generate break: By setting TXBREAK, the output will be held low during the stop-bit period to generate a framing error. A receiver that supports break detection detects this state, allowing it to be used e.g. for framing of larger data packets. The line is driven high before the next frame is transmitted so the next start condition can be identified correctly by the recipient. Continuous breaks lasting longer than a EUSART frame are thus not supported. GPIO can be used for this.
- Disable transmitter after transmission: If TXDISAT is set, the transmitter is disabled after the frame has been fully transmitted.
- Enable receiver after transmission: If RXENAT is set, the receiver is enabled after the frame has been fully transmitted. It is enabled in time to detect a start-bit directly after the last stop-bit has been transmitted.
- Unblock receiver after transmission: If UBRXAT is set, the receiver is unblocked and RXBLOCK is cleared after the frame has been fully transmitted.
- Tristate transmitter after transmission: If TXTRIAT is set, TXTRI is set after the frame has been fully transmitted, tristating the transmitter output. Note that if there are more frames in the TX FIFO after the tristating has happened and the transmitter is enabled, then the transmitter will try to send them out but since the output is tristated, nothing will show up at the transmitter output. The FIFO however will get emptied because of the transmitter attempting to send these frames out. If the target is to automatically tristate the TX line whenever the transmitter is idle, then that can be done by setting AUTOTRI in EUSARTn_CFG0. If AUTOTRI is set TXTRI status flag is always read as 0.

20.3.2.5.3 TX Status Flags

The following status flags should be used with care keeping in mind the external sources that can impact these flags as well as the synchronization delay when the status signal crosses over from the EUSART Core clock domain to the APB clock domain. FIFO related status flags were already discussed in 20.3.2.5.1 Transmit FIFO Operation, the remaining flags are mentioned below:

- TXENS: Enable sources: (1) TXEN command from software, (2) PRS trigger (when TXTEN=1). Disable Sources: (1) TXDIS command from software, (2) PERR/FERR (when ERRSTX=1), (3) Software when TXDISAT is set to 1 for a frame.
- TXTRI: Enable sources: (1) TXTRIEN command from software, (2) Software when TXTRIAT is set to 1 for a frame. Disable sources: (1) TXTRIDIS command from software.
- TXIDLE: Set whenever the TX module is idle.
- TXC: Set whenever the TX module goes to idle and both the TX FIFO and the TX shift register are empty. Cleared when either TX FIFO or the TX shift register has data.

20.3.2.5.4 Transmission Delay

By configuring TXDELAY in EUSARTn_CFG1, the transmitter can be forced to wait a number of bit-periods from when it is ready to transmit data, to when it actually transmits the data. This delay is only applied to the first frame transmitted after the transmitter has been idle. When transmitting frames back-to-back the delay is not introduced between the transmitted frames.

This is useful on half duplex buses, because the receiver always returns received frames to software during the first stop-bit. The bus may still be driven for up to 3 bit periods, depending on the current frame format. Using the transmission delay, a transmission can be started when a frame is received, and it is possible to make sure that the transmitter does not begin driving the output before the frame on the bus is completely transmitted.

20.3.2.6 Data Reception

Data reception is enabled by setting RXEN in EUSARTn_CMD. When the receiver is enabled, it actively samples the input looking for a transition from high to low indicating the start baud of a new frame. When a start baud is found, reception of the new frame begins. When the frame has been received, it is pushed into the receive FIFO, making the shift register ready for another frame of data, and the receiver starts looking for another start baud. If a frame is received while the receive FIFO is full, the received frame is discarded and the RXOF interrupt flag in EUSARTn_IF is set to indicate the FIFO overflow.

The receiver can be disabled by setting the command bit RXDIS in EUSARTn_CMD. Any frame currently being received, when the receiver is disabled, is discarded. Whether or not the receiver is enabled at a given time can be read out from RXENS in EUSARTn_STATUS.

20.3.2.6.1 Receive FIFO Operation

The receive-FIFO is a 16 deep FIFO. Data can be read from the receive FIFO via EUSARTn_RXDATA. EUSARTn_RXDATA gives access to the received frame. This register also contains parity error and framing error information of the received frame. When a frame is read from the receive FIFO using EUSARTn_RXDATA, the frame is pulled out of the FIFO, making room for a new frame. If an attempt is done to read more frames from the FIFO than what is available, the RXUF interrupt flag in EUSARTn_IF is set to signal the underflow, and the data read from the FIFO is undefined.

Frames can be read from the receive FIFO without removing the data by using EUSARTn_RXDATAP. EUSARTn_RXDATAP gives access to the first frame in the FIFO with status bits. The data read from this register when the receive FIFO is empty is undefined. No underflow interrupt is generated by a read using this register, i.e. RXUF in EUSARTn_IF is never set as a result of reading from EUSARTn_RXDATAP.

The basic operation of the receive FIFO is shown in Figure 20.7 EUSART Receive FIFO Operation on page 618.

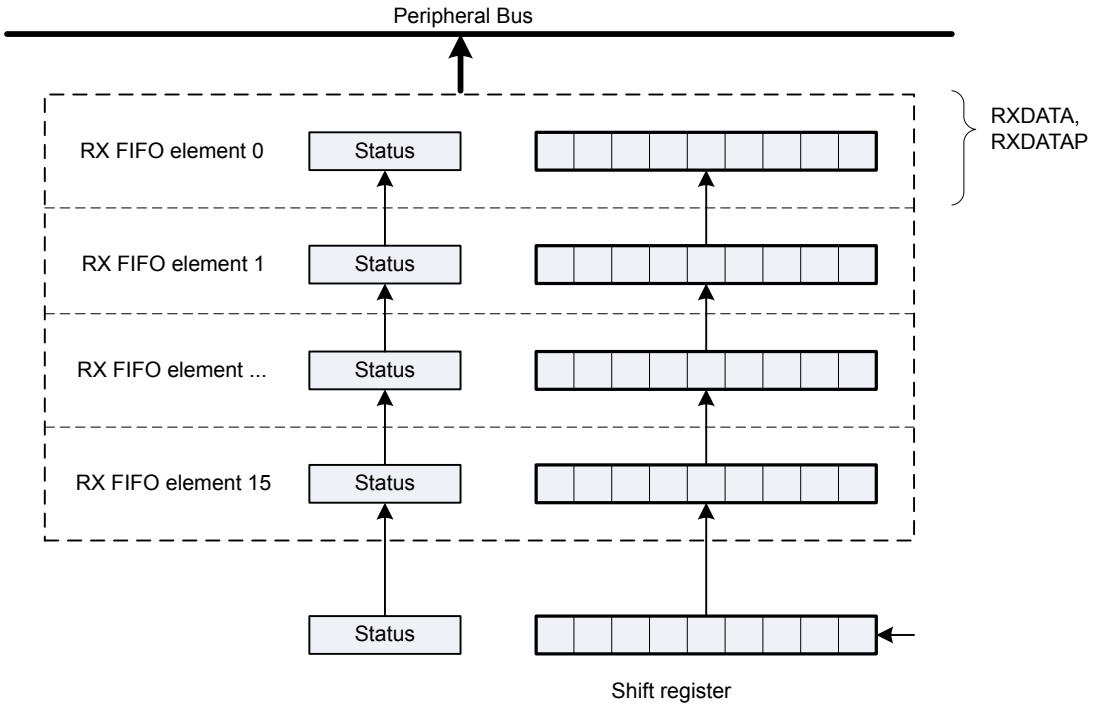


Figure 20.7. EUSART Receive FIFO Operation

Receive FIFO Status Flags

Receive FIFO has two associated status flags: RXFL (set when number of available frames in the receive FIFO is at least number of frames set by RXFIW in CFG1 register) and RXFULL (set when receive FIFO is full). These status flags remain set as long as the underlying condition is true, even if the EUSART is disabled (i.e., EUSARTn_EN is 0). It is possible to read from the receive FIFO while EUSART is disabled, this will impact the two status flags mentioned above. The status flags RXFL and RXFULL are automatically cleared by hardware when their condition is no longer true.

Receive FIFO Interrupt Flags

Receive FIFO has four associated interrupt flags: RXFL, RXFULL, RXOF and RXUF. RXFL is set when number of available frames in the receive FIFO is at least number of frames set by RXFIW in CFG1 register. RXFULL is set when receive FIFO is full. Both RXFL and RXFULL remains set as long as the underlying condition is true even if EUSART is disabled. This means that if a software clear is done for RXFL / RXFULL while the undelying condition of respective interrupt is still true, the corresponding interrupt will get set again after the clear (this will happen even if EUSART is disabled). Reading data from the FIFO or disabling the RXFL / RXFULL will block the respective interrupt to get set after a software clear (even if EUSART is disabled). RXOF is set when a new frame is received while the receive FIFO is full, indicating overflow. The new frame is discarded. RXOF triggers only once every time an overflow occurs. RXUF is set when an attempt (via RXDATA or DMA) is done to read more frames from the receive FIFO than what is available, indicating underflow. The data read from the FIFO is undefined. RXUF triggers every time an underflow occurs even if EUSART is disabled.

20.3.2.6.2 Blocking Incoming Data

When using hardware frame recognition, as detailed in 20.3.2.6.9 Multi-Processor Mode and 20.3.2.8 Collision Detection, it is necessary to be able to let the receiver sample incoming frames without loading them into the receive FIFO. This is accomplished by blocking incoming data.

Incoming data is blocked as long as RXBLOCK in EUSARTn_STATUS is set. When blocked, frames received by the receiver will not be loaded into the receive FIFO, and software is not notified by the RXFL flag in EUSARTn_STATUS or the RXFL interrupt flag in EUSARTn_IF at their arrival. For data to be loaded into the receive FIFO, RXBLOCK must be cleared in the instant a frame is fully received by the receiver. RXBLOCK is set by setting RXBLOCKEN in EUSARTn_CMD and disabled by setting RXBLOCKDIS also in EUSARTn_CMD. There is one exception where data is loaded into the receive FIFO even when RXBLOCK is set. This is when an address frame is received while operating in multi-processor mode. See 20.3.2.6.9 Multi-Processor Mode for more information.

Frames received containing framing or parity errors will not result in the FERR and PERR interrupt flags in EUSARTn_IF being set while RXBLOCK in EUSARTn_STATUS is set. Hardware recognition is not applied to these erroneous frames, and they are silently discarded.

The overflow interrupt flag RXOF in EUSARTn_IF will also not be set while RXBLOCK in EUSARTn_STATUS is set.

20.3.2.6.3 Data Sampling and Filtering

The receiver can sample incoming signal at a rate 16, 8, 6 or 4 times higher than the given baud rate, depending on the oversampling mode given by OVS in EUSARTn_CFG0. Lower oversampling rates make higher baud rates possible, but give less room for errors.

When a high-to-low transition is registered on the input while the receiver is idle, this is recognized as a start-bit, and the baud rate generator is synchronized with the incoming frame.

For oversampling modes 16, 8 and 6, every bit in the incoming frame is sampled three times to gain a level of noise immunity. These samples are aimed at the middle of the bit-periods, as visualized in Figure 20.8 EUSART Sampling of Start and Data Bits on page 620. With OVS=0 in EUSARTn_CFG0, the start and data bits are thus sampled at locations 8, 9 and 10 in the figure, locations 4, 5 and 6 for OVS=1 and locations 3, 4, and 5 for OVS=2. The value of a sampled bit is determined by majority vote. If two or more of the three bit-samples are high, the resulting bit value is high. If the majority is low, the resulting bit value is low.

Majority vote is used for all oversampling modes except 4X oversampling and when oversampling is disabled. In 4X oversampling mode, a single sample is taken at position 3 as shown in Figure 20.8 EUSART Sampling of Start and Data Bits on page 620.

When oversampling is disabled i.e. OVS = DISABLE, there is only one available location for sampling the start and data bits and so majority vote is not used.

Note: When operating in HF clock mode, oversampling must be set to 4, 6, 8, or 16x. When operating in LF clock mode, oversampling must be disabled.

Software can disable majority vote by setting MVDIS in EUSARTn_CFG0. When majority vote is disabled by software, a single sample is taken at location 9 in the figure for OVS=0, location 5 for OVS=1 and location 4 for OVS=2.

If the value of the start bit is found to be high, the reception of the frame is aborted, filtering out false start bits possibly generated by noise on the input.

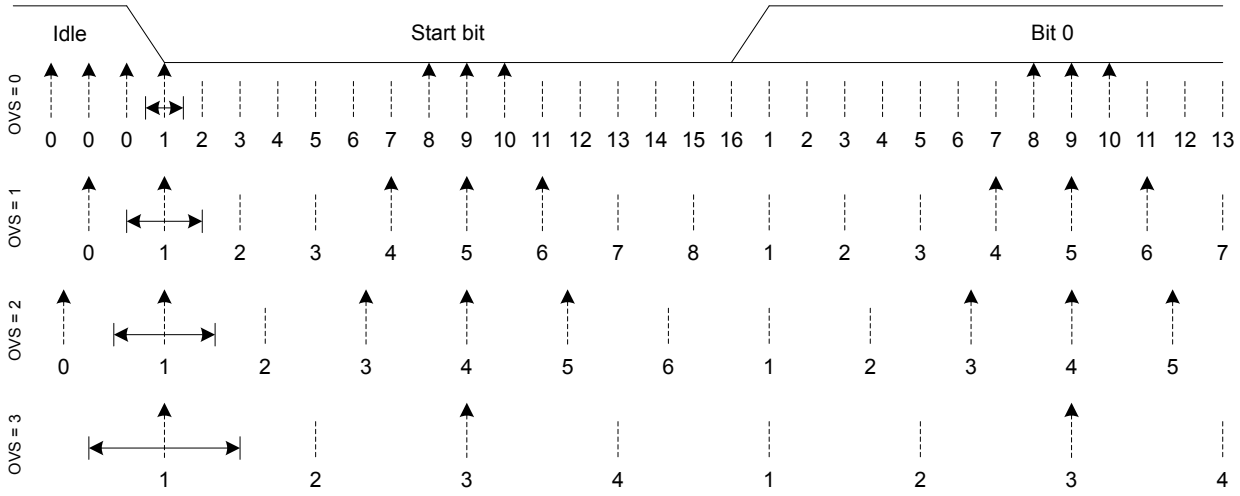


Figure 20.8. EUSART Sampling of Start and Data Bits

If the baud rate of the transmitter and receiver differ, the location each bit is sampled will be shifted towards the previous or next bit in the frame. This is acceptable for small errors in the baud rate, but for larger errors, it will result in transmission errors.

When the number of stop bits is 1 or more, stop bits are sampled like the start and data bits as seen in Figure 20.9 EUSART Sampling of Stop Bits when Number of Stop Bits are 1 or More on page 621. When a stop bit has been detected, EUSARTn is ready for a new start bit.

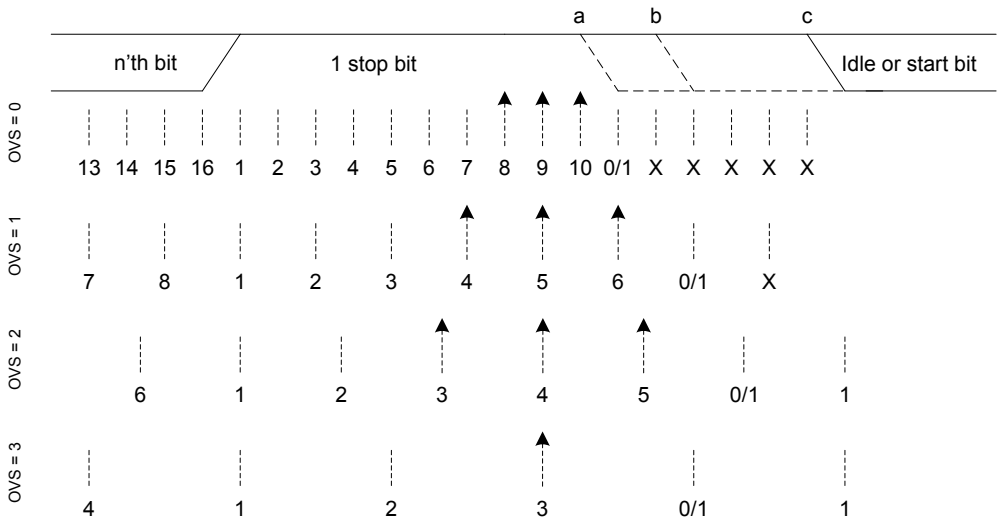


Figure 20.9. EUSART Sampling of Stop Bits when Number of Stop Bits are 1 or More

When working with stop bit lengths of half a baud period, the above sampling scheme no longer suffices. In this case, the stop-bit is not sampled, and no framing error is generated in the receiver if the stop-bit is not generated. However, the line must still be driven high before the next start bit for the EUSART to successfully identify the start bit.

20.3.2.6.4 RX Status Flags

The following status flags should be used with care keeping in mind the external sources that can impact these flags as well as the synchronization delay when the status signal crosses over from the EUSART Core clock domain to the APB clock domain. FIFO related status flags were already discussed in 20.3.2.6.1 Receive FIFO Operation, the remaining flags are mentioned below:

- RXENS: set when RX is enabled. Enable sources: (1) RXEN command from software, (2) PRS trigger (when RXTEN=1), (3) TX (when RXENAT=1). Disable Sources: (1) RXDIS command from software, (2) Parity / framing error (when ERRSRX=1).
- RXBLOCK: set when RX is blocked which means RX keeps receiving frames but does not push received frame to receive FIFO. Enable sources: (1) RXBLOCKEN command from software. Disable Sources: (1) RXBLOCKDIS command from software, (2) STARTFRAME match (when SFUBRX=1), (3) TX (when UBRXAT=1).
- RXIDLE: Set when RX module completes pushing the last received frame to receive FIFO (if receive FIFO has space) and is not receiving any new frame or when RX gets disabled.
- AUTOBAUDDONE: Set when auto baud detection is complete and DIV field in EUSARTn_CLKDIV is overwritten with the detected value.

20.3.2.6.5 Parity Error

When parity bits are enabled, a parity check is automatically performed on incoming frames. When a parity error is detected in an incoming frame, the data parity error bit PERR in the frame is set, as well as the interrupt flag PERR in EUSARTn_IF. Frames with parity errors are loaded into the receive FIFO like regular frames.

PERR can be accessed by reading the frame from the receive FIFO using the EUSARTn_RXDATA, or EUSARTn_RXDATAP registers.

If ERRSTX in EUSARTn_CFG0 is set, the transmitter is disabled on received parity errors. If ERRSRX in EUSARTn_CFG0 is set, the receiver is disabled on parity errors.

20.3.2.6.6 Framing Error and Break Detection

A framing error is the result of an asynchronous frame where the stop bit was sampled to a value of 0. This can be the result of noise and baud rate errors, but can also be the result of a break generated by the transmitter on purpose.

When a framing error is detected in an incoming frame, the framing error bit FERR in the frame is set. The interrupt flag FERR in EUSARTn_IF is also set. Frames with framing errors are loaded into the receive FIFO like regular frames.

If ERRSTX in EUSARTn_CFG0 is set, the transmitter is disabled on received framing errors. If ERRSRX in EUSARTn_CFG0 is set, the receiver is disabled on framing errors.

20.3.2.6.7 Programmable Start Frame

The EUSART can be configured to start receiving data when a special start frame is detected on the input. This can be useful when operating in low energy modes, allowing other devices to gain the attention of the EUSART by transmitting a given frame.

When SFUBRX in EUSARTn_CFG1 is set, an incoming frame matching the frame defined in EUSARTn_STARTFRAME will result in RXBLOCK in EUSARTn_STATUS being cleared. This can be used to enable reception when a specified start frame is detected. If the receiver is enabled and blocked, i.e. RXENS and RXBLOCK in EUSARTn_STATUS are set, the receiver will receive all incoming frames, but unless an incoming frame is a start frame it will be discarded and not loaded into the receive FIFO. When a start frame is detected, the block is cleared, and frames received from that point, including the start frame, are loaded into the receive FIFO.

An incoming start frame results in the STARTFIF interrupt flag in EUSARTn_IF being set, regardless of the value of SFUBRX in EUSARTn_CFG1. This allows an interrupt to be made when the start frame is detected. The interrupt will be set even if the receiver is blocked i.e. EUSARTn_STATUS_RXBLOCK = 1.

Note: The receiver must be enabled for start frames to be detected. Please note that, if another UART device sends a start frame but a parity and/or framing error occurs during the receiption, the received frame is not detected as a start frame.

20.3.2.6.8 Programmable Signal Frame

As well as the configurable start frame, a special signal frame can be specified. When a frame matching the frame defined in EUSARTn_SIGFRAME is detected by the receiver, the SIGF interrupt flag in EUSARTn_IF is set. As like start frame detection, the interrupt will be set even if the receiver is blocked i.e. EUSARTn_STATUS_RXBLOCK = 1.

One use of the programmable signal frame is to signal the end of a multi-frame message transmitted to the EUSART. An interrupt will then be triggered when the packet has been completely received, allowing software to process it. Used in conjunction with the programmable start frame and DMA, this makes it possible for the EUSART to automatically begin the reception of a packet on a specified start frame, load the entire packet into memory, and give an interrupt when reception of a packet has completed. When one of the low frequency oscillators (LFXO,LFRCO) is used as EUSART0 peripheral clock source, the device can thus wait for data packets in EM2, and only be woken up when a packet has been completely received.

Note: The receiver must be enabled for a signal frame to be detected. If a parity and/or framing error occurs during the reception of a signal frame, the received frame is not detected as a signal frame.

20.3.2.6.9 Multi-Processor Mode

To simplify communication between multiple processors, the EUSART supports a special multi-processor mode. In this mode the 9th data bit in each frame is used to indicate whether the content of the remaining 8 bits is data or an address.

When multi-processor mode is enabled, an incoming 9-bit frame with the 9th bit equal to the value of MPAB in EUSARTn_CFG0 is identified as an address frame. When an address frame is detected, the MPAF interrupt flag in EUSARTn_IF is set, and the address frame is loaded into the receive register. This happens regardless of the value of RXBLOCK in EUSARTn_STATUS.

Multi-processor mode is enabled by setting MPM in EUSARTn_CFG0, and the value of the 9th bit in address frames can be configured in EUSARTn_CFG0_MPAB. Note that the receiver must be enabled for address frames to be detected. The receiver can be blocked however, preventing data from being loaded into the receive FIFO while looking for address frames.

The following section explains basic usage of the multi-processor mode:

EUSART Multi-processor Mode Example

1. All devices on the bus enable multi-processor mode and enable and block the receiver. They will now not receive data unless it is an address frame. MPAB in EUSARTn_CFG0 is set to desired value to identify frames with the 9th bit value equal to MPAB as address frames.
2. A transmitting device sends a frame containing the address of a different device and with the 9th bit set to value of MPAB.
3. All devices receive the address frame and get an interrupt. They can read the address from the receive FIFO. The selected device unblocks the receiver to start receiving data from the transmitter.
4. The transmitter sends data with the 9th bit set to opposite value of MPAB.
5. Only the addressed device with RX enabled and unblocked receives the data. When transmission is complete, the device blocks the receiver and waits for a new address frame.

When a device has received an address frame and wants to receive the following data, it must make sure the receiver is unblocked before the next frame has been completely received in order to prevent data loss. One option is to set SFUBRX in EUSARTn_CFG1 and set the address frame as start frame in EUSARTn_STARTFRAME. This will handle automatic unblocking of RX when address frame is received.

20.3.2.6.10 RX Timeout

A RX timeout function can be enabled by setting EUSART_CFG1_RXTIMEOUT to desired value. When enabled, a timer gets started after every successful frame reception. If timeout occurs before the next RX start bit is received, EUSART_IF_RXTO gets set which can be used to wake-up the system to handle received data. This is shown in Figure 20.10 RX Timeout on page 623 If the next RX start bit is received before timeout occurs, no interrupt gets generated, the timer is reset and will only be started again after the on-going frame reception is complete.

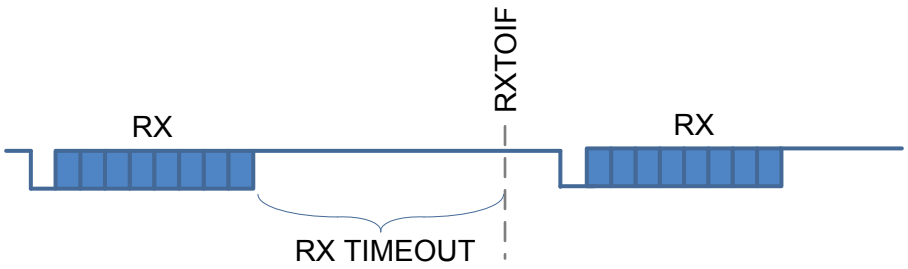


Figure 20.10. RX Timeout

Please note that the timer does not get started in following scnearios:

- If auto baud rate detection is enabled and auto baud rate has not been found, the timer does not get started after the first frame which is used to detect the baud rate automatically.
- If a frame is received while RX is blocked, the timer does not get started.
- If EUSART_CFG0_SKIPPERRF is set to '1' and a frame is received with a parity error, the timer does not get started.

Please also note that the UART RX input line must be in idle state after frame reception for the timer to start.

Note: When EUSARTn_FRAMECFG_STOPBITS is set to a fractional value i.e. 'HALF' or 'ONEANDHALF', the fractional value is rounded to nearest value and so RX timeout is extended by 0.5 baud width per frame.

20.3.2.7 Local Loopback

The EUSARTn receiver samples RX by default, and the transmitter drives TX by default. This is not the only option however. When LOOPBK in EUSARTn_CFG0 is set, the RX pin is connected to the TX pin as shown in Figure 20.11 EUSART Local Loopback on page 624. This is useful for debugging, as the EUSARTn can receive the data it transmits, but it is also used to allow the EUSARTn to read and write to the same pin, which is required for some half duplex communication modes. In this mode, the TX pin must be enabled as an output in the GPIO.

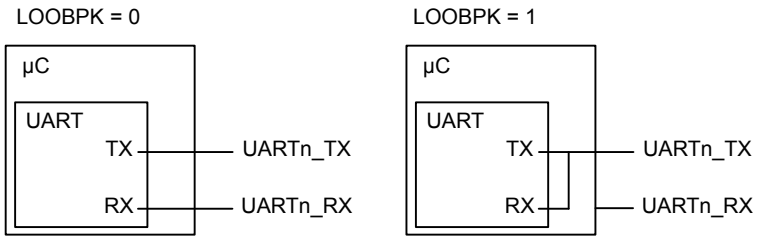


Figure 20.11. EUSART Local Loopback

20.3.2.8 Collision Detection

EUSARTn supports a basic form of collision detection. When the receiver is connected to the output of the transmitter, either by using the LOOPBK bit in EUSARTn_CFG0 or through an external connection, this feature can be used to detect whether data transmitted on the bus by the EUSARTn did get corrupted by a simultaneous transmission by another device on the bus.

For collision detection to be enabled, CCEN in EUSARTn_CFG0 must be set, and the receiver enabled. The data sampled by the receiver is then continuously compared with the data output by the transmitter. If they differ, the CCF interrupt flag in EUSARTn_IF is set. The collision check includes all bits of the transmitted frames. The CCF interrupt flag is set once for each bit sampled by the receiver that differs from the bit output by the transmitter. When the transmitter output is disabled, i.e. the transmitter is tristated, collisions are not registered.

Note: Please note that collision detection supports only baudrates up to 1mbps.

20.3.2.9 Half Duplex Communication

When doing full duplex communication, two data links are provided, making it possible for data to be sent and received at the same time. In half duplex mode, data is only sent in one direction at a time. There are several possible half duplex setups, as described in the following sections.

20.3.2.9.1 Single Data-link

In this setup, the EUSART both receives and transmits data on the same pin. This is enabled by setting LOOPBK in EUSARTn_CFG0, which connects the receiver to the transmitter output. Because they are both connected to the same line, it is important that the EUSART transmitter does not drive the line when receiving data, as this would corrupt the data on the line.

When communicating over a single data-link, the transmitter must thus be tristated whenever not transmitting data. This is done by setting the command bit TXTRIEN in EUSARTn_CMD, which tristates the transmitter. Before transmitting data, the command bit TXTRIDIS, also in EUSARTn_CMD, must be set to enable transmitter output again. Whether or not the output is tristated at a given time can be read from TXTRI in EUSARTn_STATUS. If TXTRI is set when transmitting data, the data is shifted out of the shift register, but is not put out on TX line.

When operating a half duplex data bus, it is common to have a main bus controller, which first transmits a request to one of the secondary devices on the bus, then receives a reply. In this case, the frame transmission control bits, which can be set by writing to EUSARTn_TXDATA, can be used to make the EUSART automatically disable transmission, tristate the transmitter and enable reception when the request has been transmitted, making it ready to receive a response from the secondary device.

Tristating the transmitter can also be performed automatically by the EUSART by using AUTOTRI in EUSARTn_CFG0. When AUTOTRI is set, the EUSART automatically tristates TX line whenever the transmitter is idle, and enables transmitter output when the transmitter goes active. If AUTOTRI is set TXTRI is always read as 0.

Note: Another way to tristate the transmitter is to enable wired-and or wired-or mode in GPIO. For wired-and mode, outputting a 1 will be the same as tristating the output, and for wired-or mode, outputting a 0 will be the same as tristating the output. This can only be done on buses with a pull-up or pull-down resistor respectively.

20.3.2.9.2 Single Data-link with External Driver

Some communication schemes, such as RS-485 rely on an external driver. Here, the driver has an extra input which enables it, and instead of tristating the transmitter when receiving data, the external driver must be disabled.

This can be done manually by assigning a GPIO to turn the driver on or off.

Figure 20.12 EUSART Half Duplex Communication with External Driver on page 625 shows an example configuration using an external driver.

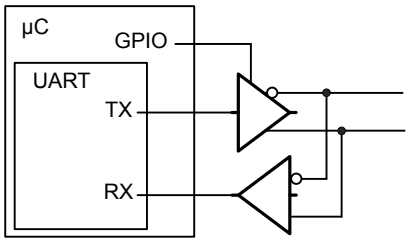


Figure 20.12. EUSART Half Duplex Communication with External Driver

20.3.2.9.3 Two Data-links

Some limited devices only support half duplex communication even though two data links are available. In this case software is responsible for making sure data is not transmitted when incoming data is expected.

20.3.2.10 Hardware Flow Control

Hardware flow control can be used to hold off the link partner's transmission until receive FIFO space is available. Use RTSPEN in GPIO_DBUSEUSARTn_ROUTEEN to enable RTS pin (CTS is an input so its pin is enabled by default). Port and Pin selection for RTS and CTS can be done in GPIO_DBUSEUSARTn_RTSROUTE/CTSROUTE. RTS is an out going signal which indicates that receive FIFO space is available to receive a frame. The link partner is being requested to send its data when RTS is active. RTS activation can be made dependent on how much space is available in the receive FIFO using RTSRXFW in EUSARTn_CFG1. For debug use set DBGHALT in EUSARTn_CFG1 which will force the RTS to request one frame from the link partner when the CPU core single steps. RTS is deactivated when RX is disabled.

CTS is an incoming signal to stop the next TX data from going out. CTS indicates that the link partner has receive FIFO space available, and the local transmitter is clear to send. When CTS deactivates in the middle of a frame, the frame currently being transmitted is completed before stopping. CTS operation needs to be enabled using CFG1.CTSEN.

The RTS and CTS are active low by default, but their polarity can be changed with RTSINV and CTSINV in EUSARTn_CFG1 respectively.

20.3.2.11 Debug Halt

During single stepping, debug halt feature allows halting EUSART frame reception by deactivating RTS when the core is halted and continuing frame reception by activating RTS when the core is unhalted. EUSART debug halt can be enabled by setting DBGHALT in EUSART_CFG1 to '1'. EUSART receiver must be enabled for debug halting.

When EUSART_CFG1_DBGHALT is not set or EUSART_CFG1_DBGHALT is set but chip halt is low, RTS is only dependent on the receive FIFO having space available to receive at least number of frames given by EUSART_CFG1_RTSRXFW settings.

When EUSART_CFG1_DBGHALT is set, RTS will remain deactivated as long as chip halt is high. When a low pulse is detected on chip halt while DBGHALT is set, RTS will be activated if receive FIFO has space available to receive at least number of frames given by EUSART_CFG1_RTSRXFW settings and no frame is being received. RTS will be deactivated again if chip halt goes back to high and receiver starts receiving a new frame or if receive FIFO does not have space available to receive at least number of frames given by EUSART_CFG1_RTSRXFW settings. This behavior allows single stepping to pulse the chip halt low for a cycle, and receive the next frame.

Please note that, if chip halt remains low for a short duration after RX is enabled while DBGHALT is set, initial low value of chip halt will be treated the same as a low pulse on chip halt.

As incoming frame is always received until receive FIFO is full regardless of the state of DBGHALT or chip halt, the link partner must stop transmitting when RTS is deactivated, or the receive FIFO could overflow.

All data in the transmit FIFO is sent out even when chip halt is asserted; therefore, DMA will need to be set to stop sending EUSART TX data during chip halt.

20.3.2.12 PRS-triggered Transmissions

If a transmission must be started on an event with very little delay, the PRS system can be used to trigger the transmission. The PRS channel to use as a trigger can be selected using EUSARTn_TRIGGER.PRSSEL in PRS. When a positive edge is detected on PRS signal, the receiver is enabled if RXTEN in EUSARTn_TRIGCTRL is set, and the transmitter is enabled if TXTEN in EUSARTn_TRIGCTRL is set. Only one signal input is supported by the EUSART.

20.3.2.13 PRS RX Input

The EUSART can be configured to receive data directly from a PRS channel by setting RXPRSEN in EUSARTn_CFG1. The PRS channel used is selected using EUSARTn_RX.PRSSEL in PRS.

20.3.2.14 DMA Support

The EUSART has full DMA support. The DMA controller can write to the transmit FIFO using the register EUSARTn_TXDATA, and it can read from the receive FIFO using the register EUSARTn_RXDATA. This enables 9 bit data + control/status bits transfers both to and from the EUSART.

A request for the DMA controller to read from the EUSART receive buffer can come from the following source:

- Receive FIFO is loaded with at least number of frames set by RXFIW.

A write request can come from the following source:

- Transmit FIFO has space for at least number of frames set by TXFIW

In some cases, it may be sensible to temporarily stop DMA read access to the EUSART when an error such as parity or framing error has occurred. This is enabled by setting ERRSDMA in EUSARTn_CFG0.

EUSART0 (EM2 Capable instance only) can also work with the DMA in low power mode so that the system does not have to wake up to EM0 to consume data. This can happen if TXDMAWU or RXDMAWU in the EUSARTn_CFG1 is set. The DMA will be triggered when TXFIW/RXFIW samples are in the corresponding FIFO. The chip will enter EM1, DMA will then pop/ push all the elements of the corresponding FIFO and then the system will be put back to EM2.

20.3.2.15 Interrupts

The interrupts generated by the EUSART are combined into two interrupt vectors. Interrupts related to reception are assigned to one interrupt vector, and interrupts related to transmission are assigned to the other. Separating the interrupts in this way allows different priorities to be set for transmission and reception interrupts.

The transmission interrupt vector groups the transmission-related interrupts generated by the following interrupt flags:

- TXC
- TXFL
- TXOF
- CCF
- TXIDLE

The reception interrupt on the other hand groups the reception-related interrupts, triggered by the following interrupt flags:

- RXFL
- RXFULL
- RXOF
- RXUF
- PERR
- FERR
- MPAF
- START
- SIGF
- AUTOBAUDDONE
- RXTO

If EUSARTn interrupts are enabled, an interrupt will be made if one or more of the interrupt flags in EUSARTn_IF and their corresponding bits in EUSARTn_IEN are set. All interrupts can serve as wake up interupts if enabled (EM2 Capable instance only).

20.3.2.16 EM2 operation (EUSART0 Only)

EUSART0 can operate in EM2 when running from an LF oscillator source (LF mode). Note that sending and receiving data in EM2 requires that the EUSART be connected to GPIO that are capable of operating in EM2. This includes all pins on Port A and Port B. Pins on Port C and Port D are not available for digital peripheral signalling in EM2 or EM3.

EM2 operation allows the EUSART to wait for an incoming UART frame, or even wait on the programmable start or signal frames while the system is consumiung very little energy. When a UART frame is completely received, or a start/signal frame is detected, the CPU can quickly be woken up. Alternatively, multiple frames can be transferred via the Direct Memory Access (DMA) module into RAM memory before waking up the CPU. Similarly, data can be transmitted in EM2 with data from the CPU or through use of the DMA.

All interrupts can be used as wake up interrupts if enabled. None of the interrupts are sticky, i.e., the interrupt triggers only once whenever the interrupt condition is reached.

Note: When RXDMAWU or TXDMAWU is set in EUSART0_CFG1, the system will not be able to go to EM2 before all related EUSART0 DMA requests have been processed. This means that if RXDMAWU is set and the EUSART receives a frame, the system will wait to go to EM2 before the frame has been read from the EUSART. In order for the system to go back to EM2 during or after the final transmission (i.e. when DMA will no add more data to the TX FIFO), the wake request to DMA must be removed. There are two methods for doing this:

1. If RX does not need to remain active, software can disable the peripheral and clear the TXDMAWU bit in the ISR to prevent further DMA requests. The peripheral may be re-enabled after TXDMAWU is cleared. Note that while the peripheral is disabled, the EUSART cannot receive any new data, so this option should only be used if no data is expected.
2. If RX must remain active, it is recommended to disable TX, and then write dummy information into the FIFO until the TXFL flag will no longer trigger a new wakeup. This will prevent new DMA requests.

20.3.2.17 PRS

All PRS inputs are synchronized to the peripheral clock (clk_per) for the EUSART instance.

- RX PRS: Input goes to RX module for data reception if RXPRSEN set to 1
- TRIGGER PRS: Can be used to Enable TX and/or RX if TXTEN/RXTEN are set to 1

All PRS outputs come from clk_per flops.

- PRS_TX: Same as TX output (tx_out) of EUSART
- PRS_TXC: Same as the TXC status flag
- PRS_IRDA_TX: Registered (clk_per) version of IRHF output
- PRS_RXFL: Set if (rxdata_fifo_cnt > RXFIW)
- PRS_RTS: Same as RTS output (rts_out) of EUSART

20.3.2.18 IrDA operation

The EUSART supports IrDA operation using both HF and LF clocks. For IrDA with HF clock, the controls are given in IRHFCFG register. For IrDA with LF clock (32.768 kHz), only RX is supported and the controls are given in IRLFCFG register. Note that OVS must be disabled for LF operation.

Note: Note that break generation/ detection feature is not supported while IrDA is enabled, i.e., either IRHFEN or IRLFEN is set.

20.3.2.18.1 IRHF

The IRHF modulator implements the physical layer of the IrDA specification, which is necessary for communication over IrDA. The modulator takes the signal output from the EUSART module, and modulates it before it leaves the EUSART. In the same way, the input signal is demodulated before it enters the actual EUSART module. The modulator implements the original Rev. 1.0 physical layer and one high speed extension which supports speeds from 2.4 kbps to 1.152 Mbps.

The data from and to the EUSART is represented in a NRZ (Non Return to Zero) format, where the signal value is at the same level through the entire bit period. For IrDA, the required format is RZI (Return to Zero Inverted), a format where a “1” is signalled by holding the line low, and a “0” is signalled by a short high pulse. An example is given in Figure 20.13 EUSART Example RZI Signal for a given EUSART Frame on page 628.

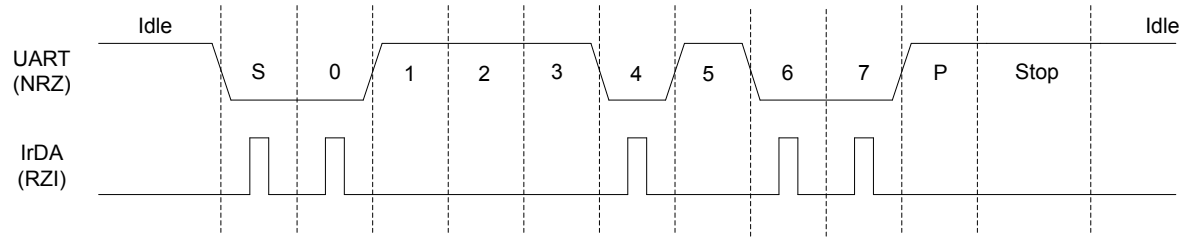


Figure 20.13. EUSART Example RZI Signal for a given EUSART Frame

The IrDA HF module is enabled by setting IRHFEN in IRHFCFG. The EUSART transmitter output and receiver input is then routed through the IrDA HF modulator.

The width of the pulses generated by the IrDA HF modulator is set by configuring IRHFPW in IRHFCFG register. Four pulse widths are available, each defined relative to the configured bit period as listed in Table 20.9 EUSART IrDA Pulse Widths on page 628.

Table 20.9. EUSART IrDA Pulse Widths

IRHFPW	Pulse width OVS=0	Pulse width OVS=1	Pulse width OVS=2	Pulse width OVS=3
00	1/16	1/8	1/6	1/4
01	2/16	2/8	2/6	N/A
10	3/16	3/8	N/A	N/A
11	4/16	N/A	N/A	N/A

By default, no filter is enabled in the IrDA HF demodulator. A filter can be enabled by setting IRHFFILT in IRHFCFG. When the filter is enabled, an incoming pulse has to last for 5 consecutive clock cycles to be detected by the IrDA demodulator. When the filter is enabled, the minimum clock frequency required is based on the baud rate and OVS chosen. The frequency requirements are listed in table Table 20.10 EUSART IrDA IRHFFILT=1, Min Clock Frequency Requirement (MHz) on page 628.

Table 20.10. EUSART IrDA IRHFFILT=1, Min Clock Frequency Requirement (MHz)

OVS	2.4 kb/s	9.6 kb/s	19.2 kb/s	38.4 kb/s	57.6 kb/s	115.2 kb/s	0.576 Mb/s	1.152 Mb/s
3 (x4)	1.0 MHz	1.0 MHz	1.0 MHz	1.0 MHz	1.4 MHz	2.8 MHz	13.8 MHz	27.6 MHz
2 (x6)	1.0 MHz	1.0 MHz	1.0 MHz	1.4 MHz	2.1 MHz	4.1 MHz	20.7 MHz	41.5 MHz
1 (x8)	1.0 MHz	1.0 MHz	1.0 MHz	1.0 MHz	1.4 MHz	2.8 MHz	13.8 MHz	27.6 MHz
0 (x16)	1.0 MHz	1.0 MHz	1.0 MHz	1.0 MHz	1.1 MHz	2.2 MHz	11.1 MHz	22.1 MHz

Note that by default, the idle value of the EUSART data signal is high. This means that the IrDA modulator generates negative pulses, and the IrDA demodulator expects negative pulses. To make the IrDA module use RZI signalling, both TXINV and RXINV in EUSARTn_CFG0 must be set.

Since the incoming signal is only sampled on positive clock edges, the width of the incoming pulses must be at least two clk_per periods wide for reliable detection by the receiver.

20.3.2.18.2 IRLF

IRLF only supports RX operation. This feature will stay operational even in EM2. It is possible to cause a wake up when a certain frame is received and then switch to IRHF if TX is required. IRLFEN in IRLFCFG must be set for this to work.

20.3.3 Synchronous Operation

Synchronous mode shares some common features with asynchronous mode such as: Loopback, inversion of RX/TX, MSBF, TX/RX Interrupt watermarks. They both share the EUSARTn_TRIGCTRL, CMD, TXDATA, RXDATA, STATUS and Interrupt registers.

In synchronous mode, EUSART can be configured to work either as a main (clock driver) or secondary (clock receiver) interface through EUSARTn_CFG2.MASTER.

20.3.3.1 Frame Format

The frames used in synchronous mode need no start and stop bits since a single clock is available to all parts participating in the communication. Parity bits cannot be used in synchronous mode.

The EUSART supports frame lengths of 8 to 16 bits per frame. Larger frames can be simulated by transmitting multiple smaller frames, i.e. a 22 bit frame can be sent using two 11-bit frames, and a 24 bit frame can be generated by transmitting three 8-bit frames. The number of bits in a frame is set using DATABITS in EUSARTn_FRAMECFG.

The frames in synchronous mode are by default transmitted with the least significant bit first like in asynchronous mode. The bit-order can be reversed by setting MSBF in EUSARTn_CFG0.

The frame format used by the transmitter can be inverted by setting TXINV in EUSARTn_CFG0, and the format expected by the receiver can be inverted by setting RXINV, also in EUSARTn_CFG0.

20.3.3.2 Clock Generation

The bit-rate in synchronous mode is given by Figure 20.14 EUSART Synchronous Mode Bit Rate on page 630. The clock division is derived from EUSARTn_CFG2.SDIV, which is applicable when acting as a Main interface only.

$$br = f_{HFPERCLK}/(1 + EUSARTn\_CFG2.SDIV)$$

Figure 20.14. EUSART Synchronous Mode Bit Rate

Given a desired baud rate brdesired, the clock divider EUSARTn_CFG2.SDIV can be calculated using Figure 20.15 EUSART Synchronous Mode Clock Division Factor on page 630

$$EUSARTn\_CFG2.SDIV = (f_{HFPERCLK}/brdesired - 1)$$

Figure 20.15. EUSART Synchronous Mode Clock Division Factor

On every clock edge data on the data lines, MOSI and MISO, is either set up or sampled. When CLKPHA in EUSARTn_CTRL is cleared, data is sampled on the leading clock edge and set-up is done on the trailing edge. If CLKPHA is set however, data is set-up on the leading clock edge, and sampled on the trailing edge. In addition to this, the polarity of the clock signal can be changed by setting CLKPOL in EUSARTn_CTRL, which also defines the idle state of the clock. This results in four different modes which are summarized in Table 20.11 EUSART SPI Modes on page 630. Figure 20.16 EUSART SPI Timing on page 630 shows the resulting timing of data set-up and sampling relative to the bus clock.

Table 20.11. EUSART SPI Modes

SPI mode	CLKPOL	CLKPHA	Leading edge	Trailing edge
0	0	0	Rising, sample	Falling, set-up
1	0	1	Rising, set-up	Falling, sample
2	1	0	Falling, sample	Rising, set-up
3	1	1	Falling, set-up	Rising, sample

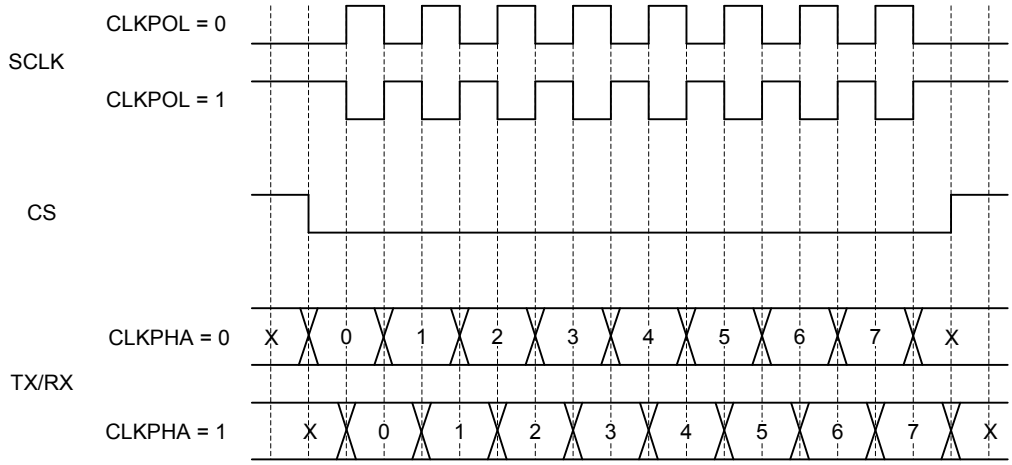


Figure 20.16. EUSART SPI Timing

The RX overflow interrupt flag, RXOF, is set at the end of the overflow frame if the receive FIFO is full. When a transfer has been performed, interrupt flags TXC are updated on the first setup clock edge of the succeeding frame, or when CS is deasserted.

20.3.3.3 Main SPI Interface Mode (Clock Driver)

EUSART operating as a main SPI interface is available only in EM0/EM1 with an HF clock source selected. When configured as a main interface, the EUSART is in full control of the data flow on the synchronous bus. When operating in full duplex mode, the secondary devices cannot transmit data to the main device without the main device transmitting to the secondary. The main device outputs the bus clock on SCLK.

Communication starts whenever there is data in the transmit FIFO and the transmitter is enabled. The EUSART clock then starts, and the main device shifts bits out from the transmit shift register using the internal clock.

When there are no more frames in the transmit FIFO and the transmit shift register is empty, the clock stops, and communication ends. When the receiver is enabled, it samples incoming data when the transmitter transmits data. Operation of the RX and TX FIFOs is as in asynchronous mode.

EUSARTn_CFG2.RXBLOCK can be used to block incoming RX data from pushing into RX FIFO.

20.3.3.4 Operation of CS Pin

When operating as a main interface, the CS pin can have one of two functions, or it can be disabled by clearing CSPEN in GPIO_EUSARn_ROUTEEN register.

If CS is disabled and there is a need to disable TX during the operation, then TX should be disabled at the end of transaction only indicated by the trigger of TXC interrupt to maintain synchronicity between TX and RX and avoid causing sudden stop to secondary devices.

If CS is configured as an output, it can be used to automatically generate a chip select for a single secondary device by setting AUTOCS in EUSARTn_CTRL. If AUTOCS is set, CS is activated before a transmission begins, and deactivated after the last bit has been transmitted and there is no more data in the transmit FIFO.

The time duration between assertion of CS and the start of transmission can be controlled using CSSETUP in EUSARTn_TIMINGCFG. If new data is ready for transmission before CS is deasserted, the data is sent without deasserting CS in between. CSHOLD in EUSARTn_TIMINGCFG keeps CS asserted after the end of frame for the number of baud-times specified.

By default, CS is active low, but its polarity can be inverted by setting CSINV in EUSARTn_CFG2.

20.3.3.5 AUTOTX

The main device on a synchronous bus is required to transmit data (send a clock) to a secondary device in order to receive data from that device. In some cases, only a few words are transmitted and a lot of data is then received from the secondary device. In that case, one solution is to keep feeding the TX with data to transmit, but that consumes system bandwidth. Instead AUTOTX can be used.

When AUTOTX in EUSARTn_CFG2 is set and TX FIFO is filled with initial data, EUSART will fully transmit the loaded data and then continue transmitting the last sent bit as long as there is available space in the RX FIFO for the chosen frame size. This happens even though there is no more data in the TX FIFO. The TX underflow interrupt flag TXUF in EUSARTn_IF is set on the first word that is transmitted which does not contain valid data.

During AUTOTX the EUSART will always send the previous sent bit, thus reducing the number of transitions on the TX output. So if the last bit sent was a 0, 0's will be sent during AUTOTX and if the last bit sent was a 1, 1's will be sent during AUTOTX.

20.3.3.6 Secondary SPI Interface Mode (Clock Receiver)

When the EUSART is in secondary interface mode, data transmission is not controlled by the EUSART, but by an external main SPI device. The EUSART is therefore not able to initiate a transmission, and has no control over the number of bytes written to the external main device.

The output and input to the EUSART are also swapped when in secondary mode, making the receiver take its input from TX (MOSI) and the transmitter drive RX (MISO).

To transmit data when in secondary mode, the device must load data into the transmit FIFO and enable the transmitter. The data will remain in the EUSART until the main device starts a transmission by pulling the CS input low and transmitting data. For every frame transmitted from main to secondary device, a frame is transferred from secondary to main as well.

If the transmitter is enabled in synchronous secondary mode and the main device starts transmission of a frame, the underflow interrupt flag TXUF in EUSARTn_IF will be set if no data is available for transmission. At the same time, the secondary device will transmit the default TX data, which can be set through EUSARTn_DTXDATCFG for the current and subsequent frames until the FIFO is filled. Note that when TX is enabled (with or without data in TXFIFO) in the middle of transaction, TXUF can be triggered if it's transmitting default TX data.

Similar to when operating as a synchronous main interface, EUSARTn_CFG2.RXBLOCK can be used to block incoming RX data from pushing into RX FIFO.

If the secondary device needs to control its own chip select signal, this can be achieved by clearing CSPEN in the GPIO_EUSARTn_ROUTEEN register. The internal chip select signal can then be controlled through CSINV in the CTRL register. The chip select signal will be CSINV inverted, i.e. if CSINV is cleared, the chip select is active and vice versa. In such cases, SCLK could arrive anytime that the device doesn't have prior notification from CS. Hence, EUSARTn_CFG2.FORCELOAD bit can be used to control how the device transmits the first dataword. If this bit is not set, the next outgoing dataword will be a DEFAULT TX data even if the FIFO had been loaded before SCLK arrives, followed by the loaded TX data. EUSARTn_IF.LOADERRIF Interrupt will never be triggered when EUSARTn_CFG2.FORCELOAD is not set. If this bit is set, as soon as the transmitter becomes ready the shift register will be loaded immediately and send data once SCLK arrives. The transmitter becomes ready when TX is enabled and TXFIFO is filled. On top of that, at word-boundary it will automatically trigger setup window check against the programmed EUSARTn_TIMINGCFG.SETUPWINDOW, which specifies the minimum duration (in APB bus clock cycles) between transmitter becoming ready and the incoming SCLK. If the measured duration is less than SETUPWINDOW bus clock cycles, an EUSARTn_IF.LOADERRIF Interrupt will be triggered. Besides, if the transmitter is enabled or disabled or the empty TXFIFO is loaded not at word-boundary during a transaction (SCLK is toggling), LOADERRIF will also be triggered immediately without checking for SETUPWINDOW. It's recommended that the transmitter should be ready while SCLK is idling or at word-boundary and at sufficient margin before SCLK toggles. Once LOADERRIF Interrupt is set, it may require to reset the secondary interface by disabling the module and re-enabling it because the transmitted data could be undeterministic.

20.3.4 Debug Halt

When DBGHALT in EUSART_CTRLX is clear, RTS is only dependent on the RX FIFO having space available to receive data. Incoming data is always received until both the RX FIFO is full and the RX shift register is full regardless of the state of DBGHALT or chip halt. Additional incoming data is discarded. When DBGHALT is set, RTS deasserts on RX FIFO full or when chip halt is high. However, a low pulse detected on chip halt will keep RTS asserted when no frame is being received. At the start of frame reception, RTS will deassert if chip halt is high and DBGHALT is set. This behavior allows single stepping to pulse the chip halt low for a cycle, and receive the next frame. The link partner must stop transmitting when RTS is deasserted, or the RX FIFO could overflow. All data in the transmit FIFO is sent out even when chip halt is asserted; therefore, the DMA will need to be set to stop sending the EUSART TX data during chip halt.

20.3.5 PRS-triggered Transmissions

If a transmission must be started on an event with very little delay, the PRS system can be used to trigger the transmission. The PRS channel to use as a trigger can be selected using TSEL in EUSARTn_TRIGCTRL. When a positive edge is detected on this signal, the receiver is enabled if RXTEN in EUSARTn_TRIGCTRL is set, and the transmitter is enabled if TXTEN in EUSARTn_TRIGCTRL is set. Only one signal input is supported by the EUSART.

AUTOTXTEN can also be combined with TXTEN to make the EUSART transmit a command to the external device prior to clocking out data. To do this, disable TX using the TXDIS command, load the TX FIFO with the command and enable AUTOTXTEN and TXTEN. When the selected PRS input goes high, the EUSART will now transmit the loaded command, and then continue clocking out while both the PRS input is high and there is room in the RX FIFO

20.3.6 PRS RX Input

The EUSART can be configured to receive data directly from a PRS channel by setting RXPRS in EUSARTn_INPUT. The PRS channel used is selected using RXPRSSEL in EUSARTn_INPUT. This way, for example, a differential RX signal can be input to the ACMP and the output routed via PRS to the EUSART.

20.3.7 PRS CLK Input

The EUSART can be configured to receive clock directly from a PRS channel by setting CLKPRS in EUSARTn_INPUT. The PRS channel used is selected using CLKPRSSEL in EUSARTn_INPUT. This is useful in secondary synchronous mode and can together with RX PRS input be used to input data from PRS.

20.3.8 DMA Support

The EUSART has full DMA support. The DMA controller can write to the transmit FIFO using the registers EUSARTn_TXDATA and it can read from the receive FIFO using the registers EUSARTn_RXDATA. This enables single byte transfers, 9 bit data + control/status bits, double byte and double byte + control/status transfers both to and from the EUSART.

A request for the DMA controller to read from the EUSART receive FIFO can come from the following source:

- Receive FIFO level satisfying EUSARTn_CFG1.RXFIW setting

A write request can come from one of the following sources:

- Transmit FIFO level satisfying EUSARTn_CFG1.TXFIW setting.

Even though there are two sources for write requests to the DMA, only one should be used at a time, since the requests from both sources are cleared even though only one of the requests are used.

In some cases, it may be sensible to temporarily stop DMA access to the EUSART when an error such as a framing error has occurred. This is enabled by setting ERRSDMA in EUSARTn_CTRL.

Note: For Synchronous mode full duplex operation, if both receive FIFO and transmit FIFO are served by DMA, to make sure receive FIFO is not overflowed the settings below should be followed.

- The DMA channel that serves receive FIFO should have higher priority than the DMA channel that serves transmit FIFO.
- IGNORESREQ should be set for both DMA channel.

20.4 EUSART Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	EUSART_IPVERSION	R	IP Version ID
0x004	EUSART_EN	RW ENABLE	Enable Register
0x008	EUSART_CFG0	RW CONFIG	Configuration 0 Register
0x00C	EUSART_CFG1	RW CONFIG	Configuration 1 Register
0x010	EUSART_CFG2	RW CONFIG	Configuration 2 Register
0x014	EUSART_FRAMECFG	RW CONFIG	Frame Format Register
0x018	EUSART_DTXDATCFG	RW CONFIG	Default TX DATA Register
0x01C	EUSART_IRHFCFG	RW CONFIG	HF IrDA Mod Config Register
0x020	EUSART_IRLFCFG	RW CONFIG	LF IrDA Pulse Config Register
0x024	EUSART_TIMINGCFG	RW CONFIG	Timing Register
0x028	EUSART_STARTFRAMECFG	RW CONFIG	Start Frame Register
0x02C	EUSART_SIGFRAMECFG	RW CONFIG	Signal Frame Register
0x030	EUSART_CLKDIV	RWH LFSYNC	Clock Divider Register
0x034	EUSART_TRIGCTRL	RW LFSYNC	Trigger Control Register
0x038	EUSART_CMD	W LFSYNC	Command Register
0x03C	EUSART_RXDATA	RH	RX Data Register
0x040	EUSART_RXDATAP	RH	RX Data Peek Register
0x044	EUSART_TXDATA	W	TX Data Register
0x048	EUSART_STATUS	RH	Status Register
0x04C	EUSART_IF	RWH INTFLAG	Interrupt Flag Register
0x050	EUSART_IEN	RW	Interrupt Enable Register
0x054	EUSART_SYNCBUSY	RH	Synchronization Busy Register
0x1000	EUSART_IPVERSION_SET	R	IP Version ID
0x1004	EUSART_EN_SET	RW ENABLE	Enable Register
0x1008	EUSART_CFG0_SET	RW CONFIG	Configuration 0 Register
0x100C	EUSART_CFG1_SET	RW CONFIG	Configuration 1 Register
0x1010	EUSART_CFG2_SET	RW CONFIG	Configuration 2 Register
0x1014	EUSART_FRAMECFG_SET	RW CONFIG	Frame Format Register
0x1018	EUSART_DTXDATCFG_SET	RW CONFIG	Default TX DATA Register
0x101C	EUSART_IRHFCFG_SET	RW CONFIG	HF IrDA Mod Config Register
0x1020	EUSART_IRLFCFG_SET	RW CONFIG	LF IrDA Pulse Config Register
0x1024	EUSART_TIMINGCFG_SET	RW CONFIG	Timing Register
0x1028	EUSART_STARTFRA-MECFG_SET	RW CONFIG	Start Frame Register
0x102C	EUSART_SIGFRAMECFG_SET	RW CONFIG	Signal Frame Register

Offset	Name	Type	Description
0x1030	EUSART_CLKDIV_SET	RWH LFSYNC	Clock Divider Register
0x1034	EUSART_TRIGCTRL_SET	RW LFSYNC	Trigger Control Register
0x1038	EUSART_CMD_SET	W LFSYNC	Command Register
0x103C	EUSART_RXDATA_SET	RH	RX Data Register
0x1040	EUSART_RXDATAP_SET	RH	RX Data Peek Register
0x1044	EUSART_TXDATA_SET	W	TX Data Register
0x1048	EUSART_STATUS_SET	RH	Status Register
0x104C	EUSART_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1050	EUSART_IEN_SET	RW	Interrupt Enable Register
0x1054	EUSART_SYNCBUSY_SET	RH	Synchronization Busy Register
0x2000	EUSART_IPVERSION_CLR	R	IP Version ID
0x2004	EUSART_EN_CLR	RW ENABLE	Enable Register
0x2008	EUSART_CFG0_CLR	RW CONFIG	Configuration 0 Register
0x200C	EUSART_CFG1_CLR	RW CONFIG	Configuration 1 Register
0x2010	EUSART_CFG2_CLR	RW CONFIG	Configuration 2 Register
0x2014	EUSART_FRAMECFG_CLR	RW CONFIG	Frame Format Register
0x2018	EUSART_DTXDATCFG_CLR	RW CONFIG	Default TX DATA Register
0x201C	EUSART_IRHFCFG_CLR	RW CONFIG	HF IrDA Mod Config Register
0x2020	EUSART_IRLFCFG_CLR	RW CONFIG	LF IrDA Pulse Config Register
0x2024	EUSART_TIMINGCFG_CLR	RW CONFIG	Timing Register
0x2028	EUSART_STARTFRA-MECFG_CLR	RW CONFIG	Start Frame Register
0x202C	EUSART_SIGFRAMECFG_CLR	RW CONFIG	Signal Frame Register
0x2030	EUSART_CLKDIV_CLR	RWH LFSYNC	Clock Divider Register
0x2034	EUSART_TRIGCTRL_CLR	RW LFSYNC	Trigger Control Register
0x2038	EUSART_CMD_CLR	W LFSYNC	Command Register
0x203C	EUSART_RXDATA_CLR	RH	RX Data Register
0x2040	EUSART_RXDATAP_CLR	RH	RX Data Peek Register
0x2044	EUSART_TXDATA_CLR	W	TX Data Register
0x2048	EUSART_STATUS_CLR	RH	Status Register
0x204C	EUSART_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2050	EUSART_IEN_CLR	RW	Interrupt Enable Register
0x2054	EUSART_SYNCBUSY_CLR	RH	Synchronization Busy Register
0x3000	EUSART_IPVERSION_TGL	R	IP Version ID
0x3004	EUSART_EN_TGL	RW ENABLE	Enable Register
0x3008	EUSART_CFG0_TGL	RW CONFIG	Configuration 0 Register
0x300C	EUSART_CFG1_TGL	RW CONFIG	Configuration 1 Register

Offset	Name	Type	Description
0x3010	EUSART_CFG2_TGL	RW CONFIG	Configuration 2 Register
0x3014	EUSART_FRAMECFG_TGL	RW CONFIG	Frame Format Register
0x3018	EUSART_DTXDATCFG_TGL	RW CONFIG	Default TX DATA Register
0x301C	EUSART_IRHFCFG_TGL	RW CONFIG	HF IrDA Mod Config Register
0x3020	EUSART_IRLFCFG_TGL	RW CONFIG	LF IrDA Pulse Config Register
0x3024	EUSART_TIMINGCFG_TGL	RW CONFIG	Timing Register
0x3028	EUSART_STARTFRA-MECFG_TGL	RW CONFIG	Start Frame Register
0x302C	EUSART_SIGFRAMECFG_TGL	RW CONFIG	Signal Frame Register
0x3030	EUSART_CLKDIV_TGL	RWH LFSYNC	Clock Divider Register
0x3034	EUSART_TRIGCTRL_TGL	RW LFSYNC	Trigger Control Register
0x3038	EUSART_CMD_TGL	W LFSYNC	Command Register
0x303C	EUSART_RXDATA_TGL	RH	RX Data Register
0x3040	EUSART_RXDATAP_TGL	RH	RX Data Peek Register
0x3044	EUSART_TXDATA_TGL	W	TX Data Register
0x3048	EUSART_STATUS_TGL	RH	Status Register
0x304C	EUSART_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3050	EUSART_IEN_TGL	RW	Interrupt Enable Register
0x3054	EUSART_SYNCBUSY_TGL	RH	Synchronization Busy Register

20.5 EUSART Register Description

20.5.1 EUSART_IPVERSION - IP Version ID

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

20.5.2 EUSART_EN - Enable Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status**
	Disabling status when the module is disabled.			
0	EN	0x0	RW	**Module enable**
	Set to enable the module.			

20.5.3 EUSART_CFG0 - Configuration 0 Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0						0x0	0x0	0x0		0x0			0x0			0x0	0x0			0x0			0x0			0x0	0x0	0x0	0x0	0x0
Access	RW	RW						RW	RW	RW		RW			RW			RW	RW			RW			RW			RW	RW	RW	RW	RW
Name	AUTOBAUDEN	MVDIS						ERRSTX	ERRSRX	ERRSDMA		SKIPPERRF			AUTOTRI			TXINV	RXINV			MSBF			OVS			MPAB	MPM	CCEN	LOOPBK	SYNC

Bit	Name	Reset	Access	Description
31	AUTOBAUDEN	0x0	RW	**AUTOBAUD detection enable**
	Detects the baud rate based on receiving a 0x55 frame (0x00 for IrDA). Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
30	MVDIS	0x0	RW	**Majority Vote Disable**
	Disable majority vote for 16x, 8x and 6x oversampling modes. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
29:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24	ERRSTX	0x0	RW	**Disable TX On Error**
	When set, the transmitter is disabled on framing and parity errors in the receiver. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			

Value	Mode	Description
0	DISABLE	Received framing and parity errors have no effect on transmitter
1	ENABLE	Received framing and parity errors disable the transmitter

Bit	Name	Reset	Access	Description
23	ERRSRX	0x0	RW	**Disable RX On Error**
	When set, the receiver is disabled on framing and parity errors. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			

Value	Mode	Description
0	DISABLE	Framing and parity errors have no effect on receiver
1	ENABLE	Framing and parity errors disable the receiver

Bit	Name	Reset	Access	Description
22	ERRSDMA	0x0	RW	**Halt DMA Read On Error**
	When set, DMA read requests will be cleared on framing and parity errors. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			

Value	Mode	Description
0	DISABLE	Framing and parity errors have no effect on DMA requests from the EUSART
1	ENABLE	DMA requests from the EUSART are blocked while the PERR or FERR interrupt flags are set

Bit	Name	Reset	Access	Description
21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

Bit	Name	Reset	Access	Description
20	SKIPPERRF	0x0	RW	**Skip Parity Error Frames**
	When set, the receiver discards frames with parity errors. The PERR interrupt flag is still set. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
19:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	AUTOTRI	0x0	RW	**Automatic TX Tristate**
	When enabled, TXTRI is set by hardware whenever the transmitter is idle, and TXTRI is cleared by hardware when transmission starts. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			

Value	Mode	Description
0	DISABLE	The output on UARTn_TX when the transmitter is idle is defined by TXINV
1	ENABLE	UARTn_TX is tristated whenever the transmitter is idle

16:15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14	TXINV	0x0	RW	**Transmitter output Invert**
	The output from the EUSART transmitter can optionally be inverted by setting this bit.			

Value	Mode	Description
0	DISABLE	Output from the transmitter is passed unchanged to UARTn_TX
1	ENABLE	Output from the transmitter is inverted before it is passed to UARTn_TX

13	RXINV	0x0	RW	**Receiver Input Invert**
	Setting this bit will invert the input to the EUSART receiver.			

Value	Mode	Description
0	DISABLE	Input is passed directly to the receiver
1	ENABLE	Input is inverted before it is passed to the receiver

12:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	MSBF	0x0	RW	**Most Significant Bit First**
	Decides whether data is sent with the least significant bit first, or the most significant bit first.			

Value	Mode	Description
0	DISABLE	Data is sent with the least significant bit first
1	ENABLE	Data is sent with the most significant bit first

9:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:5	OVS	0x0	RW	**Oversampling**
	Sets the number of clock periods in a EUSART bit-period. More clock cycles gives better robustness, while less clock cycles gives better performance. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			

Value	Mode	Description

Bit	Name	Reset	Access	Description
	0	X16		16X oversampling
	1	X8		8X oversampling
	2	X6		6X oversampling
	3	X4		4X oversampling
	4	DISABLE		Disable oversampling (for LF operation)
4	MPAB	0x0	RW	**Multi-Processor Address-Bit**
	Defines the value of the multi-processor address bit. An incoming frame with its 9th bit equal to the value of this bit marks the frame as a multi-processor address frame. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
3	MPM	0x0	RW	**Multi-Processor Mode**
	Multi-processor mode uses the 9th bit of the EUSART frames to tell whether the frame is an address frame or a data frame. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
	Value	Mode		Description
	0	DISABLE		The 9th bit of incoming frames has no special function
	1	ENABLE		An incoming frame with the 9th bit equal to MPAB will be loaded into the RX FIFO regardless of RXBLOCK and will result in the MPAB interrupt flag being set
2	CCEN	0x0	RW	**Collision Check Enable**
	Enables collision checking on data when operating in half duplex modus. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
	Value	Mode		Description
	0	DISABLE		Collision check is disabled
	1	ENABLE		Collision check is enabled. The receiver must be enabled for the check to be performed
1	LOOPBK	0x0	RW	**Loopback Enable**
	Allows the receiver to be connected directly to the EUSART transmitter for loopback and half duplex communication.			
	Value	Mode		Description
	0	DISABLE		The receiver is connected to and receives data from UARTn_RX
	1	ENABLE		The receiver is connected to and receives data from UARTn_TX
0	SYNC	0x0	RW	**Synchronous Mode**
	Determines whether the EUSART is operating in asynchronous or synchronous mode. When switching between SYNC and ASYNC mode, module Disablement is required.			
	Value	Mode		Description
	0	ASYNC		The EUSART operates in asynchronous mode
	1	SYNC		The EUSART operates in synchronous mode

20.5.4 EUSART_CFG1 - Configuration 1 Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0x0					0x0						0x0				0x0				0x0	0x0	0x0			0x0			0x0	0x0	0x0	0x0
Access		RW					RW						RW				RW				RW	RW	RW			RW			RW	RW	RW	RW
Name		RXFIW					RTSRXFW						TXFIW				RXPRSEN				SFUBRX	RXDMAWU	TXDMAWU			RXTIMEOUT			RTSINV	CTSEN	CTSINV	DBGHALT

Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
30:27	RXFIW	0x0	RW	**RX FIFO Interrupt Watermark**

Determines the interrupt and status level of the Receive FIFO. Also impacts RX DMA request.

Value	Mode	Description
0	ONEFRAME	RXFL status flag and IF are set when the RX FIFO has at least one frame in it.
1	TWOFRAMES	RXFL status flag and IF are set when the RX FIFO has at least two frames in it.
2	THREEFRAMES	RXFL status flag and IF are set when the RX FIFO has at least three frames in it.
3	FOURFRAMES	RXFL status flag and IF are set when the RX FIFO has at least four frames in it.
4	FIVEFRAMES	RXFL status flag and IF are set when the RX FIFO has at least five frames in it.
5	SIXFRAMES	RXFL status flag and IF are set when the RX FIFO has at least six frames in it.
6	SEVENFRAMES	RXFL status flag and IF are set when the RX FIFO has at least seven frames in it.
7	EIGHTFRAMES	RXFL status flag and IF are set when the RX FIFO has at least eight frames in it.
8	NINEFRAMES	RXFL status flag and IF are set when the RX FIFO has at least nine frames in it.
9	TENFRAMES	RXFL status flag and IF are set when the RX FIFO has at least ten frames in it.
10	ELEVENFRAMES	RXFL status flag and IF are set when the RX FIFO has at least eleven frames in it.
11	TWELVEFRAMES	RXFL status flag and IF are set when the RX FIFO has at least twelve frames in it.
12	THIRTEENFRAMES	RXFL status flag and IF are set when the RX FIFO has at least thriteen frames in it.
13	FOURTEENFRAMES	RXFL status flag and IF are set when the RX FIFO has at least fourteen frames in it.

Bit	Name	Reset	Access	Description
	14	FIFTEENFRAMES		RXFL status flag and IF are set when the RX FIFO has at least fifteen frames in it.
	15	SIXTEENFRAMES		RXFL status flag and IF are set when the RX FIFO has at least sixteen frames in it.
26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25:22	RTSRXFW	0x0	RW	**Request-to-send RX FIFO Watermark**
	Set Request-to-send watermark level. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
	Value	Mode		Description
	0	ONEFRAME		RTS is set if there is space for at least one more frame in the RX FIFO.
	1	TWOFRAMES		RTS is set if there is space for at least two more frames in the RX FIFO.
	2	THREEFRAMES		RTS is set if there is space for at least three more frames in the RX FIFO.
	3	FOURFRAMES		RTS is set if there is space for four more frames in the RX FIFO.
	4	FIVEFRAMES		RTS is set if there is space for five more frames in the RX FIFO.
	5	SIXFRAMES		RTS is set if there is space for six more frames in the RX FIFO.
	6	SEVENFRAMES		RTS is set if there is space for seven more frames in the RX FIFO.
	7	EIGHTFRAMES		RTS is set if there is space for eight more frames in the RX FIFO.
	8	NINEFRAMES		RTS is set if there is space for nine more frames in the RX FIFO.
	9	TENFRAMES		RTS is set if there is space for ten more frames in the RX FIFO.
	10	ELEVENFRAMES		RTS is set if there is space for eleven more frames in the RX FIFO.
	11	TWELVEFRAMES		RTS is set if there is space for twelve more frames in the RX FIFO.
	12	THIRTEENFRAMES		RTS is set if there is space for thirteen more frames in the RX FIFO.
	13	FOURTEENFRAMES		RTS is set if there is space for fourteen more frames in the RX FIFO.
	14	FIFTEENFRAMES		RTS is set if there is space for fifteen more frames in the RX FIFO.
	15	SIXTEENFRAMES		RTS is set if there is space for sixteen more frames in the RX FIFO.
21:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	TXFIW	0x0	RW	**TX FIFO Interrupt Watermark**
	Determines the interrupt and status level of the transmit FIFO. Also impacts TX DMA request.			
	Value	Mode		Description

Bit	Name	Reset	Access	Description
	0	ONEFRAME		TXFL status flag and IF are set when the TX FIFO has space for at least one more frame.
	1	TWOFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least two more frames.
	2	THREEFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least three more frames.
	3	FOURFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least four more frames.
	4	FIVEFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least five more frames.
	5	SIXFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least six more frames.
	6	SEVENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least seven more frames.
	7	EIGHTFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least eight more frames.
	8	NINEFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least nine more frames.
	9	TENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least ten more frames.
	10	ELEVENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least eleven more frames.
	11	TWELVEFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least twelve more frames.
	12	THIRTEENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least thriteen more frames.
	13	FOURTEENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least fourteen more frames.
	14	FIFTEENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least fifteen more frames.
	15	SIXTEENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least sixteen more frames.
15	RXPRSEN	0x0	RW	**PRS RX Enable** When set, the PRS channel selected as input to RX.
14:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	SFUBRX	0x0	RW	**Start Frame Unblock Receiver** Set to unblock RX on Start frame reception. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.
10	RXDMAWU	0x0	RW	**Receiver DMA Wakeup** Set to enable wakeup from EM2 to EM1 for DMA/ RX interaction. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.
9	TXDMAWU	0x0	RW	**Transmitter DMA Wakeup** Set to enable wakeup from EM2 to EM1 for DMA/ TX interaction. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.

Bit	Name	Reset	Access	Description
8:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	RXTIMEOUT	0x0	RW	**RX Timeout**

When enabled, determines how long, in units of frame, RX needs to remain idle after a frame reception before RXTOIF gets set. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.

Value	Mode	Description
0	DISABLED	
1	ONEFRAME	
2	TWOFRAMES	
3	THREEFRAMES	
4	FOURFRAMES	
5	FIVEFRAMES	
6	SIXFRAMES	
7	SEVENFRAMES	

3	RTSINV	0x0	RW	**Request-to-send Invert Enable**

When set, the RTS pin polarity is inverted. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.

Value	Mode	Description
0	DISABLE	The RTS pin is active low
1	ENABLE	The RTS pin is active high

2	CTSEN	0x0	RW	**Clear-to-send Enable**

When set, frames in the TX FIFO will not be sent until link partner asserts CTS. Any data in the TX shift register will continue transmitting, the next TX FIFO data will not load into the TX shift register. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.

Value	Mode	Description
0	DISABLE	Ignore CTS
1	ENABLE	Stop transmitting when CTS is inactive

1	CTSINV	0x0	RW	**Clear-to-send Invert Enable**

When set, the CTS pin polarity is inverted. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.

Value	Mode	Description
0	DISABLE	The CTS pin is active low
1	ENABLE	The CTS pin is active high

0	DBGHALT	0x0	RW	**Debug halt**

Set to halt operation when core is halted. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.

Value	Mode	Description
0	DISABLE	Continue normal EUSART operation even if core is halted

Bit	Name	Reset	Access	Description
	1	ENABLE		If core is halted, receive one frame and then halt reception by deactivating RTS. Next frame reception happens when the core is unhalted during single stepping.

20.5.5 EUSART_CFG2 - Configuration 2 Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																								0x0	0x0	0x1	0x0	0x0	0x0	0x0	0x0
Access	RW																								RW	RW	RW	RW	RW	RW	RW	RW
Name	SDIV																								FORCELOAD	CLKPRSEN	AUTOCS	AUTOTX	CSINV	CLKPHA	CLKPOL	MASTER

Bit	Name	Reset	Access	Description
31:24	SDIV	0x0	RW	**Sync Clock Div**
	Sets the clock rate for synchronous main mode operation only (To set the clock rate for asynchronous operation, see the CLKDIV field). Clock division value = SDIV + 1. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'MASTER'.			
23:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	FORCELOAD	0x0	RW	**Force Load to Shift Register**
	When this bit is set, if TXEN is already enabled, any loading of an empty FIFO will immediately load the data into the shift register. This bit is recommended to be used in 3-wire setting where CS is not used, or in a custom 4-wire mode where SCLK is in long idle state between two transactions with CS always in active mode (this typically happens between 2 transactions where the main interface holds SCLK idling to give additional time for the secondary interface to load data). If FORCELOAD bit is not set, the next out-going frame will be a DEFAULT TX data, followed by the loaded TX data in TX FIFO. When FORCELOAD is set, it will automatically trigger setup window check against the programmed EUSARTn_TIMINGCFG.SETUPWINDOW, which specifies the minimum duration between empty fifo loading event and first encountered edge of SCLK. If the measured duration is less than SETUPWINDOW bus clock cycles, a EUSARTn_IF.LOADERRIF Interrupt will be triggered. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'SLAVE'.			
6	CLKPRSEN	0x0	RW	**PRS CLK Enable**
	When set, the PRS channel selected as input to CLK. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'SLAVE'.			
5	AUTOCS	0x1	RW	**Automatic Chip Select**
	When enabled, the output on CS will be activated one baud-period before transmission starts, and deactivated when transmission ends. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'MASTER'.			
4	AUTOTX	0x0	RW	**Always Transmit When RXFIFO Not Full**
	Transmits as long as RXFIFO is not full. If TX is empty, underflows are generated. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'MASTER'.			
3	CSINV	0x0	RW	**Chip Select Invert**
	Default value is active low. This affects both the selection of external secondary devices, as well as the selection of the microcontroller in secondary mode. Only applicable when CFG0.SYNC bit is set to 'SYNC'.			

Value	Mode	Description
0	AL	Chip select is active low
1	AH	Chip select is active high

Bit	Name	Reset	Access	Description
2	CLKPHA	0x0	RW	**Clock Edge for Setup/Sample**

Bit	Name	Reset	Access	Description
	Determines where data is set-up and sampled according to the bus clock when in synchronous mode. Only applicable when CFG0.SYNC bit is set to 'SYNC'.			

Value	Mode	Description
0	SAMPLELEADING	Data is sampled on the leading edge and set-up on the trailing edge of the bus clock in synchronous mode
1	SAMPLETRAILING	Data is set-up on the leading edge and sampled on the trailing edge of the bus clock in synchronous mode

Bit	Name	Reset	Access	Description
1	CLKPOL	0x0	RW	**Clock Polarity**
	Determines the clock polarity of the bus clock used in synchronous mode. Only applicable when CFG0.SYNC bit is set to 'SYNC'.			

Value	Mode	Description
0	IDLELOW	The bus clock used in synchronous mode has a low base value
1	IDLEHIGH	The bus clock used in synchronous mode has a high base value

Bit	Name	Reset	Access	Description
0	MASTER	0x0	RW	**Main mode**
	Set this bit to put EUSART to main interface mode. When unset, EUSART operates in secondary interface mode. When changing between Main and Secondary mode, module Disablement is required. Only applicable when CFG0.SYNC bit is set to 'SYNC'.			

Value	Mode	Description
0	SLAVE	Secondary mode
1	MASTER	Main mode

20.5.6 EUSART_FRAMECFG - Frame Format Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																			0x1				0x0						0x2			
Access																			RW				RW						RW			
Name																			STOPBITS				PARITY						DATABITS			

Bit	Name	Reset	Access	Description
31:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:12	STOPBITS	0x1	RW	**Stop-Bit Mode**

Determines the number of stop-bits used. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.

Value	Mode	Description
0	HALF	The transmitter generates a half stop bit. Stop-bits are not verified by receiver
1	ONE	One stop bit is generated and verified
2	ONEANDAHALF	The transmitter generates one and a half stop bit. The receiver verifies the first stop bit
3	TWO	The transmitter generates two stop bits. The receiver checks the first stop-bit only

11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	PARITY	0x0	RW	**Parity-Bit Mode**

Determines whether parity bits are enabled, and whether even or odd parity should be used. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.

Value	Mode	Description
0	NONE	Parity bits are not used
2	EVEN	Even parity are used. Parity bits are automatically generated and checked by hardware.
3	ODD	Odd parity is used. Parity bits are automatically generated and checked by hardware.

7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	DATABITS	0x2	RW	**Data-Bit Mode**

Sets the number of data bits in a EUSART frame.

Value	Mode	Description
1	SEVEN	Each frame contains 7 data bits
2	EIGHT	Each frame contains 8 data bits

Bit	Name	Reset	Access	Description
	3	NINE		Each frame contains 9 data bits
	4	TEN		Each frame contains 10 data bits
	5	ELEVEN		Each frame contains 11 data bits
	6	TWELVE		Each frame contains 12 data bits
	7	THIRTEEN		Each frame contains 13 data bits
	8	FOURTEEN		Each frame contains 14 data bits
	9	FIFTEEN		Each frame contains 15 data bits
	10	SIXTEEN		Each frame contains 16 data bits

20.5.7 EUSART_DTXDATCFG - Default TX DATA Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	RW															
Name																	DTXDAT															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	DTXDAT	0x0	RW	**Default TX DATA**
	This is the default transmitted data when the TXFIFO is empty. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'SLAVE'.			

20.5.8 EUSART_IRHFCFG - HF IrDA Mod Config Register

Offset	Bit Position																															
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0		0x0
Access																													RW	RW		RW
Name																													IRHFFILT	IRHFPW		IRHFEN

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	IRHFFILT	0x0	RW	**IrDA RX Filter**
	Set to enable filter on demodulator.			

Value	Mode	Description
0	DISABLE	No filter enabled
1	ENABLE	Filter enabled. IrDA pulse must be high for at least 5 consecutive clock cycles to be detected

Bit	Name	Reset	Access	Description
2:1	IRHFPW	0x0	RW	**IrDA TX Pulse Width**
	Configure the pulse width generated by the modulator as a fraction of the configured EUSART bit period.			

Value	Mode	Description
0	ONE	IrDA pulse width is 1/16 for OVS=0 and 1/8 for OVS=1
1	TWO	IrDA pulse width is 2/16 for OVS=0 and 2/8 for OVS=1
2	THREE	IrDA pulse width is 3/16 for OVS=0 and 3/8 for OVS=1
3	FOUR	IrDA pulse width is 4/16 for OVS=0 and 4/8 for OVS=1

Bit	Name	Reset	Access	Description
0	IRHFEN	0x0	RW	**Enable IrDA Module**
	Enable IrDA module and route EUSART signals through it. Used when EUSART has HF clock. Only applicable when CFG0.SYNC bit is set to 'ASYNC'			

20.5.9 EUSART_IRLFCFG - LF IrDA Pulse Config Register

Offset	Bit Position																															
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																IRLFEN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	IRLFEN	0x0	RW	**Pulse Generator/Extender Enable**
		Filter EUSART output through pulse generator and the EUSART input through the pulse extender. Used for LF operation. Only applicable when CFG0.SYNC bit is set to 'ASYNC'		

20.5.10 EUSART_TIMINGCFG - Timing Register

Offset	Bit Position																															
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x5					0x0				0x0				0x0					0x0	
Access													RW					RW				RW				RW					RW	
Name													SETUPWINDOW					ICS				CSHOLD				CSSETUP					TXDELAY	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	SETUPWINDOW	0x5	RW	**Setup Window**
	When CFG2.FORCELOAD is set, this defines the number of bus clock cycles that empty FIFO load or enabling of TX or disabling of TX must be performed before the sampling edge of SCLK at word-boundary to avoid load error. Word boundary is defined as followings:before the transaction starts, or between 2 transactions or the first bit between 2 datawords. If baud-rate is more than 5 MHz, a value of 4 is recommended, any values smaller than that can be tried out but avoid using 0. If baud-rate is less than 5 MHz, value of 5 is recommended, any values higher than 5 can be used but it may make the load error easy to occur. The recommended values for frequency bands should be sufficient to work all the time. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'SLAVE' and CFG2.FORCELOAD is set.			
15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14:12	ICS	0x0	RW	**Inter-Character Spacing**
	Inter-character spacing after each TX frame while the TX FIFO is not empty. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'MASTER'.			

Value	Mode	Description
0	ZERO	There is no space between charcters
1	ONE	Create a space of 1 baud-times between frames
2	TWO	Create a space of 2 baud-times between frames
3	THREE	Create a space of 3 baud-times between frames
4	FOUR	Create a space of 4 baud-times between frames
5	FIVE	Create a space of 5 baud-times between frames
6	SIX	Create a space of 6 baud-times between frames
7	SEVEN	Create a space of 7 baud-times between frames

11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10:8	CSHOLD	0x0	RW	**Chip Select Hold**
	Chip Select will be de-asserted after the end of frame transmission. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'MASTER'.			

Value	Mode	Description

Bit	Name	Reset	Access	Description
	0	ZERO		CS is de-asserted half or 1 baud-time after the end of transmission depending on CLKPHASE equal to 1 or 0 respectively
	1	ONE		CS is de-asserted 1 additional baud-time after the end of transmission
	2	TWO		CS is de-asserted 2 additional baud-times after the end of transmission
	3	THREE		CS is de-asserted 3 additional baud-times after the end of transmission
	4	FOUR		CS is de-asserted 4 additional baud-times after the end of transmission
	5	FIVE		CS is de-asserted 5 additional baud-times after the end of transmission
	6	SIX		CS is de-asserted 6 additional baud-times after the end of transmission
	7	SEVEN		CS is de-asserted 7 additional baud-times after the end of transmission
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	CSSETUP	0x0	RW	**Chip Select Setup**
	Chip Select will be asserted before the start of frame transmission. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'MASTER'.			
	Value	Mode		Description
	0	ZERO		CS is asserted half or 1 baud-time before the start of transmission depending on CLKPHASE equal to 1 or 0 respectively
	1	ONE		CS is asserted 1 additional baud-time before start of transmission
	2	TWO		CS is asserted 2 additional baud-times before start of transmission
	3	THREE		CS is asserted 3 additional baud-times before start of transmission
	4	FOUR		CS is asserted 4 additional baud-times before start of transmission
	5	FIVE		CS is asserted 5 additional baud-times before start of transmission
	6	SIX		CS is asserted 6 additional baud-times before start of transmission
	7	SEVEN		CS is asserted 7 additional baud-times before start of transmission
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	TXDELAY	0x0	RW	**TX Delay Transmission**
	Configurable delay before new rtansfers. Frames sent back-to-back are not delayed. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
	Value	Mode		Description

Bit	Name	Reset	Access	Description
	0	NONE		Frames are transmitted immediately.
	1	SINGLE		Transmission of new frames is delayed by a single bit period.
	2	DOUBLE		Transmission of new frames is delayed by a two bit periods.
	3	TRIPPLE		Transmission of new frames is delayed by a three bit periods.

20.5.11 EUSART_STARTFRAMECFG - Start Frame Register

Offset	Bit Position																																
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0				
Access																												RW				
Name																												STARTFRAME				

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	STARTFRAME	0x0	RW	**Start Frame**
	When a frame matching STARTFRAME is received, the receiver detects that and STARTF interrupt flag is set. If SFUBRX is set, RXBLOCK is cleared and the start frame is loaded in to the RX FIFO. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			

20.5.12 EUSART_SIGFRAMECFG - Signal Frame Register

Offset	Bit Position																																
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0				
Access																												RW				
Name																												SIGFRAME				

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	SIGFRAME	0x0	RW	**Signal Frame Value**
	When a frame matching SIGFRAME is detected by the receiver, SIGF interrupt flag is set. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			

20.5.13 EUSART_CLKDIV - Clock Divider Register

Offset	Bit Position																															
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset										0x0																						
Access										RW																						
Name										DIV																						

Bit	Name	Reset	Access	Description
31:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:3	DIV	0x0	RW	**Fractional Clock Divider**
	Specifies the fractional clock divider. Setting AUTOBAUDEN in CFG1 register will overwrite the DIV field. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
2:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

20.5.14 EUSART_TRIGCTRL - Trigger Control Register

Offset	Bit Position																															
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														AUTOTXTEN	TXTEN	RXTEN

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	AUTOTXTEN	0x0	RW	**AUTOTX Trigger Enable**
	When set, AUTOTX is enabled as long as the selected PRS channel has a high value. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'MASTER'.			
1	TXTEN	0x0	RW	**Transmit Trigger Enable**
	When set, the positive edge of the selected PRS channel sets TXEN, enabling the transmitter.			
0	RXTEN	0x0	RW	**Receive Trigger Enable**
	When set, the positive edge of the selected PRS channel sets RXEN, enabling the receiver.			

20.5.15 EUSART_CMD - Command Register

Offset	Bit Position																														
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																								W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)
Name																								CLEARTX	TXTRIDIS	TXTRIEN	RXBLOCKDIS	RXBLOCKEN	TXDIS	TXEN	RXDIS	RXEN

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	CLEARTX	0x0	W(nB)	**Clear TX FIFO**
	Set to clear TX FIFO. Only applicable when CFG0.SYNC bit is set to 'ASYNC'. Note that before issuing this command, firmware should first set the transmitter disable bit (CMD.TXDIS) and then poll the transmitter enable status bit until cleared (STATUS.TXENS).			
7	TXTRIDIS	0x0	W(nB)	**Transmitter Tristate Disable**
	Disables tristating of the transmitter output.			
6	TXTRIEN	0x0	W(nB)	**Transmitter Tristate Enable**
	Tristates the transmitter output. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
5	RXBLOCKDIS	0x0	W(nB)	**Receiver Block Disable**
	Set to clear RXBLOCK, resulting in all incoming frames being loaded into the RX FIFO.			
4	RXBLOCKEN	0x0	W(nB)	**Receiver Block Enable**
	Set to set RXBLOCK, resulting in all incoming frames being discarded.			
3	TXDIS	0x0	W(nB)	**Transmitter Disable**
	Set to disable transmission. STATUS.TXENS should be polled to ensure disabled status in Synchrounous mode.			
2	TXEN	0x0	W(nB)	**Transmitter Enable**
	Set to enable data transmission. STATUS.TXENS should be polled to ensure enabled status in Synchrounous mode.			
1	RXDIS	0x0	W(nB)	**Receiver Disable**
	Set to disable data reception. If a frame is under reception when the receiver is disabled, the incoming frame is discarded. STATUS.RXENS should be polled to ensure disabled status in Synchrounous mode.			
0	RXEN	0x0	W(nB)	**Receiver Enable**
	Set to activate data reception. STATUS.RXENS should be polled to ensure enabled status in Synchrounous mode.			

20.5.16 EUSART_RXDATA - RX Data Register

Offset	Bit Position																													
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									RXDATA							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	RXDATA	0x0	R	**RX Data and Control bits**
	Use this register to access data read from the EUSART, cleared on read access. In Synchronous mode, Bit 15:0 is actual RXDATA. In Asynchronous mode, Bit 8:0 is actual RXDATA, bit 9 is PERR (set if received data has a parity error.), bit 10 is FERR (set if received data has a framing error, can be result of a break condition). Note that FIFO is not retained in EM2.			

20.5.17 EUSART_RXDATAP - RX Data Peek Register

Offset	Bit Position																													
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									RXDATAP							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	RXDATAP	0x0	R	**RX Data Peek**
	Use this register to access data read from the EUSART without popping the FIFO.			

20.5.18 EUSART_TXDATA - TX Data Register

Offset	Bit Position																													
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W							
Name																									TXDATA							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	TXDATA	0x0	W	**TX Data and Control bits**
	Use this register to write data to the EUSART. If TXEN is set, a transfer will be initiated at the first opportunity. In Synchronous mode, 15:0 is TXDATA. In asynchronous mode, 8:0 is TXDATA, bit 9 is UBRXAT (Set to clear RXBLOCK after transmission, unblocking the receiver), bit 10 is TXTRIAT (Set to tristate transmitter by setting TXTRI after tranmission), bit 11 is TXBREAK (Set to send data as a break. Recipient will see a framing error or a break condition depending on its configuration and the value of TXDATA), bit 12 is TXDISAT (Set to disable trasmitter and release data bus directly after transmission), bit 13 is RXENAT (Set to enable reception after transmission). Note that FIFO is not retained in EM2			

20.5.19 EUSART_STATUS - Status Register

Offset	Bit Position																													
0x048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset							0x0	0x0						0x0					0x1	0x1				0x0	0x0	0x1	0x0	0x0	0x0		0x0	0x0
Access							R	R						R					R	R				R	R	R	R	R	R		R	R
Name							CLEARTXBUSY	AUTOBAUDDONE						TXFCNT					TXIDLE	RXIDLE				RXFULL	RXFL	TXFL	TXC	TXTRI	RXBLOCK		TXENS	RXENS

Bit	Name	Reset	Access	Description
31:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25	CLEARTXBUSY	0x0	R	**TX FIFO Clear Busy**
	After issuing CLEARTX command, wait on this status flag until it goes low.			
24	AUTOBAUDDONE	0x0	R	**Auto Baud Rate Detection Completed**
	Set when auto baud rate has been detected and CLKDIV has been updated with required value. If AUTOBAUDEN is not set in CFG0 register, this bit is always read as '0'. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
23:21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20:16	TXFCNT	0x0	R	**Valid entries in TX FIFO**
	Count of TX valid FIFO entries.			
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	TXIDLE	0x1	R	**TX Idle**
	Set when TX idle. In Synchronous secondary mode, TX is not considered idle when transmitting Default TX data.			
12	RXIDLE	0x1	R	**RX Idle**
	Set when RX is idle.			
11:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	RXFULL	0x0	R	**RX FIFO Full**
	Set when the RX FIFO is full.			
7	RXFL	0x0	R	**RX FIFO Level**
	Set when data is available in the RX FIFO. Depends on the RXFIW setting in the CFG1 register.			
6	TXFL	0x1	R	**TX FIFO Level**
	Set when there is space for data in the TX FIFO. Depends on the TXFIW setting in CFG1 register.			
5	TXC	0x0	R	**TX Complete**
	Set when a transmission has completed and no more data is available in the TX FIFO and shift register.			
4	TXTRI	0x0	R	**Transmitter Tristated**

Bit	Name	Reset	Access	Description
	Set when the transmitter is tristated, and cleared when transmitter output is enabled. If AUTOTRI in UARTn_CFG is set, then this bit is always read as 0. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
3	RXBLOCK	0x0	R	**Block Incoming Data**
	When set, the receiver discards incoming frames. An incoming frame will not be loaded into the RX FIFO if this bit is set at the instant the frame has been completely received.			
2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	TXENS	0x0	R	**Transmitter Enable Status**
	Set when the transmitter is enabled. In Synchronous secondary mode, default TX data will be transmitted when the transmitter is disabled.			
0	RXENS	0x0	R	**Receiver Enable Status**
	Set when the receiver is enabled.			

20.5.20 EUSART_IF - Interrupt Flag Register

Offset	Bit Position																													
0x04C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset							0x0	0x0					0x0	0x0		0x0			0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access							RW	RW					RW	RW		RW			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name							RXTO	AUTOBAUDDONE					SIGF	STARTF		CSWU			TXIDLE	CCF	LOADERR	MPAF	FERR	PERR	TXUF	TXOF	RXUF	RXOF	RXFULL	RXFL	TXFL	TXC

Bit	Name	Reset	Access	Description
31:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25	RXTO	0x0	RW	**RX Timeout Interrupt Flag**
	Set when RX timeout occurs. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
24	AUTOBAUDDONE	0x0	RW	**Auto Baud Complete Interrupt Flag**
	Set when auto baud rate detection is complete. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19	SIGF	0x0	RW	**Signal Frame Interrupt Flag**
	Set when a signal frame is detected. Please note that when MPA, START, and SIGNAL are set to match the same frame, corresponding interrupts might get triggered in arbitrary sequence due to synchronization uncertainty. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
18	STARTF	0x0	RW	**Start Frame Interrupt Flag**
	Set when a start frame is detected. Please note that when MPA, START, and SIGNAL are set to match the same frame, corresponding interrupts might get triggered in arbitrary sequence due to synchronization uncertainty. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	CSWU	0x0	RW	**CS Wake-up Interrupt Flag**
	Set when the CS asserts. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'SLAVE'.			
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	TXIDLE	0x0	RW	**TX Idle Interrupt Flag**
	Set when TX goes idle. In Synchronous mode, if TX is disabled during the middle of the transactions, TXIDLEIF won't be triggered when the engine becomes disabled.			
12	CCF	0x0	RW	**Collision Check Fail Interrupt Flag**
	Set when a collision check notices an error in the transmitted data. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
11	LOADERR	0x0	RW	**Load Error Interrupt Flag**

Bit	Name	Reset	Access	Description
	Set when the empty TX FIFO is loaded less than the required TIMINGCFG.SETUPWINDOW bus clock cycles before the first edge of the incoming SCLK. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'SLAVE' and CFG2.FORCELOAD is set.			
10	MPAF	0x0	RW	**Multi-Processor Address Frame Interrupt**
	Set when a multi-processor address frame is detected. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
9	FERR	0x0	RW	**Framing Error Interrupt Flag**
	Set when a frame with a framing error is received while RXBLOCK is cleared. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
8	PERR	0x0	RW	**Parity Error Interrupt Flag**
	Set when a frame with a parity error is received while RXBLOCK is cleared. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
7	TXUF	0x0	RW	**TX FIFO Underflow Interrupt Flag**
	In Sync secondary mode, Set when reading an empty TX FIFO with TX enabled. In Synchronous main Mode with AUTOTX enabled, set when transmitting the word that does not contain valid data.			
6	TXOF	0x0	RW	**TX FIFO Overflow Interrupt Flag**
	Set when a write is done to the TX FIFO while it is full. The data already in the TX FIFO is preserved.			
5	RXUF	0x0	RW	**RX FIFO Underflow Interrupt Flag**
	Set when trying to read from the RX FIFO when it is empty.			
4	RXOF	0x0	RW	**RX FIFO Overflow Interrupt Flag**
	Set when data is complettly received in the receive shift register but the RX FIFO is full. RX FIFO is not overwirtten by new data.			
3	RXFULL	0x0	RW	**RX FIFO Full Interrupt Flag**
	Set when the RX FIFO becomes full.			
2	RXFL	0x0	RW	**RX FIFO Level Interrupt Flag**
	Set when data becomes available in the RX FIFO. This field depends on the RXFIW field in the CFG1 register.			
1	TXFL	0x0	RW	**TX FIFO Level Interrupt Flag**
	Set when space becomes available in the TX FIFO. This depends on the TXFIW field in the CFG1 register.			
0	TXC	0x0	RW	**TX Complete Interrupt Flag**
	This interrupt is set after a transmission when both the TX FIFO and shift register are empty.			

20.5.21 EUSART_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset							0x0	0x0					0x0	0x0		0x0			0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access							RW	RW					RW	RW		RW			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name							RXTO	AUTOBAUDDONE					SIGF	STARTF		CSWU			TXIDLE	CCF	LOADERR	MPAF	FERR	PERR	TXUF	TXOF	RXUF	RXOF	RXFULL	RXFL	TXFL	TXC

Bit	Name	Reset	Access	Description
31:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25	RXTO	0x0	RW	**RX Timeout Enable**
	Interrupt enable for RXTOIF.			
24	AUTOBAUDDONE	0x0	RW	**Auto Baud Complete Enable**
	Interrupt enable for AUTOBAUDDONEIF.			
23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19	SIGF	0x0	RW	**Signal Frame Enable**
	Interrupt enable for SIGFIF.			
18	STARTF	0x0	RW	**Start Frame Enable**
	Interrupt enable for STARTFIF.			
17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	CSWU	0x0	RW	**CS Wake-up Enable**
	Interrupt enable for CSWUIF.			
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	TXIDLE	0x0	RW	**TX IDLE Enable**
	Interrupt enable for TXIDLEIF.			
12	CCF	0x0	RW	**Collision Check Fail Enable**
	Interrupt enable for CCFIF.			
11	LOADERR	0x0	RW	**Load Error Enable**
	Interrupt enable for LOADERRIF.			
10	MPAF	0x0	RW	**Multi-Processor Addr Frame Enable**
	Interrupt enable for MPAFIF.			
9	FERR	0x0	RW	**Framing Error Enable**
	Interrupt enable for FERRIF.			

Bit	Name	Reset	Access	Description
8	PERR	0x0	RW	**Parity Error Enable**
	Interrupt enable for PERRIF.			
7	TXUF	0x0	RW	**TX FIFO Underflow Enable**
	Interrupt enable for TXUFIF.			
6	TXOF	0x0	RW	**TX FIFO Overflow Enable**
	Interrupt enable for TXOFIF.			
5	RXUF	0x0	RW	**RX FIFO Underflow Enable**
	Interrupt enable for RXUFIF.			
4	RXOF	0x0	RW	**RX FIFO Overflow Enable**
	Interrupt enable for RXOFIF.			
3	RXFULL	0x0	RW	**RX FIFO Full Enable**
	Interrupt enable for RXFULLIF.			
2	RXFL	0x0	RW	**RX FIFO Level Enable**
	Interrupt enable for RXFLIF.			
1	TXFL	0x0	RW	**TX FIFO Level Enable**
	Interrupt enable for TXFLIF.			
0	TXC	0x0	RW	**TX Complete Enable**
	Interrupt enable for TXCIF.			

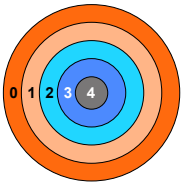
20.5.22 EUSART_SYNCBUSY - Synchronization Busy Register

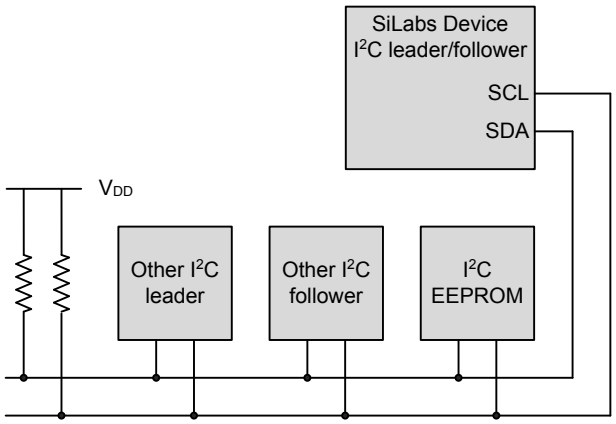
Offset	Bit Position																													
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																					R	R	R	R	R	R	R	R	R	R	R	R
Name																					AUTOTXTEN	TXTRIDIS	TXTRIEN	RXBLOCKDIS	RXBLOCKEN	TXDIS	TXEN	RXDIS	RXEN	TXTEN	RXTEN	DIV

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	AUTOTXTEN	0x0	R	**SYNCBUSY for AUTOTXTEN in TRIGCTRL**
	This bit is set when there is an ongoing synchronization of AUTOTXTEN field. Do not do another write to the same field while this bit is set.			
10	TXTRIDIS	0x0	R	**SYNCBUSY in TXTRIDIS in CMD**
	This bit is set when there is an ongoing synchronization of TXTRIDIS field. Do not do another write to the same field while this bit is set. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
9	TXTRIEN	0x0	R	**SYNCBUSY for TXTRIEN in CMD**
	This bit is set when there is an ongoing synchronization of TXTRIEN field. Do not do another write to the same field while this bit is set.			
8	RXBLOCKDIS	0x0	R	**SYNCBUSY for RXBLOCKDIS in CMD**
	This bit is set when there is an ongoing synchronization of RXBLOCKDIS field. Do not do another write to the same field while this bit is set.			
7	RXBLOCKEN	0x0	R	**SYNCBUSY for RXBLOCKEN in CMD**
	This bit is set when there is an ongoing synchronization of RXBLOCKEN field. Do not do another write to the same field while this bit is set.			
6	TXDIS	0x0	R	**SYNCBUSY for TXDIS in CMD**
	This bit is set when there is an ongoing synchronization of TXDIS field. Do not do another write to the same field while this bit is set.			
5	TXEN	0x0	R	**SYNCBUSY for TXEN in CMD**
	This bit is set when there is an ongoing synchronization of TXEN field. Do not do another write to the same field while this bit is set.			
4	RXDIS	0x0	R	**SYNCBUSY for RXDIS in CMD**
	This bit is set when there is an ongoing synchronization of RXDIS field. Do not do another write to the same field while this bit is set.			
3	RXEN	0x0	R	**SYNCBUSY for RXEN in CMD**
	This bit is set when there is an ongoing synchronization of RXEN field. Do not do another write to the same field while this bit is set.			
2	TXTEN	0x0	R	**SYNCBUSY for TXTEN in TRIGCTRL**
	This bit is set when there is an ongoing synchronization of TXTEN field. Do not do another write to the same field while this bit is set.			

Bit	Name	Reset	Access	Description
1	RXTEN	0x0	R	**SYNCBUSY for RXTEN in TRIGCTRL** This bit is set when there is an ongoing synchronization of RXTEN field. Do not do another write to the same field while this bit is set.
0	DIV	0x0	R	**SYNCBUSY for DIV in CLKDIV** This bit is set when there is an ongoing synchronization of DIV field. Do not do another write to the same field while this bit is set.

21. I2C - Inter-Integrated Circuit Interface





Quick Facts

What?

The I²C interface allows communication on I²C-buses with the lowest energy consumption possible.

Why?

I²C is a popular serial bus that enables communication with a number of external devices using only two I/O pins.

How?

With the help of DMA, the I²C interface allows I²C communication with minimal CPU intervention. Address recognition is available in all energy modes (except EM4), allowing the MCU to wait for data on the I²C-bus with sub-µA current consumption.

21.1 Introduction

The I²C module provides an interface between the MCU and a serial I²C-bus. It is capable of acting as both a leader and a follower and supports multi-leader buses. Standard-mode, fast-mode and fast-mode plus speeds are supported, allowing transmission rates all the way from 10 kbit/s up to 1 Mbit/s. Follower arbitration and timeouts are also provided to allow implementation of an SMBus compliant system. The interface provided to software by the I²C module allows precise control of the transmission process and highly automated transfers. Automatic recognition of follower addresses is provided in all energy modes (except EM4).

21.2 Features

- True multi-leader capability
- Support for different bus speeds
  - Standard-mode (Sm) bit rate up to 100 kbit/s
  - Fast-mode (Fm) bit rate up to 400 kbit/s
  - Fast-mode Plus (Fm+) bit rate up to 1 Mbit/s
- Arbitration for both leader and follower (allows SMBus ARP)
- Clock synchronization and clock stretching
- Hardware address recognition
  - 7-bit masked address
  - General call address
  - Supported in EM2/3 (I2C0-only)
- 10-bit address support
- Error handling
  - Clock low timeout
  - Bus idle (clock high) timeout
  - Arbitration lost
  - Bus error detection
- Separate receive/ transmit 2-level buffers, with additional separate shift registers
- Full DMA support

21.3 Functional Description

An overview of the I2C module is shown in Figure 21.1 I2C Overview on page 668.

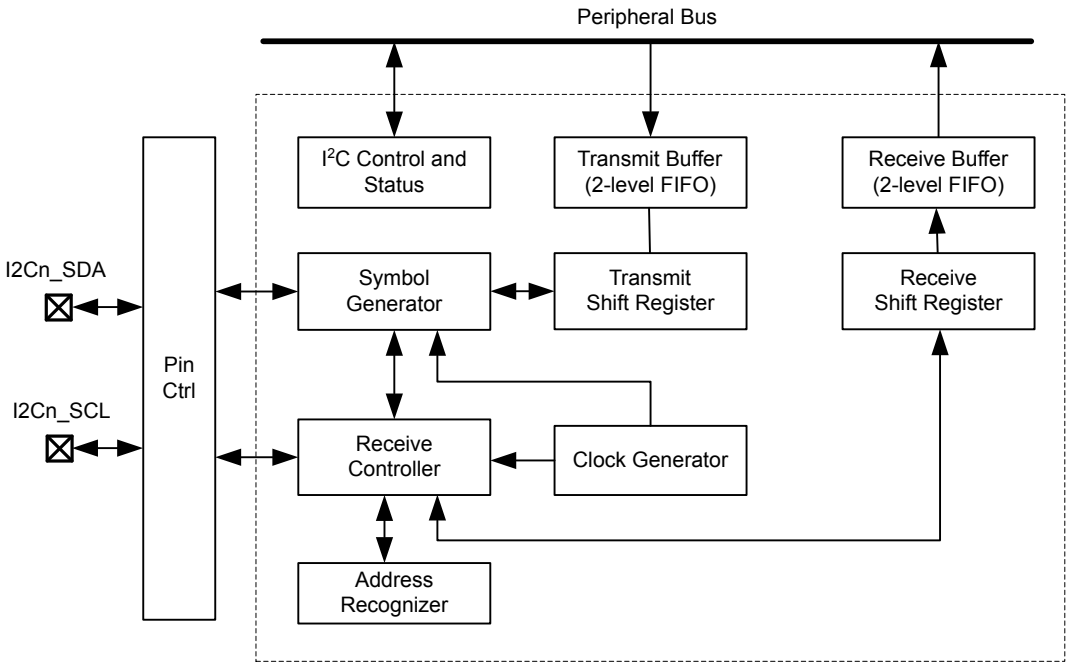


Figure 21.1. I2C Overview

21.3.1 I2C-Bus Overview

The I²C-bus uses two wires for communication; a serial data line (SDA) and a serial clock line (SCL) as shown in Figure 21.2 I2C-Bus Example on page 669. As a true multi-leader bus it includes collision detection and arbitration to resolve situations where multiple leaders transmit data at the same time without data loss.

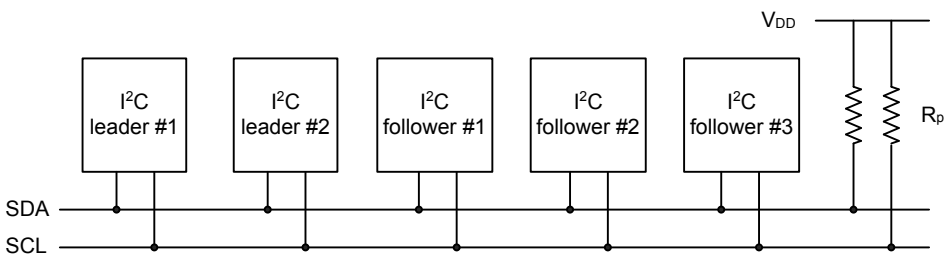


Figure 21.2. I2C-Bus Example

Each device on the bus is addressable by a unique address, and an I²C leader can address all the devices on the bus, including other leaders.

Both the bus lines are open-drain. The maximum value of the pull-up resistor can be calculated as a function of the maximal rise-time **tr** for the given bus speed, and the estimated bus capacitance **Cb** as shown in Figure 21.3 I2C Pull-up Resistor Equation on page 669.

$$R_{p(max)} = t_r/(0.8473 \times C_b).$$

Figure 21.3. I2C Pull-up Resistor Equation

The maximal rise times for 100 kHz, 400 kHz and 1 MHz I²C are 1 µs, 300 ns and 120 ns respectively.

Note: The GPIO slew rate control should be set for the desired slew rate..

Note: If V_{dd} drops below the voltage on SCL and SDA lines, the MCU could become back powered and pull the SCL and SDA lines low.

21.3.1.1 START and STOP Conditions

START and STOP conditions are used to initiate and stop transactions on the I²C-bus. All transactions on the bus begin with a START condition (S) and end with a STOP condition (P). As shown in Figure 21.4 I2C START and STOP Conditions on page 670, a START condition is generated by pulling the SDA line low while SCL is high, and a STOP condition is generated by pulling the SDA line high while SCL is high.

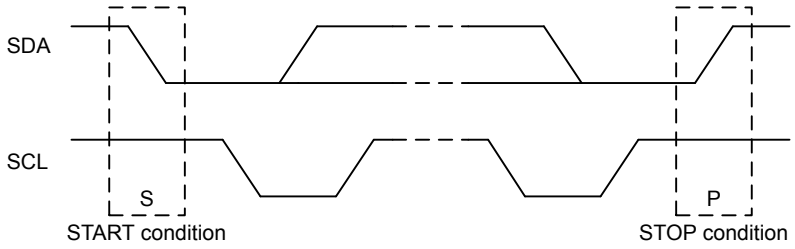


Figure 21.4. I2C START and STOP Conditions

The START and STOP conditions are easily identifiable bus events as they are the only conditions on the bus where a transition is allowed on SDA while SCL is high. During the actual data transmission, SDA is only allowed to change while SCL is low, and must be stable while SCL is high. One bit is transferred per clock pulse on the I²C-bus as shown in Figure 21.5 I2C Bit Transfer on I²C-Bus on page 670.

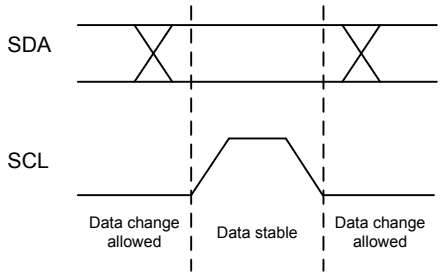


Figure 21.5. I2C Bit Transfer on I²C-Bus

21.3.1.2 Bus Transfer

When a leader wants to initiate a transfer on the bus, it waits until the bus is idle and transmits a START condition on the bus. The leader then transmits the address of the follower it wishes to interact with and a single R/W bit telling whether it wishes to read from the follower (R/W bit set to 1) or write to the follower (R/W bit set to 0).

After the 7-bit address and the R/W bit, the leader releases the bus, allowing the follower to acknowledge the request. During the next bit-period, the follower pulls SDA low (ACK) if it acknowledges the request, or keeps it high if it does not acknowledge it (NACK).

Following the address acknowledge, either the follower or leader transmits data, depending on the value of the R/W bit. After every 8 bits (one byte) transmitted on the SDA line, the transmitter releases the line to allow the receiver to transmit an ACK or a NACK. Both the data and the address are transmitted with the most significant bit first.

The number of bytes in a bus transfer is unrestricted. The leader ends the transmission after a (N)ACK by sending a STOP condition on the bus. After a STOP condition, any leader wishing to initiate a transfer on the bus can try to gain control of it. If the current leader wishes to make another transfer immediately after the current, it can start a new transfer directly by transmitting a repeated START condition (Sr) instead of a STOP followed by a START.

Examples of I^2C transfers are shown in Figure 21.6 I2C Single Byte Write to Follower on page 671, Figure 21.7 I2C Double Byte Read from Follower on page 671, and Figure 21.8 I2C Single Byte Write, then Repeated Start and Single Byte Read on page 671. The identifiers used are:

- ADDR - Address
- DATA - Data
- S - Start bit
- Sr - Repeated start bit
- P - Stop bit
- W/R - Read(1)/Write(0)
- A - ACK
- N - NACK

Figure 21.6. I2C Single Byte Write to Follower

Figure 21.7. I2C Double Byte Read from Follower

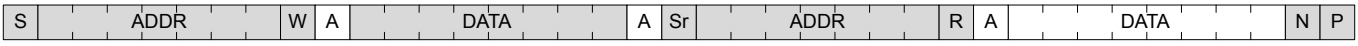


Figure 21.8. I2C Single Byte Write, then Repeated Start and Single Byte Read

21.3.1.3 Addresses

I²C supports both 7-bit and 10-bit addresses. When using 7-bit addresses, the first byte transmitted after the START-condition contains the address of the follower that the leader wants to contact. In the 7-bit address space, several addresses are reserved. These addresses are summarized in Table 21.1 I2C Reserved I²C Addresses on page 672, and include a General Call address which can be used to broadcast a message to all followers on the I²C-bus.

Table 21.1. I2C Reserved I²C Addresses

I²C Address	R/W	Description
0000-000	0	General Call address
0000-000	1	START byte
0000-001	X	Reserved for the C-Bus format
0000-010	X	Reserved for a different bus format
0000-011	X	Reserved for future purposes
0000-1XX	X	Reserved for future purposes
1111-1XX	X	Reserved for future purposes
1111-0XX	X	10 Bit follower addressing mode

21.3.1.4 10-bit Addressing

To address a follower using a 10-bit address, two bytes are required to specify the address instead of one. The seven first bits of the first byte must then be 1111 0XX, where XX are the two most significant bits of the 10-bit address. As with 7-bit addresses, the eighth bit of the first byte determines whether the leader wishes to read from or write to the follower. The second byte contains the eight least significant bits of the follower address.

When a follower receives a 10-bit address, it must acknowledge both the address bytes if they match the address of the follower.

When performing a leader transmitter operation, the leader transmits the two address bytes and then the remaining data, as shown in Figure 21.9 I2C Leader Transmitter/Follower Receiver with 10-bit Address on page 672.

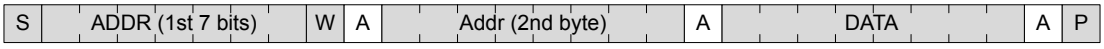


Figure 21.9. I2C Leader Transmitter/Follower Receiver with 10-bit Address

When performing a leader receiver operation however, the leader first transmits the two address bytes in a leader transmitter operation, then sends a repeated START followed by the first address byte and then receives data from the addressed follower. The follower addressed by the 10-bit address in the first two address bytes must remember that it was addressed, and respond with data if the address transmitted after the repeated start matches its own address. An example of this (with one byte transmitted) is shown in Figure 21.10 I2C Leader Receiver/Follower Transmitter with 10-bit Address on page 672.

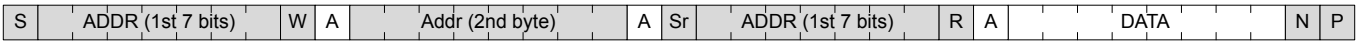


Figure 21.10. I2C Leader Receiver/Follower Transmitter with 10-bit Address

21.3.1.5 Arbitration, Clock Synchronization, Clock Stretching

Arbitration and clock synchronization are features aimed at allowing multi-leader buses. Arbitration occurs when two devices try to drive the bus at the same time. If one device drives it low, while the other drives it high, the one attempting to drive it high will not be able to do so due to the open-drain bus configuration. Both devices sample the bus, and the one that was unable to drive the bus in the desired direction detects the collision and backs off, letting the other device continue communication on the bus undisturbed.

Clock synchronization is a means of synchronizing the clock outputs from several leaders driving the bus at once, and is a requirement for effective arbitration.

Followers on the bus are allowed to force the clock output on the bus low in order to pause the communication on the bus and give themselves time to process data or perform any real-time tasks they might have. This is called clock stretching.

Arbitration is supported by the I^{2}C module for both leaders and followers. Clock synchronization and clock stretching is also supported.

21.3.2 Enable and Reset

The I^{2}C is enabled by setting the EN bit in the I2C_EN register.

To reset the internal state of the I^{2}C module and terminate any ongoing transfers, set the CORERST bit in I2C_CTRL. After resetting, the CORERST bit must be cleared to resume I^{2}C operation.

Note: When enabling the I^{2}C, the ABORT command or the Bus Idle Timeout feature must be applied prior to use even if the BUSY flag is not set.

21.3.3 Pin Configuration

The I^{2}C SDA and SCL pins are configured and enabled in the GPIO_I2Cn_ROUTEEN, GPIO_I2Cn_SCLROUTE, and GPIO_I2Cn_SDAROUTE registers.

The I^{2}C module must be configured to use pins on either Port A or B if wakeup on address recognition from EM2/3 is desired. All other ports are available only in EM0/1. See GPIO chapter for more details on Port limitations.

If the I^{2}C module is configured to use pins other than Port A or B, firmware should reset the module before entering EM2/3 by setting the CORERST bit in I2C_CTRL. After resuming EM0/1 operation, firmware should then clear CORERST.

21.3.4 Safely Disabling and Changing Follower Configuration

The I^{2}C follower is partially asynchronous, and some precautions are necessary to always ensure a safe follower disable or follower configuration change. These measures should be taken, if (while the follower is enabled) the user cannot guarantee that an address match will not occur at the exact time of follower disable or follower configuration change.

Worst case consequences for an address match while disabling follower or changing configuration is that the follower may end up in an undefined state. To reset the follower back to a known state, the EN bit in I2C_EN must be cleared. This should be done regardless of whether the follower is going to be re-enabled or not.

21.3.5 Clock Generation

The I^2C peripheral clock (I2CCLK) for I2C0 is derived from the LSPCLK, and for I2C1 is derived from the PCLK.

The SCL signal generated by the I^2C leader determines the maximum transmission rate on the bus. The clock is generated as a division of the peripheral clock (I2CCLK), and is given by the following equation:

$$f_{SCL} = f_{I2CCLK}/(((N_{low} + N_{high}) \times (DIV + 1)) + 8 + N_{fall} + N_{rise})$$

Figure 21.11. I2C Maximum Transmission Rate

Where DIV is the clock divider value set in I2C_CLKDIV, the values of N_{low} and N_{high} (and thus the ratio between the high and low parts of the clock signal) are controlled by CLHR in the I2C_CTRL register, and N_{fall} and N_{rise} represent the number of I2CCLK cycles required for clock synchronization.

The values of N_{low} and N_{high} specify the number of prescaled clock cycles in the low and high periods of the clock signal respectively. The worst case low and high periods of the signal are:

$$t_{high} >= (N_{high} \times (DIV + 1) + 4)/f_{I2CCLK}$$

$$t_{low} >= (N_{low} \times (DIV + 1) + 4)/f_{I2CCLK}$$

Figure 21.12. I2C High and Low Cycles Equations

Clock synchronization is used to ensure that requested low and high times are met on the bus. The counters establishing high and low time are only active once the pin logic has reached the high or low logic levels, and so the rise and fall times will impact the maximum transmission rate on the bus. The clock logic level is sampled at a rate of f_{I2CCLK}, and will therefore be quantized to an integer number of I2CCLK clock cycles, as:

$$N_{rise} = CEILING(t_{rise}/t_{I2CCLK})$$

$$N_{fall} = CEILING(t_{fall}/t_{I2CCLK})$$

Figure 21.13. I2C Clock Synchronization Cycles

Note that there is an inherent propagation delay between driving SCL and sampling the signal, so the above equations can result in N = 0 for fast rise or fall times.

For example, a system with a weak pull-up on SCL may result in long t_{rise}, requiring several N_{rise} synchronization cycles, and subsequently impact the I2C transmission rate. In the same system, t_{fall} may be faster than the sampling delay, resulting in N_{fall} = 0.

Note: DIV must be set to 1 during follower mode operation.

21.3.6 Arbitration

Arbitration is enabled by default, but can be disabled by setting the ARBDIS bit in I2C_CTRL. When arbitration is enabled, the value on SDA is sensed every time the I^2C module attempts to change its value. If the sensed value is different than the value the I^2C module tried to output, it is interpreted as a simultaneous transmission by another device, and that the I^2C module has lost arbitration.

Whenever arbitration is lost, the ARBLOST interrupt flag in I2C_IF is set, any lines held are released, and the I^2C device goes idle. If an I^2C leader loses arbitration during the transmission of an address, another leader may be trying to address it. The leader therefore receives the rest of the address, and if the address matches the follower address of the leader, the leader goes into either follower transmitter or follower receiver mode.

Note:

Arbitration can be lost both when operating as a leader and when operating as a follower.

21.3.7 Buffers

21.3.7.1 Transmit Buffer and Shift Register

The I²C transmitter has a 2-level FIFO transmit buffer and a transmit shift register as shown in Figure 21.1 I2C Overview on page 668. A byte is loaded into the transmit buffer by writing to I2C_TXDATA or 2 bytes can be loaded simultaneously in the transmit buffer by writing to I2C_TXDOUBLE. Figure 21.14 I2C Transmit Buffer Operation on page 675 shows the basics of the transmit buffer. When the transmit shift register is empty and ready for new data, the byte from the transmit buffer is then loaded into the shift register. The byte is then kept in the shift register until it is transmitted. When a byte has been transmitted, a new byte is loaded into the shift register (if available in the transmit buffer). If the transmit buffer is empty, then the shift register also remains empty. The TXC flag in I2C_STATUS and the TXC interrupt flags in I2C_IF are then set, signaling that the transmit shift register is out of data. TXC is cleared when new data becomes available, but the TXC interrupt flag must be cleared by software.

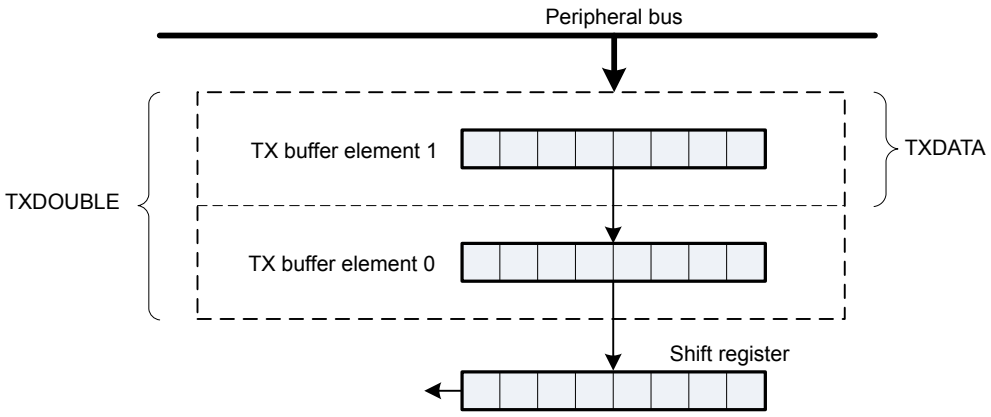


Figure 21.14. I2C Transmit Buffer Operation

The TXBL flags in I2C_STATUS and I2C_IF are used to indicate the level of the transmit buffer. The TXBIL bit in I2C_CTRL controls the level at which these flag bits are set:

- If TXBIL is cleared, the TXBL flags are set whenever the transmit buffer becomes empty (used when transmitting using I2C_TXDOUBLE).
- If TXBIL is set, the TXBL flags are set whenever the transmit buffer goes from full to half-empty or empty (used when transmitting with I2C_TXDATA).

The TXBL status flag in I2C_STATUS is cleared automatically when the condition becomes false. After the transmit FIFOs are filled, software needs to manually clear the TXBL interrupt flag. Note that the TXBL interrupt flag is 0 by default, but immediately after software sets I2C_EN.EN = 1, the TXBL interrupt flag will be set to indicate the transmit FIFO is empty. When the I²C module is disabled (I2C_EN.EN= 0), software needs to manually clear the TXBL interrupt flag (or ignore it).

Additionally, the TXBUFCNT bitfield in I2C_STATUS can be read to determine the exact number of transmit buffers filled with valid data. This is particularly useful for determining whether the transmit buffers are full. For example, if TXBUFCNT = '2', firmware can determine that both transmit buffers are filled, and that any additional data written to the transmit buffer would result in an overflow condition. Note that the TXBUFCNT count does not include the TX shift register.

If an attempt is made to write more bytes to the transmit buffer than the space available, the TXOF interrupt flag in I2C_IF is set, indicating the overflow. The data already in the buffer remains preserved, and no new data is written.

The transmit buffer and the transmit shift register can be cleared by setting command bit CLEARTX in I2C_CMD. This will prevent the I²C module from transmitting the data in the buffer and the shift register, and will make them available for new data. Any byte currently being transmitted will not be aborted. Transmission of this byte will be completed.

21.3.7.2 Receive Buffer and Shift Register

The I²C receiver uses a 2-level FIFO receive buffer and a receive shift register as shown in Figure 21.15 I2C Receive Buffer Operation on page 676. When a byte has been fully received by the receive shift register, it is loaded into the receive buffer if there is room for it, making the shift register empty to receive another byte. Otherwise, the byte waits in the shift register until space becomes available in the buffer.

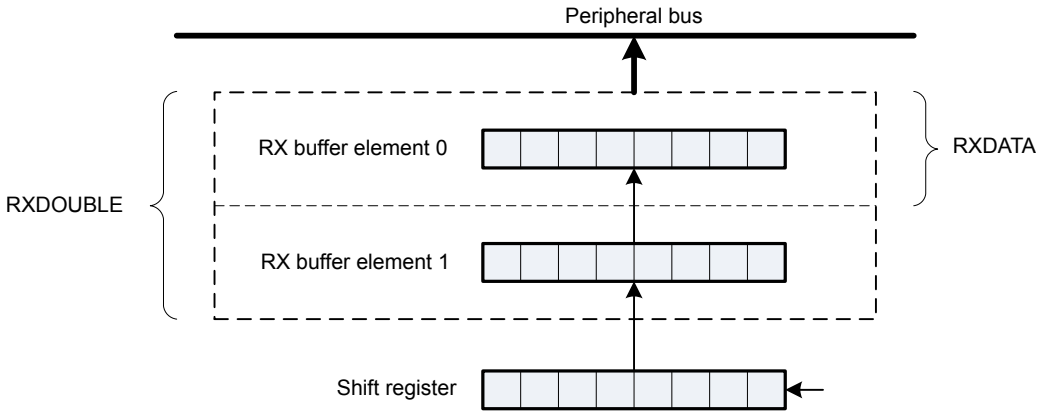


Figure 21.15. I2C Receive Buffer Operation

When a byte becomes available in the receive buffer, the RXDATAV flags in I2C_STATUS and I2C_IF are set. When the buffer becomes full, the RXFULL flags in I2C_STATUS and I2C_IF are set. The RXDATAV and RXFULL flags in I2C_STATUS are automatically cleared by hardware when their condition is no longer true. The RXDATAV and RXFULL flags in I2C_IF must be manually cleared by software after the receive FIFO is emptied. Note that when the RXFULL flag is set, indicating the buffer is full, space is still available in the receive shift register for one more byte.

The data can be fetched from the buffer in two ways. I2C_RXDATA gives access to the received byte (if two bytes are received then the one received first is fetched first). I2C_RXDOUBLE makes it possible to read the two received bytes simultaneously. If an attempt is made to read more bytes from the buffer than available, the RXUF interrupt flag in I2C_IF is set to signal the underflow, and the data read from the buffer is undefined.

When using I2C_RXDOUBLE to pick data, AUTOACK in I2C_CTRL should be set to 1. This ensures that an ACK is automatically sent out after the first byte is received so that the reception of the next byte can begin. In order to stop receiving data bytes, a NACK must be sent out through the I2C_CMD register.

I2C_RXDATAP and I2C_RXDOUBLEP can be used to read data from the receive buffer without removing it from the buffer. The RXUF interrupt flag in I2C_IF will never be set as a result of reading from I2C_RXDATAP and I2C_RXDOUBLEP, but the data read through I2C_RXDATAP when the receive buffer is empty is still undefined.

Once a transaction is complete (STOP sent or received), the receive buffer needs to be flushed (all received data must be read) before starting a new transaction.

21.3.8 Leader Operation

A bus transaction is initiated by transmitting a START condition (S) on the bus. This is done by setting the START bit in I2C_CMD. The command schedules a START condition, and makes the I^2C module generate a start condition whenever the bus becomes free.

The I^2C-bus is considered busy whenever another device on the bus transmits a START condition. Until a STOP condition is detected, the bus is owned by the leader issuing the START condition. The bus is considered free when a STOP condition is transmitted on the bus. After a STOP is detected, all leaders that have data to transmit send a START condition and begin transmitting data. Arbitration ensures that collisions are avoided.

When the START condition has been transmitted, the leader must transmit a follower address (ADDR) with an R/W bit on the bus. If this address is available in the transmit buffer, the leader transmits it immediately, but if the buffer is empty, the leader holds the I^2C-bus while waiting for software to write the address to the transmit buffer.

After the address has been transmitted, a sequence of bytes can be read from or written to the follower, depending on the value of the R/W bit (bit 0 in the address byte). If the bit was cleared, the leader has entered a leader transmitter role, where it now transmits data to the follower. If the bit was set, it has entered a leader receiver role, where it now should receive data from the follower. In either case, an unlimited number of bytes can be transferred in one direction during the transmission.

At the end of the transmission, the leader either transmits a repeated START condition (Sr) if it wishes to continue with another transfer, or transmits a STOP condition (P) if it wishes to release the bus. When operating in the leader mode, I2CCLK frequency must be higher than 2 MHz for Standard-mode, 9 MHz for Fast-mode, and 20 MHz for Fast-mode Plus.

21.3.8.1 Leader State Machine

The leader state machine is shown in Figure 21.16 I2C Leader State Machine on page 678. A leader operation starts in the far left of the state machine, and follows the solid lines through the state machine, ending the operation or continuing with a new operation when arriving at the right side of the state machine.

Branches in the path through the state machine are the results of bus events and choices made by software, either directly or indirectly. The dotted lines show where I²C-specific interrupt flags are set along the path and the full-drawn circles show places where interaction may be required by software to let the transmission proceed.

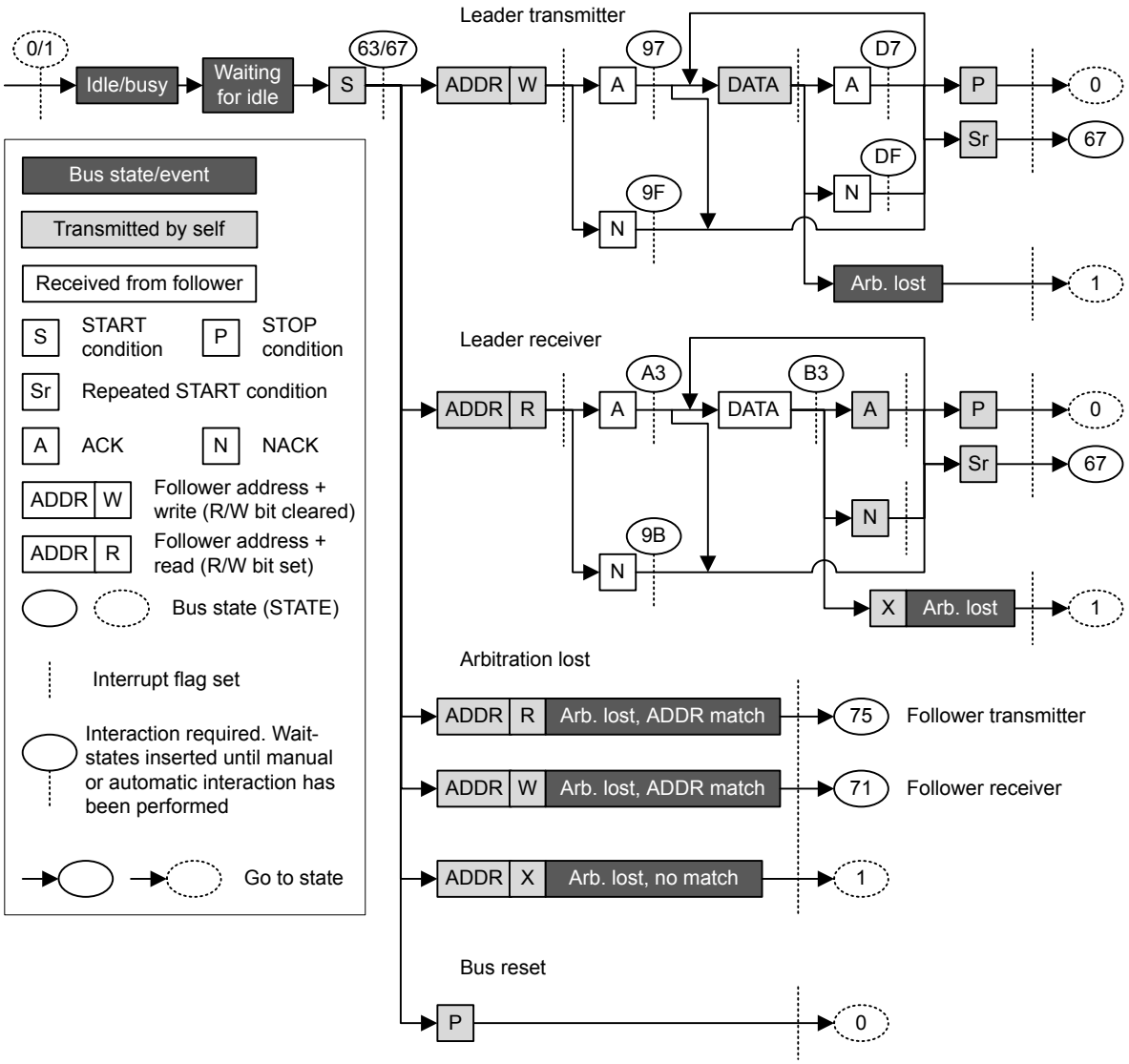


Figure 21.16. I2C Leader State Machine

21.3.8.2 Interactions

Whenever the I²C module is waiting for interaction from software, it holds the bus clock SCL low, freezing all bus activities, and the BUSHOLD interrupt flag in I2C_IF is set. The action(s) required by software depends on the current state the of the I²C module. This state can be read from the I2C_STATE register.

As an example, Table 21.3 I2C Leader Transmitter on page 681 shows the different states the I²C goes through when operating as a Leader Transmitter, i.e., a leader that transmits data to a follower. As seen in the table, when a start condition has been transmitted, a requirement is that there is an address and an R/W bit in the transmit buffer. If the transmit buffer is empty, then the BUSHOLD interrupt flag is set, and the bus is held until data becomes available in the buffer. While waiting for the address, I2C_STATE has a value 0x67, which can be used to identify exactly what the I²C module is waiting for.

Note: The bus would never stop at state 0x67 if the address was available in the transmit buffer.

The BUSHOLD interrupt flag needs to be manually cleared by software after the appropriate action has been taken.

The different interactions used by the I²C module are listed in Table 21.2 I2C Interactions in Prioritized Order on page 679 in a prioritized order. If the I²C module is in such a state that multiple courses of action are possible, then the action chosen is the one that has the highest priority. For example, after sending out a START, if an address is present in the buffer and a STOP is also pending, then the I²C will send out the STOP since it has the higher priority.

Table 21.2. I2C Interactions in Prioritized Order

Interaction	Priority	Software action	Automatically continues if
STOP*	1	Set the STOP command bit in I2C_CMD	PSTOP is set (STOP pending) in I2C_STATUS
ABORT	2	Set the ABORT command bit in I2C_CMD	Never, the transmission is aborted
CONT*	3	Set the CONT command bit in I2C_CMD	PCONT is set in I2C_STATUS (CONT pending)
NACK*	4	Set the NACK command bit in I2C_CMD	PNACK is set in I2C_STATUS (NACK pending)
ACK*	5	Set the ACK command bit in I2C_CMD	AUTOACK is set in I2C_CTRL or PACK is set in I2C_STATUS (ACK pending)
ADDR+W -> TXDATA	6	Write an address to the transmit buffer with the R/W bit set	Address is available in transmit buffer with R/W bit set
ADDR+R -> TXDATA	7	Write an address to the transmit buffer with the R/W bit cleared	Address is available in transmit buffer with R/W bit cleared
START*	8	Set the START command bit in I2C_CMD	PSTART is set in I2C_STATUS (START pending)
TXDATA/ TXDOUBLE	9	Write data to the transmit buffer	Data is available in transmit buffer
RXDATA/ RXDOUBLE	10	Read data from receive buffer	Space is available in receive buffer
None	11	No interaction is required	

The commands marked with a * in Table 21.2 I2C Interactions in Prioritized Order on page 679 can be issued before an interaction is required. When such a command is issued before it can be used/consumed by the I²C module, the command is set in a pending state, which can be read from the STATUS register. A pending START command can for instance be identified by PSTART having a high value.

Whenever the I²C module requires an interaction, it checks the pending commands. If one or a combination of these can fulfill an interaction, they are consumed by the module and the transmission continues without setting the BUSHOLD interrupt flag in I2C_IF to get an interaction from software. The pending status of a command goes low when it is consumed.

When several interactions are possible from a set of pending commands, the interaction with the highest priority, i.e., the interaction closest to the top of Table 21.2 I2C Interactions in Prioritized Order on page 679 is applied to the bus.

Pending commands can be cleared by setting the CLEARPC command bit in I2C_CMD.

21.3.8.3 Automatic ACK Interaction

When receiving addresses and data, an ACK command in I2C_CMD is normally required after each received byte. When AUTOACK is set in I2C_CTRL, an ACK is always pending, and the ACK-pending bit PACK in I2C_STATUS is thus always set, even after an ACK has been consumed. This is used when data is picked using I2C_RXDOUBLE and can also be used with I2C_RXDATA in order to reduce the amount of software interaction required during a transfer.

21.3.8.4 Reset State

After a reset, the state of the I^2C-bus is unknown. To avoid interrupting transfers on the I^2C-bus after a reset of the I^2C module or the entire MCU, the I^2C-bus is assumed to be busy when coming out of a reset, and the BUSY flag in I2C_STATUS is thus set. To be able to carry through leader operations on the I^2C-bus, the bus must be idle.

The bus goes idle when a STOP condition is detected on the bus, but on buses with little activity, the time before the I^2C module detects that the bus is idle can be significant. There are two ways of assuring that the I^2C module gets out of the busy state.

- Use the ABORT command in I2C_CMD. When the ABORT command is issued, the I^2C module is instructed that the bus is idle. The I^2C module can then initiate leader operations.
- Use the Bus Idle Timeout. When SCL has been high for a long period of time, it is very likely that the bus is idle. Set BITO in I2C_CTRL to an appropriate timeout period and set GIBITO in I2C_CTRL. If activity has not been detected on the bus within the timeout period, the bus is then automatically assumed idle, and leader operations can be initiated.

Note: If operating in follower mode, the above approach is not necessary.

21.3.8.5 Leader Transmitter

To transmit data to a follower, the leader must operate as a leader transmitter. Table 21.3 I2C Leader Transmitter on page 681 shows the states the I²C module goes through while acting as a leader transmitter. Every state where an interaction is required has the possible interactions listed, along with the result of the interactions. The table also shows which interrupt flags are set in the different states. The interrupt flags enclosed in parenthesis may be set. If the BUSHOLD interrupt in I2C_IF is set, the module is waiting for an interaction, and the bus is frozen. The value of I2C_STATE will be equal to the values given in the table when the BUSHOLD interrupt flag is set, and can be used to determine which interaction is required to make the transmission continue.

The interrupt flag START in I2C_IF is set when the I²C module transmits the START.

A leader operation is started by issuing a START command by setting START in I2C_CMD. ADDR+W, i.e., the address of the follower + the R/W bit is then required by the I²C module. If this is not available in the transmit buffer, then the bus is held and the BUSHOLD interrupt flag is set. The value of I2C_STATE will then be 0x67. As seen in the table, the I²C module also stops in this state if the address is not available after a repeated start condition.

To continue, write a byte to I2C_TXDATA with the address of the follower in the 7 most significant bits and the least significant bit cleared (ADDR+W). This address will then be transmitted, and the follower will reply with an ACK or a NACK. If no follower replies to the address, the response will also be NACK. If the address was acknowledged, the leader now has four choices. It can send data by placing it in I2C_TXDATA/ I2C_TXDOUBLE (the leader should check the TXBL interrupt flag before writing to the transmit buffer), this data is then transmitted. The leader can also stop the transmission by sending a STOP, it can send a repeated start by sending START, or it can send a STOP and then a START as soon as possible. If the leader wishes to make another transfer immediately after the current, the preferred way is to start a new transfer directly by transmitting a repeated START instead of a STOP followed by a START. This is so because if a STOP is sent out, then any leader wishing to initiate a transfer on the bus can try to gain control of it.

If a NACK was received, the leader has to issue a CONT command in addition to providing data in order to continue transmission. This is not standard I²C, but is provided for flexibility. The rest of the options are similar to when an ACK was received.

If a new byte was transmitted, an ACK or NACK is received after the transmission of the byte, and the leader has the same options as for when the address was sent.

The leader may lose arbitration at any time during transmission. In this case, the ARBLOST interrupt flag in I2C_IF is set. If the arbitration was lost during the transfer of an address, and SLAVE in I2C_CTRL is set, the leader then checks which address was transmitted. If it was the address of the leader, then the leader goes to follower mode.

After a leader has transmitted a START and won any arbitration, it owns the bus until it transmits a STOP. After a STOP, the bus is released, and arbitration decides which bus leader gains the bus next. The MSTOP interrupt flag in I2C_IF is set when a STOP condition is transmitted by the leader.

Table 21.3. I2C Leader Transmitter

I2C_STATE	Description	I2C_IF	Required interaction	Response
0x67	Start transmitted	START interrupt flag (BUSHOLD interrupt flag)	ADDR+W -> TXDATA	ADDR+W will be sent
			STOP	STOP will be sent and bus released.
			STOP + START	STOP will be sent and bus released. Then a START will be sent when bus becomes idle.
0x67	Repeated start transmitted	START interrupt flag (BUSHOLD interrupt flag)	ADDR+W -> TXDATA	ADDR+W will be sent
			STOP	STOP will be sent and bus released.
			STOP + START	STOP will be sent and bus released. Then a START will be sent when bus becomes idle.
-	ADDR+W transmitted	TXBL interrupt flag (TXC interrupt flag)	None	

I2C_STATE	Description	I2C_IF	Required interaction	Response
0x97	ADDR+W transmitted, ACK received	ACK interrupt flag (BUSHOLD interrupt flag)	TXDATA	DATA will be sent
			STOP	STOP will be sent. Bus will be released
			START	Repeated start condition will be sent
			STOP + START	STOP will be sent and the bus released. Then a START will be sent when the bus becomes idle
0x9F	ADDR+W transmitted,NACK received	NACK (BUSHOLD interrupt flag)	CONT + TXDATA	DATA will be sent
			STOP	STOP will be sent. Bus will be released
			START	Repeated start condition will be sent
			STOP + START	STOP will be sent and the bus released. Then a START will be sent when the bus becomes idle
-	Data transmitted	TXBL interrupt flag (TXC interrupt flag)	None	
0xD7	Data transmitted,ACK received	ACK interrupt flag (BUSHOLD interrupt flag)	TXDATA	DATA will be sent
			STOP	STOP will be sent. Bus will be released
			START	Repeated start condition will be sent
			STOP + START	STOP will be sent and the bus released. Then a START will be sent when the bus becomes idle
0xDF	Data transmitted,NACK received	NACK(BUSHOLD interrupt flag)	CONT + TXDATA	DATA will be sent
			STOP	STOP will be sent. Bus will be released
			START	Repeated start condition will be sent
			STOP + START	STOP will be sent and the bus released. Then a START will be sent when the bus becomes idle
-	Stop transmitted	MSTOP interrupt flag	None	
			START	START will be sent when bus becomes idle
-	Arbitration lost	ARBLOST interrupt flag	None	
			START	START will be sent when bus becomes idle

21.3.8.6 Leader Receiver

To receive data from a follower, the leader must operate as a leader receiver, see Table 21.4 I2C Leader Receiver on page 683. This is done by transmitting ADDR+R as the address byte instead of ADDR+W, which is transmitted to become a leader transmitter. The address byte loaded into the data register thus has to contain the 7-bit follower address in the 7 most significant bits of the byte, and have the least significant bit set.

When the address has been transmitted, the leader receives an ACK or a NACK. If an ACK is received, the ACK interrupt flag in I2C_IF is set, and if space is available in the receive shift register, reception of a byte from the follower begins. If the receive buffer and shift register is full however, the bus is held until data is read from the receive buffer or another interaction is made. Note that the STOP and START interactions have a higher priority than the data-available interaction, so if a STOP or START command is pending, the highest priority interaction will be performed, and data will not be received from the follower.

If a NACK was received, the CONT command in I2C_CMD has to be issued in order to continue receiving data, even if there is space available in the receive buffer and/or shift register.

After a data byte has been received the leader must ACK or NACK the received byte. If an ACK is pending or AUTOACK in I2C_CTRL is set, an ACK is sent automatically and reception continues if space is available in the receive buffer.

If a NACK is sent, the CONT command must be used in order to continue transmission. If an ACK or NACK is issued along with a START or STOP or both, then the ACK/NACK is transmitted and the reception is ended. If START in I2C_CMD is set alone, a repeated start condition is transmitted after the ACK/NACK. If STOP in I2C_CMD is set, a stop condition is sent regardless of whether START is set. If START is set in this case, it is set as pending.

As when operating as a leader transmitter, arbitration can be lost as a leader receiver. When this happens the ARBLOST interrupt flag in I2C_IF is set, and the leader has a possibility of being selected as a follower given the correct conditions.

Table 21.4. I2C Leader Receiver

I2C_STATE	Description	I2C_IF	Required interaction	Response
0x63	START transmitted	START interrupt flag (BUSHOLD interrupt flag)	ADDR+R -> TXDATA	ADDR+R will be sent
			STOP	STOP will be sent and bus released.
			STOP + START	STOP will be sent and bus released. Then a START will be sent when bus becomes idle.
0x67	Repeated START transmitted	START interrupt flag(BUSHOLD interrupt flag)	ADDR+R -> TXDATA	ADDR+R will be sent
			STOP	STOP will be sent and bus released.
			STOP + START	STOP will be sent and bus released. Then a START will be sent when bus becomes idle.
-	ADDR+R transmitted	TXBL interrupt flag (TXC interrupt flag)	None	
0xA3	ADDR+R transmitted, ACK received	ACK interrupt flag(BUSHOLD)	RXDATA	Start receiving
			STOP	STOP will be sent and the bus released
			START	Repeated START will be sent
			STOP + START	STOP will be sent and the bus released. Then a START will be sent when the bus becomes idle
0x9B	ADDR+R transmitted,NACK received	NACK(BUSHOLD)	CONT + RXDATA	Continue, start receiving
			STOP	STOP will be sent and the bus released
			START	Repeated START will be sent
			STOP + START	STOP will be sent and the bus released. Then a START will be sent when the bus becomes idle

I2C_STATE	Description	I2C_IF	Required interaction	Response
0xB3	Data received	RXDATA interrupt flag(BUSHOLD interrupt flag)	ACK + RXDATA	ACK will be transmitted, reception continues
			NACK + CONT + RXDATA	NACK will be transmitted, reception continues
			ACK/NACK + STOP	ACK/NACK will be sent and the bus will be released.
			ACK/NACK + START	ACK/NACK will be sent, and then a repeated start condition.
			ACK/NACK + STOP + START	ACK/NACK will be sent and the bus will be released. Then a START will be sent when the bus becomes idle
-	Stop received	MSTOP interrupt flag	None	
			START	START will be sent when bus becomes idle
-	Arbitration lost	ARBLOST interrupt flag	None	
			START	START will be sent when bus becomes idle

21.3.8.7 SDA/SCL Status Monitor

The I^2C module supports an SDA and SCL monitoring function. Note that this functionality is only supported when the I2C module is in single leader mode, and when the follower doesn't use clock stretching. Additionally, firmware should set the ARBDIS bit in I2C_CTRL when using the SDA/SCL monitoring to prevent the bus being released.

The SDA monitor is enabled by setting the SDAMONEN in I2C_CTRL. Once enabled, the SDA monitor will check the status of the SDA line at the following points:

- At a Start Condition, before SDA falls
- At Stop Condition, after SDA rises

After checking, the monitor will set the SDAERR flag in I2C_IF it fails to read SDA==1. To allow the SDAERR flag to generate an IRQ, set the SDAERR bit in I2C_IEN.

Similarly, the SCL monitor is enabled by setting the SCLMONEN in I2C_CTRL. Once enabled, the SCL monitor will check the status of the SCL line at the following points:

- At a Start Condition, before SCL falls
- At every clock cycle, before SCL falls
- At Stop Condition, after SCL rises

After checking, the monitor will set the SCLERR flag in I2C_IF it fails to read SCL==1. To allow the SCLERR flag to generate an IRQ, set the SCLERR bit in I2C_IEN.

21.3.9 Bus States

The I2C_STATE register can be used to determine which state the I²C module and the I²C bus are in at a given time. The register consists of the STATE bit-field, which shows which state the I²C module is at in any ongoing transmission, and a set of single-bits, which reveal the transmission mode, whether the bus is busy or idle, and whether the bus is held by this I²C module waiting for a software response.

The possible values of the STATE field are summarized in Table 21.5 I2C STATE Values on page 685. When this field is cleared, the I²C module is not a part of any ongoing transmission. The remaining status bits in the I2C_STATE register are listed in Table 21.6 I2C Transmission Status on page 685.

Table 21.5. I2C STATE Values

Mode	Value	Description
IDLE	0	No transmission is being performed by this module.
WAIT	1	Waiting for idle. Will send a start condition as soon as the bus is idle.
START	2	Start transmit phase
ADDR	3	Address transmit or receive phase
ADDRACK	4	Address ACK/NACK transmit or receive phase
DATA	5	Data transmit or receive phase
DATAACK	6	Data ACK/NACK transmit or receive phase

Table 21.6. I2C Transmission Status

Bit	Description
BUSY	Set whenever there is activity on the bus. Whether or not this module is responsible for the activity cannot be determined by this byte.
MASTER	Set when operating as a leader. Cleared at all other times.
TRANSMITTER	Set when operating as a transmitter; either a leader transmitter or a follower transmitter. Cleared at all other times
BUSHOLD	Set when the bus is held by this I²C module because an action is required by software.
NACK	Only valid when bus is held and STATE is ADDRACK or DATAACK. In that case it is set if a NACK was received. In all other cases, the bit is cleared.

Note: I2C_STATE reflects the internal state of the I²C module, and therefore only held constant as long as the bus is held, i.e., as long as BUSHOLD in I2C_STATUS is set.

21.3.10 Follower Operation

The I²C module operates in leader mode by default. To enable follower operation, i.e., to allow the device to be addressed as an I²C follower, the SLAVE bit in I2C_CTRL must be set. In this case the I²C module operates in a mixed mode, both capable of starting transmissions as a leader, and being addressed as a follower. When operating in the follower mode, I2CCLK frequency must be higher than 2 MHz for Standard-mode, 5 MHz for Fast-mode, and 14 MHz for Fast-mode Plus.

21.3.10.1 Follower State Machine

The follower state machine is shown in Figure 21.17 I2C Follower State Machine on page 686. The dotted lines show where I²C-specific interrupt flags are set. The full-drawn circles show places where interaction may be required by software to let the transmission proceed.

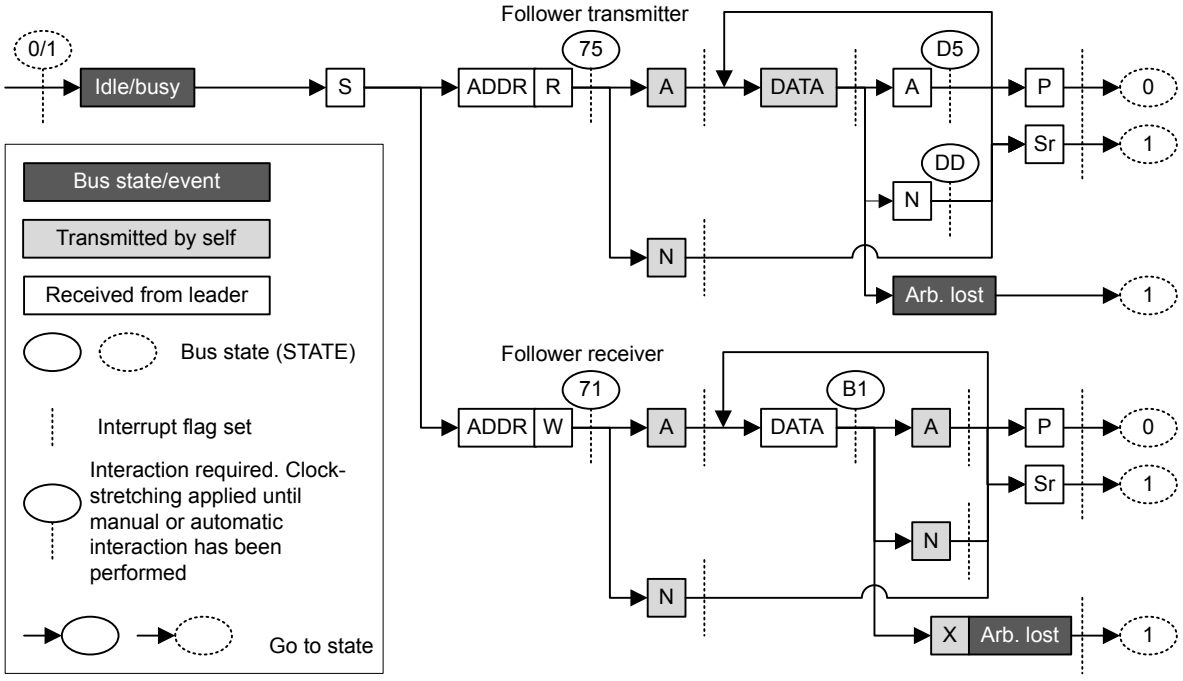


Figure 21.17. I2C Follower State Machine

21.3.10.2 Address Recognition

The I²C module provides automatic address recognition for 7-bit addresses. 10-bit address recognition is not fully automatic, but can be assisted by the 7-bit address comparator as shown in 21.3.12 Using 10-bit Addresses. Address recognition is supported in EM2/3 for I2C0 - however, the I²C0 module must be configured to use pins on either Port A or B if wakeup on address recognition from EM2/3 is desired. All other ports are available only in EM0/1. See GPIO chapter for more details.

The follower address, i.e., the address which the I²C module should be addressed with, is defined in the I2C_SADDR register. In addition to the address, a mask must be specified, telling the address comparator which bits of an incoming address to compare with the address defined in I2C_SADDR. The mask is defined in I2C_SADDRMASK, and for every zero in the mask, the corresponding bit in the follower address is treated as a don't-care, i.e., the 0-masked bits are ignored.

An incoming address that fails address recognition is automatically replied to with a NACK. Since only the bits defined by the mask are checked, a mask with a value 0x00 will result in all addresses being accepted. A mask with a value 0x7F will only match the exact address defined in I2C_SADDR, while a mask 0x70 will match all addresses where the three most significant bits in I2C_SADDR and the incoming address are equal.

If GCAMEN in I2C_CTRL is not set, the start-byte, i.e., the general call address with the R/W bit set is ignored unless it is included in the defined follower address and the address mask.

When an address is accepted by the address comparator, the decision of whether to ACK or NACK the address is passed to software.

21.3.10.3 Follower Transmitter

When SLAVE in I2C_CTRL is set, the RSTART interrupt flag in I2C_IF will be set when repeated START conditions are detected. After a START or repeated START condition, the bus leader will transmit an address along with an R/W bit. If there is no room in the receive shift register for the address, the bus will be held by the follower until room is available in the shift register. Transmission then continues and the address is loaded into the shift register. If this address does not pass address recognition, it is automatically NACK'ed by the follower, and the follower goes to an idle state. The address byte is in this case discarded, making the shift register ready for a new address. It is not loaded into the receive buffer.

If the address was accepted and the R/W bit was set (R), indicating that the leader wishes to read from the follower, the follower now goes into the follower transmitter mode. Software interaction is now required to decide whether the follower wants to acknowledge the request or not. The accepted address byte is loaded into the receive buffer like a regular data byte. If no valid interaction is pending, the bus is held until the follower responds with a command. The follower can reject the request with a single NACK command.

The follower will in that case go to an idle state, and wait for the next start condition. To continue the transmission, the follower must make sure data is loaded into the transmit buffer and send an ACK. The loaded data will then be transmitted to the leader, and an ACK or NACK will be received from the leader.

Data transmission can also continue after a NACK if a CONT command is issued along with the NACK. This is not standard I^2C however.

If the leader responds with an ACK, it may expect another byte of data, and data should be made available in the transmit buffer. If data is not available, the bus is held until data is available.

If the response is a NACK however, this is an indication of that the leader has received enough bytes and wishes to end the transmission. The follower now automatically goes idle, unless CONT in I2C_CMD is set and data is available for transmission. The latter is not standard I^2C.

The leader ends the transmission by sending a STOP or a repeated START. The SSTOP interrupt flag in I2C_IF is set when the leader transmits a STOP condition. If the transmission is ended with a repeated START, then the SSTOP interrupt flag is not set.

Note: The SSTOP interrupt flag in I2C_IF will be set regardless of whether the follower is participating in the transmission or not, as long as SLAVE in I2C_CTRL is set and a STOP condition is detected

If arbitration is lost at any time during transmission, the ARBLOST interrupt flag in I2C_IF is set, the bus is released and the follower goes idle.

See Table 21.7 I2C Follower Transmitter on page 687 for more information.

Table 21.7. I2C Follower Transmitter

I2C_STATE	Description	I2C_IF	Required interaction	Response
0x01	Repeated START received	RSTART interrupt flag (BUSHOLD interrupt flag)	RXDATA	Receive and compare address
0x75	ADDR + R received	ADDR interrupt flag	ACK + TXDATA	ACK will be sent, then DATA
		RXDATA interrupt flag	NACK	NACK will be sent, follower goes idle
		(BUSHOLD interrupt flag)	NACK + CONT + TXDATA	NACK will be sent, then DATA.
-	Data transmitted	TXBL interrupt flag (TXC interrupt flag)	None	
0xD5	Data transmitted, ACK received	ACK interrupt flag (BUSHOLD interrupt flag)	TXDATA	DATA will be transmitted
0xDD	Data transmitted, NACK received	NACK interrupt flag	None	The follower goes idle
		(BUSHOLD interrupt flag)	CONT + TXDATA	DATA will be transmitted

I2C_STATE	Description	I2C_IF	Required interaction	Response
-	Stop received	SSTOP interrupt flag	None	The follower goes idle
			START	START will be sent when bus becomes idle
-	Arbitration lost	ARBLOST interrupt flag	None	The follower goes idle
			START	START will be sent when the bus becomes idle

21.3.10.4 Follower Receiver

A follower receiver operation is started in the same way as a follower transmitter operation, with the exception that the address transmitted by the leader has the R/W bit cleared (W), indicating that the leader wishes to write to the follower. The follower then goes into follower receiver mode.

To receive data from the leader, the follower should respond to the address with an ACK and make sure space is available in the receive buffer. Transmission will then continue, and the follower will receive a byte from the leader.

If a NACK is sent without a CONT, the transmission is ended for the follower, and it goes idle. If the follower issues both the NACK and CONT commands and has space available in the receive buffer, it will be open for continuing reception from the leader.

When a byte has been received from the leader, the follower must ACK or NACK the byte. The responses here are the same as for the reception of the address byte.

The leader ends the transmission by sending a STOP or a repeated START. The SSTOP interrupt flag is set when the leader transmits a STOP condition. If the transmission is ended with a repeated START, then the SSTOP interrupt flag in I2C_IF is not set.

Note: The SSTOP interrupt flag in I2C_IF will be set regardless of whether the follower is participating in the transmission or not, as long as SLAVE in I2C_CTRL is set and a STOP condition is detected

If arbitration is lost at any time during transmission, the ARBLOST interrupt flag in I2C_IF is set, the bus is released and the follower goes idle.

See Table 21.8 I2C - Follower Receiver on page 689 for more information.

Table 21.8. I2C - Follower Receiver

I2C_STATE	Description	I2C_IF	Required interaction	Response
0x01	Repeated START received	RSTART interrupt flag (BUSHOLD interrupt flag)	RXDATA	Receive and compare address
0x71	ADDR + W received	ADDR interrupt flag RXDATA interrupt flag (BUSHOLD interrupt flag)	ACK + RXDATA	ACK will be sent and data will be received
			NACK	NACK will be sent, follower goes idle
			NACK + CONT + RXDATA	NACK will be sent and DATA will be received.
0xB1	Data received	RXDATA interrupt flag (BUSHOLD interrupt flag)	ACK + RXDATA	ACK will be sent and data will be received
			NACK	NACK will be sent and follower will go idle
			NACK + CONT + RXDATA	NACK will be sent and data will be received
-	Stop received	SSTOP interrupt flag	None	The follower goes idle
			START	START will be sent when bus becomes idle
-	Arbitration lost	ARBLOST interrupt flag	None	The follower goes idle
			START	START will be sent when the bus becomes idle

21.3.11 Transfer Automation

The I^2C can be set up to complete transfers with a minimal amount of interaction.

21.3.11.1 DMA

DMA can be used to automatically load data into the transmit buffer and load data out from the receive buffer. When using DMA, software is thus relieved of moving data to and from memory after each transferred byte.

21.3.11.2 Automatic ACK

When AUTOACK in I2C_CTRL is set, an ACK is sent automatically whenever an ACK interaction is possible and no higher priority interactions are pending.

21.3.11.3 Automatic STOP

A STOP can be generated automatically on two conditions. These apply only to the leader transmitter.

If AUTOSN in I2C_CTRL is set, the I^{2}C module ends a transmission by transmitting a STOP condition when operating as a leader transmitter and a NACK is received.

If AUTOSE in I2C_CTRL is set, the I^{2}C module always ends a transmission when there is no more data in the transmit buffer. If data has been transmitted on the bus, the transmission is ended after the (N)ACK has been received by the follower. If a START is sent when no data is available in the transmit buffer and AUTOSE is set, then the STOP condition is sent immediately following the START. Software must thus make sure data is available in the transmit buffer before the START condition has been fully transmitted if data is to be transferred.

21.3.12 Using 10-bit Addresses

When using 10-bit addresses in follower mode, set the I2C_SADDR register to 1111 0XX where XX are the two most significant bits of the 10-bit address, and set I2C_SADDRMASK to 0xFF. Address matches will now be given on all 10-bit addresses where the two most significant bits are correct.

When receiving an address match, the follower must acknowledge the address and receive the first data byte. This byte contains the second part of the 10-bit address. If it matches the address of the follower, the follower should ACK the byte to continue the transmission, and if it does not match, the follower should NACK it.

When the leader is operating as a leader transmitter, the data bytes will follow after the second address byte. When the leader is operating as a leader receiver however, a repeated START condition is sent after the second address byte. The address sent after this repeated START is equal to the first of the address bytes transmitted previously, but now with the R/W byte set, and only the follower that found a match on the entire 10-bit address in the previous message should ACK this address. The repeated start should take the leader into a leader receiver mode, and after the single address byte sent this time around, the follower begins transmission to the leader.

21.3.13 Error Handling

Note: Some registers in the I^{2}C module are considered static. This means that these need to be set before an I^{2}C transaction starts and need to stay stable during the entire transaction.

Specifically:

- The GCAMEN and SLAVE fields in the I2C_CTRL register
- The I2C_SADDR register
- The GPIO_I2Cn_ROUTEEN, GPIO_I2Cn_SCLROUTE, and GPIO_I2Cn_SDAROUTE registers

21.3.13.1 ABORT Command

Some bus errors may require software intervention to be resolved. The I^{2}C module provides an ABORT command, which can be set in I2C_CMD, to help resolve bus errors.

When the bus for some reason is locked up and the I^{2}C module is in the middle of a transmission it cannot get out of, or for some other reason the I^{2}C wants to abort a transmission, the ABORT command can be used.

Setting the ABORT command will make the I^{2}C module discard any data currently being transmitted or received, release the SDA and SCL lines and go to an idle mode. ABORT effectively makes the I^{2}C module forget about any ongoing transfers.

21.3.13.2 Bus Reset

A bus reset can be performed by setting the START and STOP commands in I2C_CMD while the transmit buffer is empty. A START condition will then be transmitted, immediately followed by a STOP condition. A bus reset can also be performed by transmitting a START command with the transmit buffer empty and AUTOSE set.

21.3.13.3 I2C-Bus Errors

An I^2C-bus error occurs when a START or STOP condition is misplaced, which happens when the value on SDA changes while SCL is high during bit-transmission on the I^2C-bus. If the I^2C module is part of the current transmission when a bus error occurs, any data currently being transmitted or received is discarded, SDA and SCL are released, the BUSERR interrupt flag in I2C_IF is set to indicate the error, and the module automatically takes a course of action as defined in Table 21.9 I2C Bus Error Response on page 691.

Table 21.9. I2C Bus Error Response

	Misplaced START	Misplaced STOP
In a leader/follower operation	Treated as START. Receive address.	Go idle. Perform any pending actions.

21.3.13.4 Bus Lockup

A lockup occurs when a leader or follower on the I^2C-bus has locked the SDA or SCL at a low value, preventing other devices from putting high values on the bus, and thus making communication on the bus impossible.

Many follower-only devices operating on an I^2C-bus are not capable of driving SCL low, but in the rare case that SCL is stuck LOW, the advice is to apply a hardware reset signal to the followers on the bus. If this does not work, cycle the power to the devices in order to make them release SCL.

When SDA is stuck low and SCL is free, a leader should send 9 clock pulses on SCL while tristating the SDA. This procedure is performed in the GPIO module after clearing the GPIO_I2Cn_ROUTEEN register and disabling the I2C module. The device that held the bus low should release it sometime within those 9 clocks. If not, use the same approach as for when SCL is stuck, resetting and possibly cycling power to the followers.

Lockup of SDA can be detected by keeping count of the number of continuous arbitration losses during address transmission. If arbitration is also lost during the transmission of a general call address, i.e., during the transmission of the STOP condition, which should never happen during normal operation, this is a good indication of SDA lockup.

Detection of SCL lockups can be done using the timeout functionality defined in 21.3.13.6 Clock Low Timeout

21.3.13.5 Bus Idle Timeout

When SCL has been high for a significant amount of time, this is a good indication of that the bus is idle. On an SMBus system, the bus is only allowed to be in this state for a maximum of 50 µs before the bus is considered idle.

The bus idle timeout BITO in I2C_CTRL can be used to detect situations where the bus goes idle in the middle of a transmission. The timeout can be configured in BITO, and when the bus has been idle for the given amount of time, the BITO interrupt flag in I2C_IF is set. The bus can also be set idle automatically on a bus idle timeout. This is enabled by setting GIBITO in I2C_CTRL.

When the bus idle timer times out, it wraps around and continues counting as long as its condition is true. If the bus is not set idle using GIBITO or the ABORT command in I2C_CMD, this will result in periodic timeouts.

Note: This timeout will be generated even if SDA is held low.

The bus idle timeout is active as long as the bus is busy, i.e., BUSY in I2C_STATUS is set. The timeout can be used to get the I^2C module out of the busy-state it enters when reset, see 21.3.8.4 Reset State.

21.3.13.6 Clock Low Timeout

The clock timeout, which can be configured in CLTO in I2C_CTRL, starts counting whenever SCL goes low, and times out if SCL does not go high within the configured timeout. A clock low timeout results in CLTOIF in I2C_IF being set, allowing software to take action.

When the timer times out, it wraps around and continues counting as long as SCL is low. An SCL lockup will thus result in periodic clock low timeouts as long as SCL is low.

21.3.13.7 Clock Low Error

The I^2C module can continue transmission in parallel with another device for the entire transaction, as long as the two communications are identical. A case may arise when (before an arbitration has been decided upon) the I^2C module decides to send out a repeated START or a STOP condition while the other device is still sending data. In the I^2C protocol specifications, such a combination results in an undefined condition. The I^2C deals with this by generating a clock low error. This means that if the I^2C is transmitting a repeated START or a STOP condition and another device (another leader or a misbehaving follower) pulls SCL low before the I^2C sends out the START/STOP condition on SDA, a clock low error is generated. The CLERR interrupt flag is then set in the I2C_IF register, any held lines are released and the I^2C device goes to idle.

21.3.14 DMA Support

The I^2C module has full DMA support. A request for the DMA controller to write to the I^2C transmit buffer can come from TXBL (transmit buffer has room for more data). The DMA controller can write to the transmit buffer using the I2C_TXDATA or the I2C_TXDOUBLE register. DMA must be configured to transfer one byte of data when writing to the I2C_TXDATA and configured for transferring two bytes of data when writing to the I2C_TXDOUBLE.

A request for the DMA controller to read from the I^2C receive buffer can come from RXDATAV (data available in the receive buffer). DMA must be configured to transfer one byte of data when reading from I2C_RXDATA and configured for transferring two bytes of data when reading from I2C_RXDOUBLE.

21.3.15 Interrupts

The interrupts generated by the I^2C module are combined into one interrupt vector, I2C_INT. If I^2C interrupts are enabled, an interrupt will be made if one or more of the interrupt flags in I2C_IF and their corresponding bits in I2C_IEN are set.

21.3.16 Wake-up

The I^2C receive section can be active all the way down to energy mode EM3 stop, and can wake up the CPU on address interrupt. All address match modes are supported.

21.4 I2C Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	I2C_IPVERSION	R	IP VERSION Register
0x004	I2C_EN	RW	Enable Register
0x008	I2C_CTRL	RW	Control Register
0x00C	I2C_CMD	W	Command Register
0x010	I2C_STATE	RH	State Register
0x014	I2C_STATUS	RH	Status Register
0x018	I2C_CLKDIV	RW	Clock Division Register
0x01C	I2C_SADDR	RW	Follower Address Register
0x020	I2C_SADDRMASK	RW	Follower Address Mask Register
0x024	I2C_RXDATA	RH	Receive Buffer Data Register
0x028	I2C_RXDOUBLE	RH	Receive Buffer Double Data Register
0x02C	I2C_RXDATAP	RH	Receive Buffer Data Peek Register
0x030	I2C_RXDOUBLEP	RH	Receive Buffer Double Data Peek Register
0x034	I2C_TXDATA	W	Transmit Buffer Data Register
0x038	I2C_TXDOUBLE	W	Transmit Buffer Double Data Register
0x03C	I2C_IF	RWH INTFLAG	Interrupt Flag Register
0x040	I2C_IEN	RW	Interrupt Enable Register
0x1000	I2C_IPVERSION_SET	R	IP VERSION Register
0x1004	I2C_EN_SET	RW	Enable Register
0x1008	I2C_CTRL_SET	RW	Control Register
0x100C	I2C_CMD_SET	W	Command Register
0x1010	I2C_STATE_SET	RH	State Register
0x1014	I2C_STATUS_SET	RH	Status Register
0x1018	I2C_CLKDIV_SET	RW	Clock Division Register
0x101C	I2C_SADDR_SET	RW	Follower Address Register
0x1020	I2C_SADDRMASK_SET	RW	Follower Address Mask Register
0x1024	I2C_RXDATA_SET	RH	Receive Buffer Data Register
0x1028	I2C_RXDOUBLE_SET	RH	Receive Buffer Double Data Register
0x102C	I2C_RXDATAP_SET	RH	Receive Buffer Data Peek Register
0x1030	I2C_RXDOUBLEP_SET	RH	Receive Buffer Double Data Peek Register
0x1034	I2C_TXDATA_SET	W	Transmit Buffer Data Register
0x1038	I2C_TXDOUBLE_SET	W	Transmit Buffer Double Data Register
0x103C	I2C_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1040	I2C_IEN_SET	RW	Interrupt Enable Register
0x2000	I2C_IPVERSION_CLR	R	IP VERSION Register

Offset	Name	Type	Description
0x2004	I2C_EN_CLR	RW	Enable Register
0x2008	I2C_CTRL_CLR	RW	Control Register
0x200C	I2C_CMD_CLR	W	Command Register
0x2010	I2C_STATE_CLR	RH	State Register
0x2014	I2C_STATUS_CLR	RH	Status Register
0x2018	I2C_CLKDIV_CLR	RW	Clock Division Register
0x201C	I2C_SADDR_CLR	RW	Follower Address Register
0x2020	I2C_SADDRMASK_CLR	RW	Follower Address Mask Register
0x2024	I2C_RXDATA_CLR	RH	Receive Buffer Data Register
0x2028	I2C_RXDOUBLE_CLR	RH	Receive Buffer Double Data Register
0x202C	I2C_RXDATAP_CLR	RH	Receive Buffer Data Peek Register
0x2030	I2C_RXDOUBLEP_CLR	RH	Receive Buffer Double Data Peek Register
0x2034	I2C_TXDATA_CLR	W	Transmit Buffer Data Register
0x2038	I2C_TXDOUBLE_CLR	W	Transmit Buffer Double Data Register
0x203C	I2C_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2040	I2C_IEN_CLR	RW	Interrupt Enable Register
0x3000	I2C_IPVERSION_TGL	R	IP VERSION Register
0x3004	I2C_EN_TGL	RW	Enable Register
0x3008	I2C_CTRL_TGL	RW	Control Register
0x300C	I2C_CMD_TGL	W	Command Register
0x3010	I2C_STATE_TGL	RH	State Register
0x3014	I2C_STATUS_TGL	RH	Status Register
0x3018	I2C_CLKDIV_TGL	RW	Clock Division Register
0x301C	I2C_SADDR_TGL	RW	Follower Address Register
0x3020	I2C_SADDRMASK_TGL	RW	Follower Address Mask Register
0x3024	I2C_RXDATA_TGL	RH	Receive Buffer Data Register
0x3028	I2C_RXDOUBLE_TGL	RH	Receive Buffer Double Data Register
0x302C	I2C_RXDATAP_TGL	RH	Receive Buffer Data Peek Register
0x3030	I2C_RXDOUBLEP_TGL	RH	Receive Buffer Double Data Peek Register
0x3034	I2C_TXDATA_TGL	W	Transmit Buffer Data Register
0x3038	I2C_TXDOUBLE_TGL	W	Transmit Buffer Double Data Register
0x303C	I2C_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3040	I2C_IEN_TGL	RW	Interrupt Enable Register

21.5 I2C Register Description

21.5.1 I2C_IPVERSION - IP VERSION Register

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x0	R	**IP version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

21.5.2 I2C_EN - Enable Register

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																EN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	EN	0x0	RW	**module enable**
	The ENABLE bit enables the module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit.			

Value	Mode	Description
0	DISABLE	Disable Peripheral Clock
1	ENABLE	Enable Peripheral Clock

21.5.3 I2C_CTRL - Control Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0	0x0		0x0			0x0		0x0				0x0		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access											RW	RW		RW			RW		RW				RW		RW	RW	RW	RW	RW	RW	RW	RW
Name											SDAMONEN	SCLMONEN		CLTO			GIBITO		BITO				CLHR		TXBIL	GCAMEN	ARBDIS	AUTOSN	AUTOSE	AUTOACK	SLAVE	CORERST

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	SDAMONEN	0x0	RW	**SDA Monitor Enable**

Set to enable SDA monitor feature. This will enable SDA rise check at loopback path. This monitor can not be enabled in Multi-Leader application

Value	Mode	Description
0	DISABLE	Disable SDA Monitor
1	ENABLE	Enable SDA Monitor

Bit	Name	Reset	Access	Description
20	SCLMONEN	0x0	RW	**SCL Monitor Enable**

Set to enable SCL monitor feature. This will enable SCL rise check at loopback path. This monitor can not be enabled in Multi-Leader application

Value	Mode	Description
0	DISABLE	Disable SCL monitor
1	ENABLE	Enable SCL monitor

Bit	Name	Reset	Access	Description
19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18:16	CLTO	0x0	RW	**Clock Low Timeout**

Use to generate a timeout when CLK has been low for the given amount of time. Wraps around and continues counting when the timeout is reached. The timeout value can be calculated by timeout = PCC/(Fscl x (Nlow + Nhigh))

Value	Mode	Description
0	OFF	Timeout disabled
1	I2C40PCC	Timeout after 40 prescaled clock cycles. In standard mode at 100 kHz, this results in a 50us timeout.
2	I2C80PCC	Timeout after 80 prescaled clock cycles. In standard mode at 100 kHz, this results in a 100us timeout.
3	I2C160PCC	Timeout after 160 prescaled clock cycles. In standard mode at 100 kHz, this results in a 200us timeout.
4	I2C320PCC	Timeout after 320 prescaled clock cycles. In standard mode at 100 kHz, this results in a 400us timeout.

Bit	Name	Reset	Access	Description
	5	I2C1024PCC		Timeout after 1024 prescaled clock cycles. In standard mode at 100 kHz, this results in a 1280us timeout.
15	GIBITO	0x0	RW	**Go Idle on Bus Idle Timeout**
	When set, the bus automatically goes idle on a bus idle timeout, allowing new transfers to be initiated.			
	Value	Mode		Description
	0	DISABLE		A bus idle timeout has no effect on the bus state.
	1	ENABLE		A bus idle timeout tells the I2C module that the bus is idle, allowing new transfers to be initiated.
14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:12	BITO	0x0	RW	**Bus Idle Timeout**
	Use to generate a timeout when SCL has been high for a given amount time between a START and STOP condition. When in a bus transaction, i.e. the BUSY flag is set, a timer is started whenever SCL goes high. When the timer reaches the value defined by BITO, it sets the BITO interrupt flag. The BITO interrupt flag will then be set periodically as long as SCL remains high. The bus idle timeout is active as long as BUSY is set. It is thus stopped automatically on a timeout if GIBITO is set. It is also stopped a STOP condition is detected and when the ABORT command is issued. The timeout is activated whenever the bus goes BUSY, i.e. a START condition is detected. The timeout value can be calculated by timeout = PCC/(Fscl x (Nlow + Nhigh))			
	Value	Mode		Description
	0	OFF		Timeout disabled
	1	I2C40PCC		Timeout after 40 prescaled clock cycles. In standard mode at 100 kHz, this results in a 50us timeout.
	2	I2C80PCC		Timeout after 80 prescaled clock cycles. In standard mode at 100 kHz, this results in a 100us timeout.
	3	I2C160PCC		Timeout after 160 prescaled clock cycles. In standard mode at 100 kHz, this results in a 200us timeout.
11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	CLHR	0x0	RW	**Clock Low High Ratio**
	Determines the values of (and ratio between) the low and high parts of the clock signal generated on SCL as leader.			
	Value	Mode		Description
	0	STANDARD		Nlow=4 and Nhigh=4, and the Nlow:Nhigh ratio is 4:4
	1	ASYMMETRIC		Nlow=6 and Nhigh=3, and the Nlow:Nhigh ratio is 6:3
	2	FAST		Nlow=11 and Nhigh=6, and the Nlow:Nhigh ratio is 11:6
7	TXBIL	0x0	RW	**TX Buffer Interrupt Level**
	Determines the interrupt and status level of the transmit buffer.			
	Value	Mode		Description
	0	EMPTY		TXBL status and the TXBL interrupt flag are set when the transmit buffer becomes empty. TXBL is cleared when the buffer becomes nonempty.

Bit	Name	Reset	Access	Description
	1	HALF_FULL		TXBL status and the TXBL interrupt flag are set when the transmit buffer goes from full to half-full or empty. TXBL is cleared when the buffer becomes full
6	GCAMEN	0x0	RW	**General Call Address Match Enable**
	Set to enable address match on general call in addition to the programmed follower address.			
	Value	Mode		Description
	0	DISABLE		General call address will be NACK'ed if it is not included by the follower address and address mask.
	1	ENABLE		When a general call address is received, a software response is required
5	ARBDIS	0x0	RW	**Arbitration Disable**
	A leader or follower will not release the bus upon losing arbitration.			
	Value	Mode		Description
	0	DISABLE		When a device loses arbitration, the ARBIF interrupt flag is set and the bus is released.
	1	ENABLE		When a device loses arbitration, the ARBIF interrupt flag is set, but communication proceeds.
4	AUTOSN	0x0	RW	**Automatic STOP on NACK**
	Write to 1 to make a leader transmitter send a STOP when a NACK is received from a follower.			
	Value	Mode		Description
	0	DISABLE		Stop is not automatically sent if a NACK is received from a follower.
	1	ENABLE		The leader automatically sends a STOP if a NACK is received from a follower.
3	AUTOSE	0x0	RW	**Automatic STOP when Empty**
	Write to 1 to make a leader transmitter send a STOP when no more data is available for transmission.			
	Value	Mode		Description
	0	DISABLE		A stop must be sent manually when no more data is to be transmitted.
	1	ENABLE		The leader automatically sends a STOP when no more data is available for transmission.
2	AUTOACK	0x0	RW	**Automatic Acknowledge**
	Set to enable automatic acknowledges.			
	Value	Mode		Description
	0	DISABLE		Software must give one ACK command for each ACK transmitted on the I2C bus.
	1	ENABLE		Addresses that are not automatically NACK'ed, and all data is automatically acknowledged.
1	SLAVE	0x0	RW	**Addressable as Follower**

Bit	Name	Reset	Access	Description
	Set this bit to allow the device to be selected as an I2C follower.			

Value	Mode	Description
0	DISABLE	All addresses will be responded to with a NACK
1	ENABLE	Addresses matching the programmed follower address or the general call address (if enabled) require a response from software. Other addresses are automatically responded to with a NACK.

Bit	Name	Reset	Access	Description
0	CORERST	0x0	RW	**Soft Reset the internal state registers**
	Set to reset the I2C_STATE register, and return the I2C module to the IDLE state. Must clear this bit to resume normal operation condition			

Value	Mode	Description
0	DISABLE	No change to internal state registers
1	ENABLE	Reset the internal state registers

21.5.4 I2C_CMD - Command Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																									W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)
Name																									CLEARPC	CLEARTX	ABORT	CONT	NACK	ACK	STOP	START

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	CLEARPC	0x0	W(nB)	**Clear Pending Commands**
	Set to clear pending commands.			
6	CLEARTX	0x0	W(nB)	**Clear TX**
	Set to clear transmit buffer and shift register. Will not abort ongoing transfer.			
5	ABORT	0x0	W(nB)	**Abort transmission**
	Abort the current transmission making the bus go idle. When used in combination with STOP, a STOP condition is sent as soon as possible before aborting the transmission. The stop condition is subject to clock synchronization.			
4	CONT	0x0	W(nB)	**Continue transmission**
	Set to continue transmission after a NACK has been received.			
3	NACK	0x0	W(nB)	**Send NACK**
	Set to transmit a NACK the next time an acknowledge is required.			
2	ACK	0x0	W(nB)	**Send ACK**
	Set to transmit an ACK the next time an acknowledge is required.			
1	STOP	0x0	W(nB)	**Send stop condition**
	Set to send stop condition as soon as possible.			
0	START	0x0	W(nB)	**Send start condition**
	Set to send start condition as soon as possible. If a transmission is ongoing and not owned, the start condition will be sent as soon as the bus is idle. If the current transmission is owned by this module, a repeated start condition will be sent. Use in combination with a STOP command to automatically send a STOP, then a START when the bus becomes idle.			

21.5.5 I2C_STATE - State Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0			0x0	0x0	0x0	0x0	0x1
Access																									R			R	R	R	R	R
Name																									STATE			BUSHOLD	NACKED	TRANSMITTER	MASTER	BUSY

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:5	STATE	0x0	R	**Transmission State**
	The state of any current transmission. Cleared if the I2C module is idle.			

Value	Mode	Description
0	IDLE	No transmission is being performed.
1	WAIT	Waiting for idle. Will send a start condition as soon as the bus is idle.
2	START	Start transmit phase
3	ADDR	Address transmit or receive phase
4	ADDRACK	Address ack/nack transmit or receive phase
5	DATA	Data transmit or receive phase
6	DATAACK	Data ack/nack transmit or receive phase

4	BUSHOLD	0x0	R	**Bus Held**
	Set if the bus is currently being held by this I2C module.			
3	NACKED	0x0	R	**Nack Received**
	Set if a NACK was received and STATE is ADDRACK or DATAACK.			
2	TRANSMITTER	0x0	R	**Transmitter**
	Set when operating as a leader transmitter or a follower transmitter. When cleared, the system may be operating as a leader receiver, a follower receiver or the current mode is not known.			
1	MASTER	0x0	R	**Leader**
	Set when operating as an I2C leader. When cleared, the system may be operating as an I2C follower.			
0	BUSY	0x1	R	**Bus Busy**
	Set when the bus is busy. Whether the I2C module is in control of the bus or not has no effect on the value of this bit. When the MCU comes out of reset, the state of the bus is not known, and thus BUSY is set. Use the ABORT command or a bus idle timeout to force the I2C module out of the BUSY state.			

21.5.6 I2C_STATUS - Status Register

Offset	Bit Position																											
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0		0x0	0x0	0x1	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																					R		R	R	R	R	R	R	R	R	R	R
Name																					TXBUFCNT		RXFULL	RXDATAV	TXBL	TXC	PABORT	PCONT	PNACK	PACK	PSTOP	PSTART

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:10	TXBUFCNT	0x0	R	**TX Buffer Count** Indicates the number of buffers filled with valid data and not transmit to tx shift register
9	RXFULL	0x0	R	**RX FIFO Full** Set when the receive buffer is full. Cleared when the receive buffer is no longer full. When this bit is set, there is still room for one more frame in the receive shift register.
8	RXDATAV	0x0	R	**RX Data Valid** Set when data is available in the receive buffer. Cleared when the receive buffer is empty.
7	TXBL	0x1	R	**TX Buffer Level** Indicates the level of the transmit buffer. if TXBIL==0, set when the transmit buffer is empty. if TXBIL==1, set when the transmit buffer is half full
6	TXC	0x0	R	**TX Complete** Set when a transmission has completed and no more data is available in the transmit buffer. Cleared when a new transmission starts.
5	PABORT	0x0	R	**Pending abort** An abort is pending and will be transmitted as soon as possible.
4	PCONT	0x0	R	**Pending continue** A continue is pending and will be transmitted as soon as possible.
3	PNACK	0x0	R	**Pending NACK** A not-acknowledge is pending and will be transmitted as soon as possible.
2	PACK	0x0	R	**Pending ACK** An acknowledge is pending and will be transmitted as soon as possible.
1	PSTOP	0x0	R	**Pending STOP** A stop condition is pending and will be transmitted as soon as possible.
0	PSTART	0x0	R	**Pending START** A start condition is pending and will be transmitted as soon as possible.

21.5.7 I2C_CLKDIV - Clock Division Register

Offset	Bit Position																															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0				
Access																												RW				
Name																												DIV				

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	DIV	0x0	RW	**Clock Divider**
	Specifies the clock divider for the I2C. Note that DIV must be 1 or higher when follower is enabled.			

21.5.8 I2C_SADDR - Follower Address Register

Offset	Bit Position																															
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0				
Access																												RW				
Name																												ADDR				

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:1	ADDR	0x0	RW	**Follower address**
	Specifies the follower address of the device.			
0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

21.5.9 I2C_SADDRMASK - Follower Address Mask Register

Offset	Bit Position																															
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									SADDRMASK							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:1	SADDRMASK	0x0	RW	**Follower Address Mask** Specifies the significant bits of the follower address. Setting the mask to 0x00 will match all addresses, while setting it to 0x7F will only match the exact address specified by ADDR.
0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

21.5.10 I2C_RXDATA - Receive Buffer Data Register

Offset	Bit Position																															
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									RXDATA							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	RXDATA	0x0	R	**RX Data** Use this register to read from the receive buffer. Buffer is emptied on read access.

21.5.11 I2C_RXDOUBLE - Receive Buffer Double Data Register

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0								0x0							
Access																	R								R							
Name																	RXDATA1								RXDATA0							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:8	RXDATA1	0x0	R	**RX Data 1**
	Second byte read from buffer. Buffer is emptied on read access.			
7:0	RXDATA0	0x0	R	**RX Data 0**
	First byte read from buffer. Buffer is emptied on read access.			

21.5.12 I2C_RXDATAP - Receive Buffer Data Peek Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									RXDATAP							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	RXDATAP	0x0	R	**RX Data Peek**
	Use this register to read from the receive buffer. Buffer is not emptied on read access.			

21.5.13 I2C_RXDOUBLEP - Receive Buffer Double Data Peek Register

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0								0x0							
Access																	R								R							
Name																	RXDATAP1								RXDATAP0							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:8	RXDATAP1	0x0	R	**RX Data 1 Peek**
	Second byte read from buffer. Buffer is not emptied on read access.			
7:0	RXDATAP0	0x0	R	**RX Data 0 Peek**
	First byte read from buffer. Buffer is not emptied on read access.			

21.5.14 I2C_TXDATA - Transmit Buffer Data Register

Offset	Bit Position																													
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0				
Access																												W(nB)				
Name																												TXDATA				

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	TXDATA	0x0	W(nB)	**TX Data**
	Use this register to write a byte to the transmit buffer.			

21.5.15 I2C_TXDOUBLE - Transmit Buffer Double Data Register

Offset	Bit Position																													
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0								0x0							
Access																	W(nB)								W(nB)							
Name																	TXDATA1								TXDATA0							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:8	TXDATA1	0x0	W(nB)	**TX Data**
	Second byte to write to buffer.			
7:0	TXDATA0	0x0	W(nB)	**TX Data**
	First byte to write to buffer.			

21.5.16 I2C_IF - Interrupt Flag Register

Offset	Bit Position																															
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset												0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access												RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name												SDAERR	SCLERR	CLERR	RXFULL	SSTOP	CLTO	BITO	RXUF	TXOF	BUSHOLD	BUSERR	ARBLOST	MSTOP	NACK	ACK	RXDATAV	TXBL	TXC	ADDR	RSTART	START

Bit	Name	Reset	Access	Description
31:21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20	SDAERR	0x0	RW	**SDA Error Interrupt Flag**
	Set when the SDA at loopback path is not equal to SDA output			
19	SCLERR	0x0	RW	**SCL Error Interrupt Flag**
	Set when the SCL at loopback path is not equal to SCL output			
18	CLERR	0x0	RW	**Clock Low Error Interrupt Flag**
	Set when the clock is pulled low before a START or a STOP condition could be transmitted.			
17	RXFULL	0x0	RW	**Receive Buffer Full Interrupt Flag**
	Set when the receive buffer becomes full.			
16	SSTOP	0x0	RW	**Follower STOP condition Interrupt Flag**
	Set when a STOP condition has been received. Will be set regardless of the follower being involved in the transaction or not.			
15	CLTO	0x0	RW	**Clock Low Timeout Interrupt Flag**
	Set on each clock low timeout. The timeout value can be set in CLTO bit field in the I2Cn_CTRL register.			
14	BITO	0x0	RW	**Bus Idle Timeout Interrupt Flag**
	Set on each bus idle timeout. The timeout value can be set in the BITO bit field in the I2Cn_CTRL register.			
13	RXUF	0x0	RW	**Receive Buffer Underflow Interrupt Flag**
	Set when data is read from the receive buffer through the I2Cn_RXDATA register while the receive buffer is empty. It is also set when data is read through the I2Cn_RXDOUBLE while the buffer is not full.			
12	TXOF	0x0	RW	**Transmit Buffer Overflow Interrupt Flag**
	Set when data is written to the transmit buffer while the transmit buffer is full.			
11	BUSHOLD	0x0	RW	**Bus Held Interrupt Flag**
	Set when the bus becomes held by the I2C module.			
10	BUSERR	0x0	RW	**Bus Error Interrupt Flag**
	Set when a bus error is detected. The bus error is resolved automatically, but the current transfer is aborted.			
9	ARBLOST	0x0	RW	**Arbitration Lost Interrupt Flag**
	Set when arbitration is lost.			
8	MSTOP	0x0	RW	**Leader STOP Condition Interrupt Flag**

Bit	Name	Reset	Access	Description
	Set when a STOP condition has been successfully transmitted. If arbitration is lost during the transmission of the STOP condition, then the MSTOP interrupt flag is not set.			
7	NACK	0x0	RW	**Not Acknowledge Received Interrupt Flag**
	Set when a NACK has been received.			
6	ACK	0x0	RW	**Acknowledge Received Interrupt Flag**
	Set when an ACK has been received.			
5	RXDATAV	0x0	RW	**Receive Data Valid Interrupt Flag**
	Set when received data is half full			
4	TXBL	0x0	RW	**Transmit Buffer Level Interrupt Flag**
	if TXBIL==0, set when the transmit buffer is empty. if TXBIL==1, set when the transmit is half full			
3	TXC	0x0	RW	**Transfer Completed Interrupt Flag**
	Set when the transmit shift register becomes empty and there is no more data in the transmit buffer.			
2	ADDR	0x0	RW	**Address Interrupt Flag**
	Set when incoming address is accepted, i.e. own address or general call address is received.			
1	RSTART	0x0	RW	**Repeated START condition Interrupt Flag**
	Set when a repeated start condition is detected.			
0	START	0x0	RW	**START condition Interrupt Flag**
	Set when a start condition is successfully transmitted.			

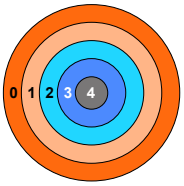
21.5.17 I2C_IEN - Interrupt Enable Register

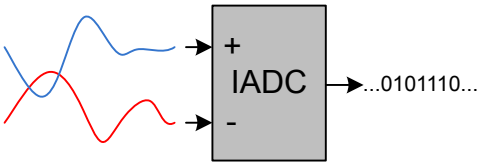
Offset	Bit Position																															
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset												0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access												RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name												SDAERR	SCLERR	CLERR	RXFULL	SSTOP	CLTO	BITO	RXUF	TXOF	BUSHOLD	BUSERR	ARBLOST	MSTOP	NACK	ACK	RXDATAV	TXBL	TXC	ADDR	RSTART	START

Bit	Name	Reset	Access	Description
31:21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20	SDAERR	0x0	RW	**SDA Error Interrupt Flag**
	Set when SDA at loopback path is not equal to SDA output			
19	SCLERR	0x0	RW	**SCL Error Interrupt Flag**
	Set when SCL at loopback path is not equal to SCL output			
18	CLERR	0x0	RW	**Clock Low Error Interrupt Flag**
	Set when the clock is pulled low before a START or a STOP condition could be transmitted.			
17	RXFULL	0x0	RW	**Receive Buffer Full Interrupt Flag**
	Set when the receive buffer becomes full.			
16	SSTOP	0x0	RW	**Follower STOP condition Interrupt Flag**
	Set when a STOP condition has been received. Will be set regardless of the follower being involved in the transaction or not.			
15	CLTO	0x0	RW	**Clock Low Timeout Interrupt Flag**
	Set on each clock low timeout. The timeout value can be set in CLTO bit field in the I2Cn_CTRL register.			
14	BITO	0x0	RW	**Bus Idle Timeout Interrupt Flag**
	Set on each bus idle timeout. The timeout value can be set in the BITO bit field in the I2Cn_CTRL register.			
13	RXUF	0x0	RW	**Receive Buffer Underflow Interrupt Flag**
	Set when data is read from the receive buffer through the I2Cn_RXDATA register while the receive buffer is empty. It is also set when data is read through the I2Cn_RXDOUBLE while the buffer is not full.			
12	TXOF	0x0	RW	**Transmit Buffer Overflow Interrupt Flag**
	Set when data is written to the transmit buffer while the transmit buffer is full.			
11	BUSHOLD	0x0	RW	**Bus Held Interrupt Flag**
	Set when the bus becomes held by the I2C module.			
10	BUSERR	0x0	RW	**Bus Error Interrupt Flag**
	Set when a bus error is detected. The bus error is resolved automatically, but the current transfer is aborted.			
9	ARBLOST	0x0	RW	**Arbitration Lost Interrupt Flag**
	Set when arbitration is lost.			
8	MSTOP	0x0	RW	**Leader STOP Condition Interrupt Flag**

Bit	Name	Reset	Access	Description
	Set when a STOP condition has been successfully transmitted. If arbitration is lost during the transmission of the STOP condition, then the MSTOP interrupt flag is not set.			
7	NACK	0x0	RW	**Not Acknowledge Received Interrupt Flag**
	Set when a NACK has been received.			
6	ACK	0x0	RW	**Acknowledge Received Interrupt Flag**
	Set when an ACK has been received.			
5	RXDATAV	0x0	RW	**Receive Data Valid Interrupt Flag**
	Set when data is available in the receive buffer. Cleared automatically when the receive buffer is read.			
4	TXBL	0x0	RW	**Transmit Buffer Level Interrupt Flag**
	Set when the transmit buffer becomes empty. Cleared automatically when new data is written to the transmit buffer.			
3	TXC	0x0	RW	**Transfer Completed Interrupt Flag**
	Set when the transmit shift register becomes empty and there is no more data in the transmit buffer.			
2	ADDR	0x0	RW	**Address Interrupt Flag**
	Set when incoming address is accepted, i.e. own address or general call address is received.			
1	RSTART	0x0	RW	**Repeated START condition Interrupt Flag**
	Set when a repeated start condition is detected.			
0	START	0x0	RW	**START condition Interrupt Flag**
	Set when a start condition is successfully transmitted.			

22. IADC - Incremental Analog to Digital Converter





Quick Facts

What?

The IADC is used to convert analog voltages into a digital representation and features high-speed, low-power operation.

Why?

In many applications there is a need to measure analog signals and record them in a digital representation, without exhausting the energy source.

How?

The low power IADC samples one or more input channels in a programmable sequence. With the help of PRS and DMA, the IADC can operate without CPU intervention in EM2 and EM3, minimizing the number of powered up resources. The IADC can be automatically shut down between conversions to further reduce the energy consumption.

22.1 Introduction

The IADC uses an Incremental Successive Approximation Architecture, with a resolution of up to 12 bits when operating at two million samples per second (2 Msps). The flexible incremental architecture uses oversampling to allow applications to trade speed for higher resolution. A high-accuracy mode enables greater than 15 bits of noise-free resolution. An integrated input multiplexer can select from external I/Os and several internal signals.

22.2 Features

- Flexible oversampled architecture allows for tradeoffs between speed and resolution.
  - Normal Mode
    - 1 Msps with oversampling ratio = 2
    - 76.9 ksps with oversampling ratio = 32
  - High Speed Mode
    - 2 Msps with oversampling ratio = 2
    - 153.8 ksps with oversampling ratio = 32
  - High Accuracy Mode
    - 10.7 ksps with oversampling ratio = 92
    - 3.8 ksps with oversampling ratio = 256
- Digital post-averaging
- Internal and external conversion trigger sources
  - Immediate (software triggered)
  - Local IADC timer
  - External TIMER module (synchronous with output / PWM generation)
  - General PRS hardware signal
- Integrated prescaler for conversion clock generation
- Can be run during EM2 and EM3, waking up the system on interrupts as needed
- Selectable reference sources
  - 1.21 V internal reference
  - External precision reference
  - Analog supply
- Support for offset and gain calibration
- Programmable input gain: 0.5x, 1x, 2x, 3x, or 4x
- Flexible output formatting
  - Unipolar or 2's complement bipolar data
  - Results can be saved in 12 bit, 16 bit, or 20 bit format
  - Programmable left or right justification
  - Optional channel ID tag
- Digital window comparison function detects when results are inside/outside a programmable window
- Two independent groups of configuration registers for setting IADC mode, clock prescaler, reference selection, oversample rate, unipolar/bipolar output formatting, and analog gain
- Programmable single channel conversion
  - Can use either configuration group
  - Triggered by any conversion trigger source
  - Can be tailgated after a scan sequence
  - One shot or continuous mode
  - Local FIFO for immediate data storage
  - Programmable watermark level to generate interrupt or initiate DMA transfer
  - Supports overflow and underflow interrupt generation
  - Supports window compare function
- Autonomous multi-channel scan
  - Up to 16 configurable slots in scan sequence
  - Each slot allows independent selection of configuration group, channel selection, and window compare enable
  - Triggered by any conversion trigger source
  - One shot or continuous mode
  - Local FIFO for immediate data storage
  - Programmable watermark level to generate interrupt or initiate DMA transfer
  - Supports overflow and underflow interrupt generation
  - Conversion tailgating support for predictable periodic scans

- Available interrupt sources:
  - Single FIFO has DVL (data valid level) entries available (also generates DMA request)
  - Scan FIFO has DVL (data valid level) entries available (also generates DMA request)
  - Single FIFO result compared true for digital compare window
  - Scan FIFO result compared true for digital compare window
  - Single queue conversion has completed
  - Scan queue entry conversion has completed
  - Scan queue table conversion has completed
  - Single FIFO overflow or underflow
  - Scan FIFO overflow or underflow
  - Polarity Error interrupt
  - Port Allocation Error interrupt
  - EM23 clock configuration error

22.3 Functional Description

The incremental ADC module block diagram is shown in Figure 22.1 IADC Overview on page 714.

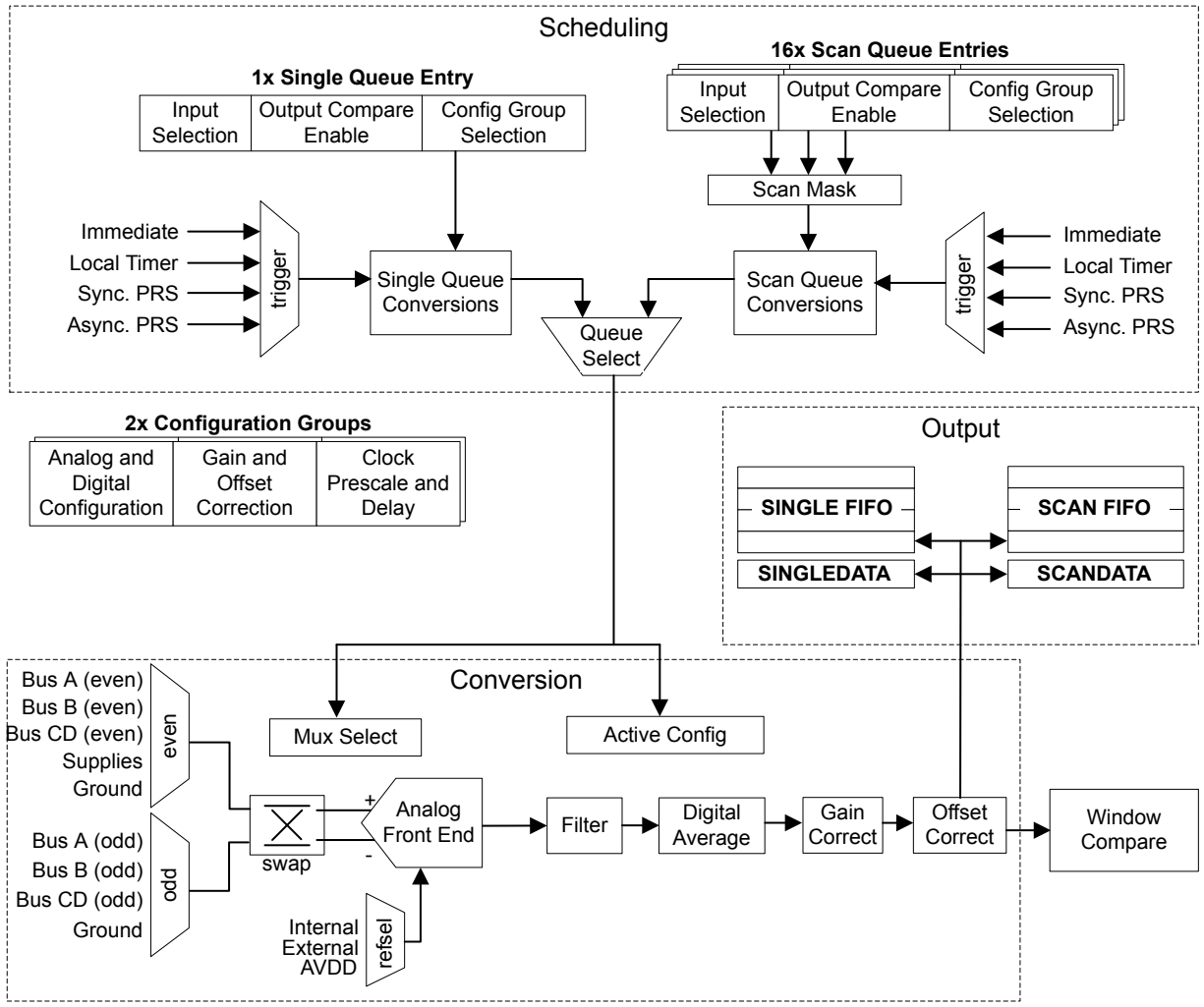


Figure 22.1. IADC Overview

22.3.1 Register Access

Many of the IADC module's configuration registers can only be written while the module is disabled (IADC_EN_EN = 0). These are IADC_CTRL, IADC_TIMER, IADC_CMPTHR, IADC_TRIGGER, IADC_CFGx, IADC_SCALEx, IADC_SCHEDx, and IADC_SCANx. A typical setup sequence for the IADC module is:

1. With the IADC disabled (IADC_EN_EN = 0), program all configuration registers listed above.
2. Enable the IADC by setting EN in IADC_EN to 1.
3. Program the remaining configuration registers.
4. Enable the single or scan queue.
5. The IADC is ready for use.

22.3.2 Clocking

The IADC logic is partitioned into two clock domains: CLK_BUS (APBIF) and CLK_SRC_ADC (CORE). The APBIF domain contains the IADC registers and FIFO read logic. The rest of the IADC is clocked mainly by CLK_SRC_ADC and ADC_CLK, both of which are derived from CLK_CMU_ADC, as shown in .

CLK_CMU_ADC is the incoming clock routed to the ADC by the CMU, and may be up to 80 MHz. It is selected within the CMU module. If the ADC is to be used synchronously with an external TIMER module, the clock should be configured to derive from the group A clock. If configuring for operation in EM2 or EM3, a clock source available in EM2 and EM3 must be used directly, as the group A clock multiplexer will be shut down in EM2 and EM3.

CLK_SRC_ADC is derived from CLK_CMU_ADC, and must be no faster than 40 MHz. The HSCLKRATE field in IADC_CTRL sets the prescaler to divide CLK_CMU_ADC. If CLK_CMU_ADC is already 40 MHz or slower, HSCLKRATE can be set to 0x0 to pass the clock through to CLK_SRC_ADC without dividing it. CLK_SRC_ADC is the clock source used for the TIMEBASE prescaler as well as the local IADC timer.

ADC_CLK is used to drive the ADC front-end and state machine logic. Another prescaler is used to reduce CLK_SRC_ADC to a suitable frequency for the ADC operating mode. Different operational modes have different restrictions on the IADC clock speed. Because the operational mode may be different for single vs. scan conversions, or even for different conversions within a scan, each configuration group has a PRESCALE bit field in the IADC_SCHEDx register.

When IADC_CFGx.ADCMODE is set to NORMAL, PRESCALE must be set to limit ADC_CLK to no faster than 10 MHz for 0.5x and 1x analog gain settings. For analog gain of 2x, 3x, and 4x, the maximum ADC_CLK is 5 MHz, 2.5 MHz, or 2.5 MHz respectively.

When IADC_CFGx.ADCMODE is set to HIGHSPEED, everything from NORMAL mode is scaled by a factor of 2. PRESCALE must be set to limit ADC_CLK to no faster than 20 MHz for 0.5x and 1x analog gain settings. For analog gain of 2x, 3x, and 4x, the maximum ADC_CLK is 10 MHz, 5 MHz, or 5 MHz respectively. It is recommended to run ADC_CLK no slower than 100 kHz.

When IADC_CFGx.ADCMODE is set to HIGHACCURACY, PRESCALE must be set to limit ADC_CLK to no faster than 5 MHz, regardless of the analog gain setting.

These restrictions are summarized in Table 22.1 Maximum ADC_CLK Speed vs Analog Gain and ADCMODE Settings on page 716.

Table 22.1. Maximum ADC_CLK Speed vs Analog Gain and ADCMODE Settings

Analog Gain	CFGx.ADCMODE = NORMAL	CFGx.ADCMODE = HIGHSPEED	CFGx.ADCMODE = HIGHACCURACY
0.5 x	10 MHz	20 MHz	5 MHz
1 x	10 MHz	20 MHz	5 MHz
2 x	5 MHz	10 MHz	5 MHz
3 x	2.5 MHz	5 MHz	5 MHz
4 x	2.5 MHz	5 MHz	5 MHz

Note: If HSCLKRATE is configured to divide CLK_CMU_ADC by more than 1 (HSCLKRATE != 0), then PRESCALE must not be set to divide by 1 (PRESCALE = 0). When this condition is detected, a PRESCALE value of 1 (divide by 2) will be automatically be used instead of the programmed PRESCALE value.

The suspend mode fields IADC_CTRL_ADCCLKSUSPEND0 (for scan conversions) or IADC_CTRL_ADCCLKSUSPEND1 (for single conversions) can be used to shut down the clock between conversions and save power. The ADC logic will wake up the clock before starting IADC warmup and performing a conversion. If the suspend mode is set, the clock will shut down again once the conversion is complete.

When IADC_TRIGGER_SCANTRIGSEL or IADC_TRIGGER_SINGLETRIGSEL is set to IMMEDIATE, IADC_CTRL_ADCCLKSUSPENDn will force the clock to only be running when one of the queues is enabled.

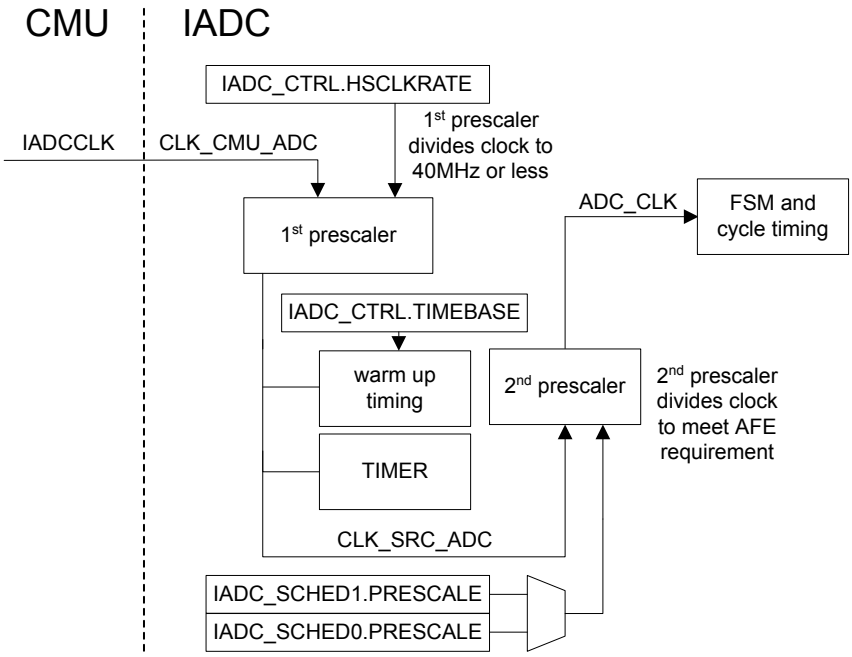


Figure 22.2. Clocking

22.3.3 Conversion Timing

The IADC takes multiple samples of the analog signal to produce each output. The number of input samples contributing to an output word is determined by the oversampling ratio (OSR). Higher OSR settings will improve the ADC's INL and DNL, and reduce system-level noise, but require more time for each conversion. Different OSRs may be specified for each configuration group. It is important to note that oversampling is an analog process which provides more input samples to the digital filter. For Normal and High-Speed modes, the OSR is configured with the OSRHS bit field in the IADC_CFGx register. The OSR options for High-Accuracy mode are different, and are configured with the OSRHA bit field in the IADC_CFGx register.

During a conversion, the effective front-end sampling frequency (F_{sample}) in Normal and High-Speed modes is equal to ADC_CLK / 4. In High-Accuracy mode, F_{sample} is ADC_CLK / 5.

22.3.3.1 Warmup Time

To save energy, the IADC can be configured to power down completely or enter a standby state between conversions, if full speed operation is not required for the application. The required ADC warm up time from a full powered-down state is 5 us. Warmup from a standby state requires 1 us. Warmup is automatically timed by the ADC logic when it is required, but software must configure the TIMEBASE field in IADC_CTRL for a minimum 1 us interval. Note that the TIMEBASE counter receives CLK_SRC_ADC, and should be programmed based on that frequency. For example, if CLK_SRC_ADC is 40 MHz, TIMEBASE should be set to at least 0x27 (39) to produce the minimum 1 us interval. When transitioning from a powered-down state, the IADC will use five TIMEBASE intervals. When in standby the IADC will use one TIMEBASE interval.

The WARMUPMODE field in the IADC_CTRL register defines whether the IADC is powered down between conversions (WARMUPMODE = NORMAL), in standby between conversions (WARMUPMODE = KEEPINSTANDBY), or remains powered up (WARMUPMODE = KEEPWARM). The resulting start-up time is shown in Figure 22.3 Start-up Timing on page 718. Note that even in WARMUPMODE = KEEPWARM or KEEPINSTANDBY, the ADC will implement 5 TIMEBASE intervals of warmup on initial power up, or any configuration change affecting PRESCALE, ADCMODE, or REFSEL. IADC_STATUS_ADCWARM reflects the current warmup status of the IADC.

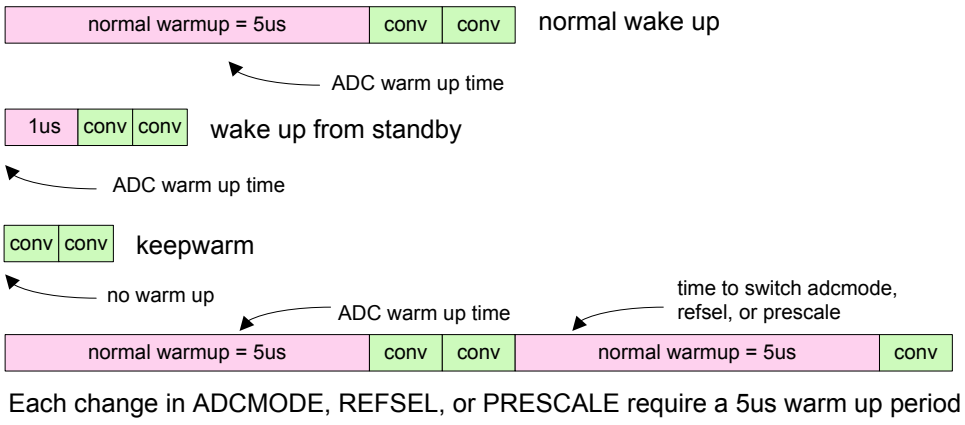


Figure 22.3. Start-up Timing

22.3.3.2 Conversion Pipeline in Normal and High-Speed Modes

The IADC uses a pipelined architecture to perform different stages of the ADC conversion in parallel.

In Normal and High-Speed modes, the conversion time for a single sample can be determined from the OSR and the pre-scaled ADC_CLK frequency (f_{ADC_CLK}) as:

Conversion Time = ((4 * OSR) + 2) / f_{ADC_CLK}

The minimum OSR is 2, meaning that the fastest possible conversion lasts 10 ADC_CLK clock cycles.

In Normal or High Speed mode, the IADC will automatically insert 2 additional cycles in the pipeline when changing channels to a new input. This allows for hold timing on the previous conversion and allows for time to tristate the ABUS analog buses before connecting the next input to the analog bus. Therefore the maximum sampling rate while continuously sampling on one channel in Normal mode (with ADC_CLK = 10 MHz) is 1 Msps, and the maximum sampling rate while switching channels is 833 ksps.

In High-Speed mode the allowed ADC_CLK speed is doubled to 20 MHz. The maximum sampling rate while continunously converting a single channel is 2 Msps, and the maximum sampling rate while switching channels is 1.67 Msps.

Figure 22.4 Normal Mode ADC Pipeline on page 719 and Figure 22.5 High-Speed Mode ADC Pipeline on page 720 show both single-channel and channel-switching scenarios powering up from a shutdown state with WARMUPMODE = NORMAL. The 5 us warm-up is shown in pink, a first conversion pipeline in green, and a second conversion in orange. The blue area in the top diagram represents the extra time to tristate while changing channels.

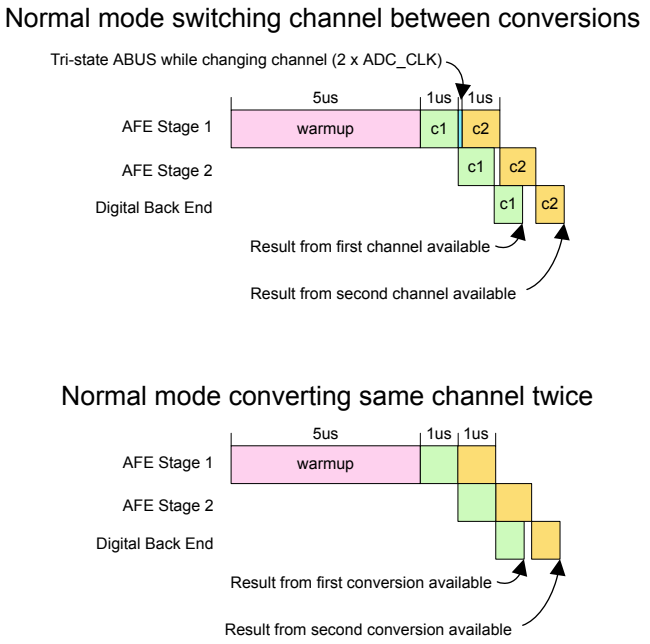


Figure 22.4. Normal Mode ADC Pipeline

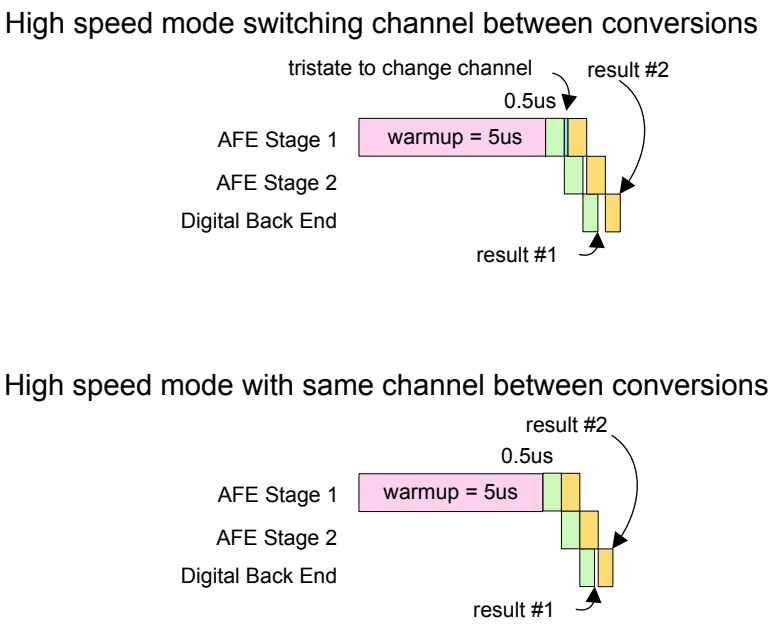


Figure 22.5. High-Speed Mode ADC Pipeline

22.3.3.3 Conversion Pipeline in High-Accuracy Mode

In High-Accuracy mode, the conversion time for a single sample can be determined from the OSR and the pre-scaled ADC_CLK frequency (f_{ADC_CLK}) as:

Conversion Time = ((5 * OSR) + 7) / f_{ADC_CLK}

The minimum OSR is 16, meaning that the fastest possible conversion in High-Accuracy mode lasts 87 ADC_CLK clock cycles.

In High-Accuracy mode, there are no additional cycles required to change channels to a new input. Figure 22.6 High-Accuracy Mode ADC Pipeline on page 720 shows the pipeline timing when powering up from a shutdown state with WARMUPMODE = NORMAL. The 5 us warmup is shown in pink, a first conversion pipeline in green, and a second conversion in orange.

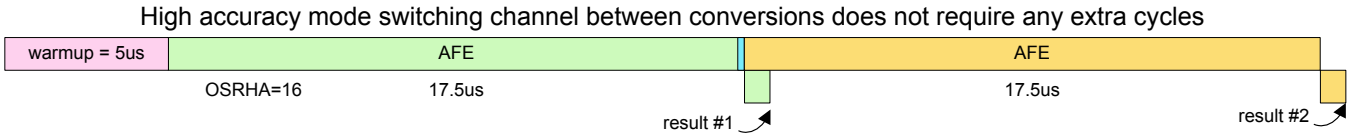


Figure 22.6. High-Accuracy Mode ADC Pipeline

22.3.3.4 Scheduling and Triggers

The IADC has several triggering options available for both the Single queue and the Scan queue. When a conversion trigger occurs and there are no other conversions active or pending, the request is serviced immediately. If both the single and scan queues are being used in an application, it is possible to serve the conversion requests as needed, and specify their priority.

Conversion triggering is configured using bit fields in the IADC_TRIGGER register. The SINGLETRIGSEL and SCANTRIGSEL fields specify the trigger source for Single and Scan conversion queues, respectively. The options for trigger source are:

- IMMEDIATE - Trigger from software. This is useful for triggering conversions on-demand from software with no specific sampling frequency requirements, or initiating continuous conversions at full speed.
- TIMER - Use the IADC local timer to trigger conversions. This is useful for triggering conversions at precise intervals.
- PRSCLKGRP - Use a synchronous PRS channel to trigger from an external peripheral in the same clock group domain (i.e. clock group A). This is useful for synchronizing conversions precisely with external TIMER events or PWM outputs.

  Note: It is recommended to configure the PRS consumer registers prior to enabling synchronous PRS triggers to avoid false triggers.

- PRSPOS - Use a positive edge of an asynchronous PRS channel to trigger conversions. The trigger source will require 1-2 CLK_SRC_ADC cycles to synchronize. This is useful for triggering conversions as needed from asynchronous peripheral sources such as GPIO inputs, SYSRTC events, etc.
- PRSNEG - Use a negative edge of an asynchronous PRS channel to trigger conversions. This is the same as PRSPOS, but operates on negative edges of the selected input.
- LESENSE (SCAN Only) - Use signalling from the LESENSE peripheral to trigger conversions. When using this mode, only one entry in the SCAN table (specified by the LESENSE channel) is converted per conversion request, and the SCAN queue is unavailable for normal operation.

Both the single and scan trigger sources can be configured to generate one request per trigger, or begin continuous conversions. Setting SINGLETRIGACTION to ONCE will make one conversion request each time the selected single trigger occurs, and a single ADC output will be converted. Setting SINGLETRIGACTION to CONTINUOUS allows the single trigger to begin the first conversion, and when a conversion completes a new one will be requested immediately without requiring a new trigger. Channel selections and configuration should not be changed while SINGLETRIGACTION is set to CONTINUOUS. Doing so can produce conversion errors. The scan queue should be used if channel or configuration switching is required.

The SCANTRIGACTION field works to request conversion scans in a similar manner. Setting SCANTRIGACTION to ONCE will make one request each time the selected scan trigger occurs, and the IADC will perform all conversions specified in the scan once before stopping. Setting SCANTRIGACTION to CONTINUOUS allows the scan trigger to initiate continuous scans. When a scan cycle completes, a new one will be requested immediately without requiring a new trigger.

Conversion priority can be adjusted using the SINGLETAILGATE bit. By default, SINGLETAILGATE is set to TAILGATEOFF, meaning that conversion triggers are queued in the order they are received. Any conversion trigger for the Single queue or the Scan queue will initiate a conversion as soon as possible. If any conversion is already in progress or pending, the new conversion will be handled after the current operation.

Setting SINGLETAILGATE to TAILGATEON gives ultimate priority to the Scan queue. The IADC will only perform single conversions immediately after completion of a scan. This allows systems to use the scan queue for high-priority conversions with tight timing requirements, and the single queue for low-priority, on-demand conversion events. Note that this setting should only be used when scan conversions are guaranteed to trigger. If no scan sequence is triggered, any single conversion trigger will remain pending indefinitely. It is also important to note that if there is not enough time between scan conversions to service a single conversion, the next scan conversion will be delayed.

22.3.3.4.1 Conversion Triggering Examples

Scheduling a Single Sample

The simplest use case for the IADC is performing one conversion on-demand from the Single queue. Figure 22.7 Immediate Single Conversion on page 722 shows the configuration and timing of this use case. The IADC warmup mode is configured for normal (shuts down between conversions). The single queue trigger is configured for immediate triggering of one conversion, and tailgating is turned off. When the conversion is requested (by setting IADC_CMD_SINGLESTART), the IADC block warms up and then begins converting. During the conversion, the CONVERTING bit in IADC_STATUS is set. When the conversion is complete, the queue is disabled, and SINGLEQEN returns low.

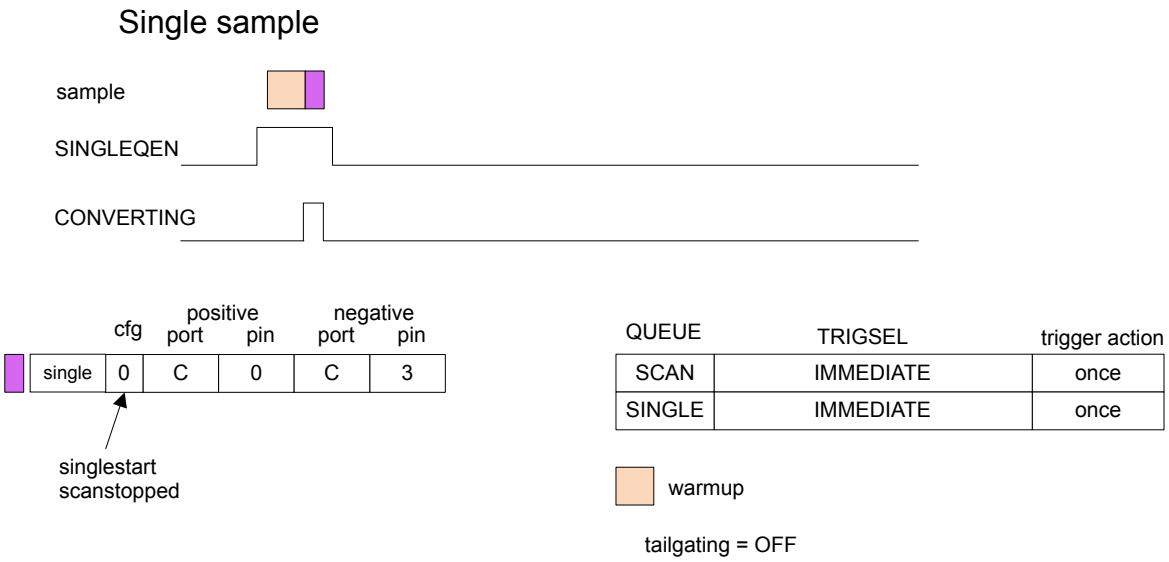


Figure 22.7. Immediate Single Conversion

Periodic Scans

Another common use case is to periodically trigger the IADC to perform a multi-channel scan. Figure 22.8 Periodic Scan Example on page 723 shows the timing of a periodic scan triggered by the IADC's local timer. The scanner is configured to sample four different channels; two using configuration 0 and two using configuration 1. Note that a single TIMER trigger is used to initiate each scan, and all four samples are taken for each trigger. Note also that the IADC inserts another warmup time between conversions 1 and 2, when it switches from configuration 0 to configuration 1. The single queue is disabled and not used in this example.

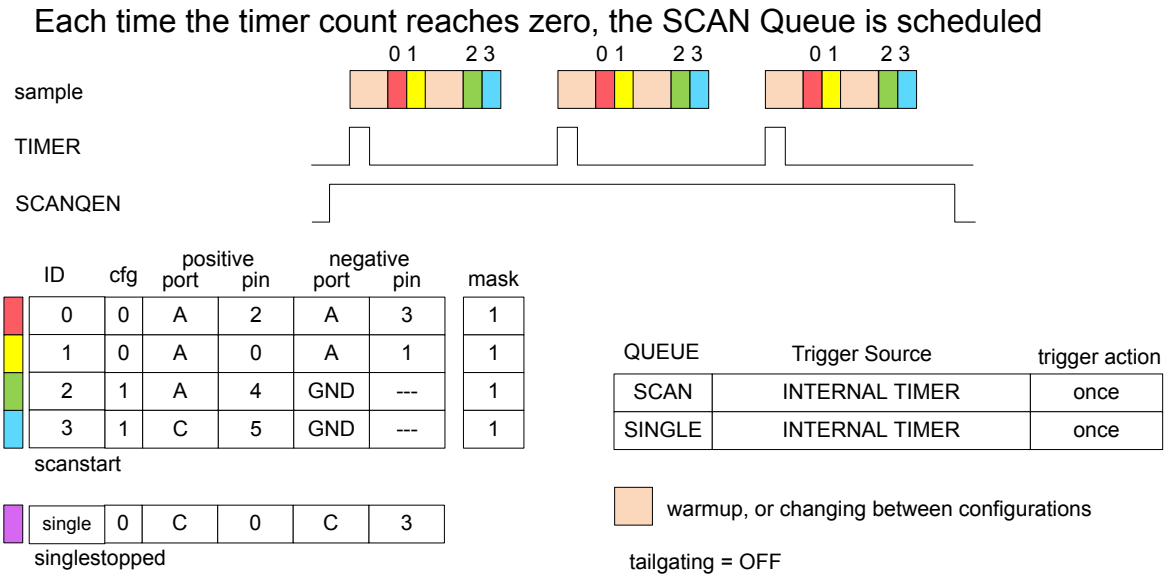


Figure 22.8. Periodic Scan Example

Tailgating Examples

An example using conversion tailgating is shown in Figure 22.9 Simple Conversion with Tailgating Enabled on page 724. In the example, the Scan queue is configured to trigger a two-channel conversion periodically on the IADC local timer, while the Single queue is configured to trigger on-demand from software. When a single conversion is requested, it waits until after the scan sequence is complete, and then the single conversion is performed. The scan conversions are using configuration 0, and the single conversion is using configuration 1, so a warmup delay is inserted between the end of the scan and the beginning of the single conversion cycle. Note that this example provides plenty of time between IADC scan conversions for the single conversion to occur, and no scan conversions are delayed.

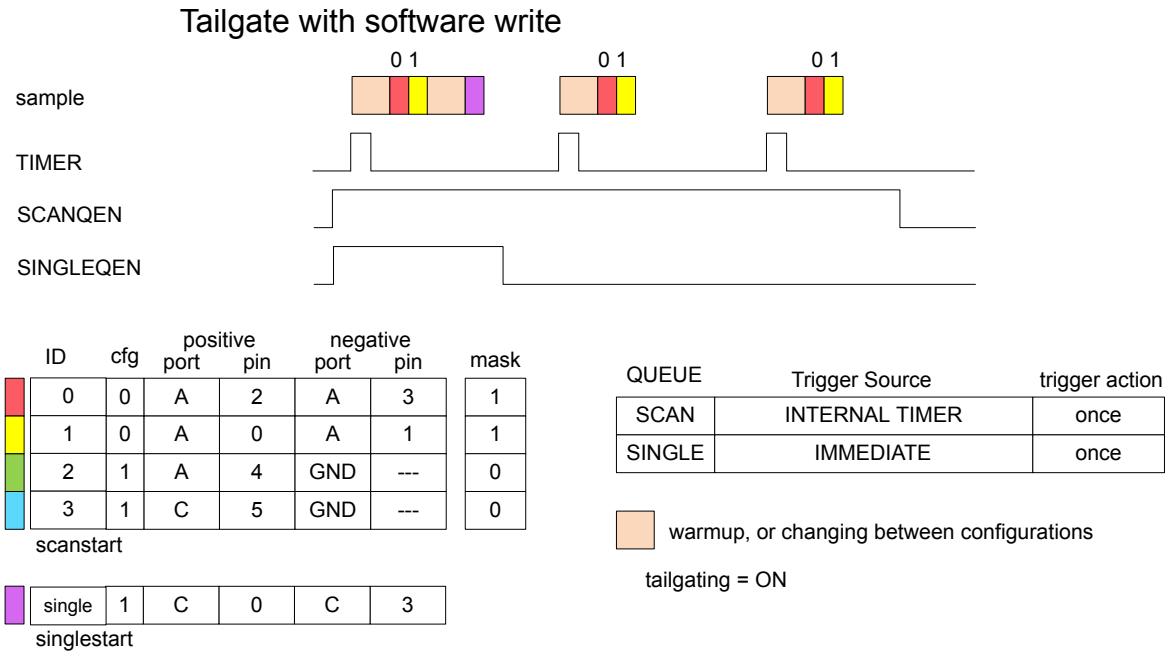


Figure 22.9. Simple Conversion with Tailgating Enabled

Another example, shown in Figure 22.10 Conversions with Tailgating Disabled on page 725, demonstrates how requests are handled on the different conversion queues with tailgating disabled.

In this example, the scan queue is being triggered on the internal timer while the single queue is being triggered on a PRS positive edge. Since tailgating is not enabled, the queues will be serviced on a first come first served basis. The first single queue trigger falls between two scan queue triggers and does not interfere with scan queue timing. The second single queue trigger happens just before the scan queue trigger. The IADC will complete this single queue conversion and delay the next scan queue conversions.

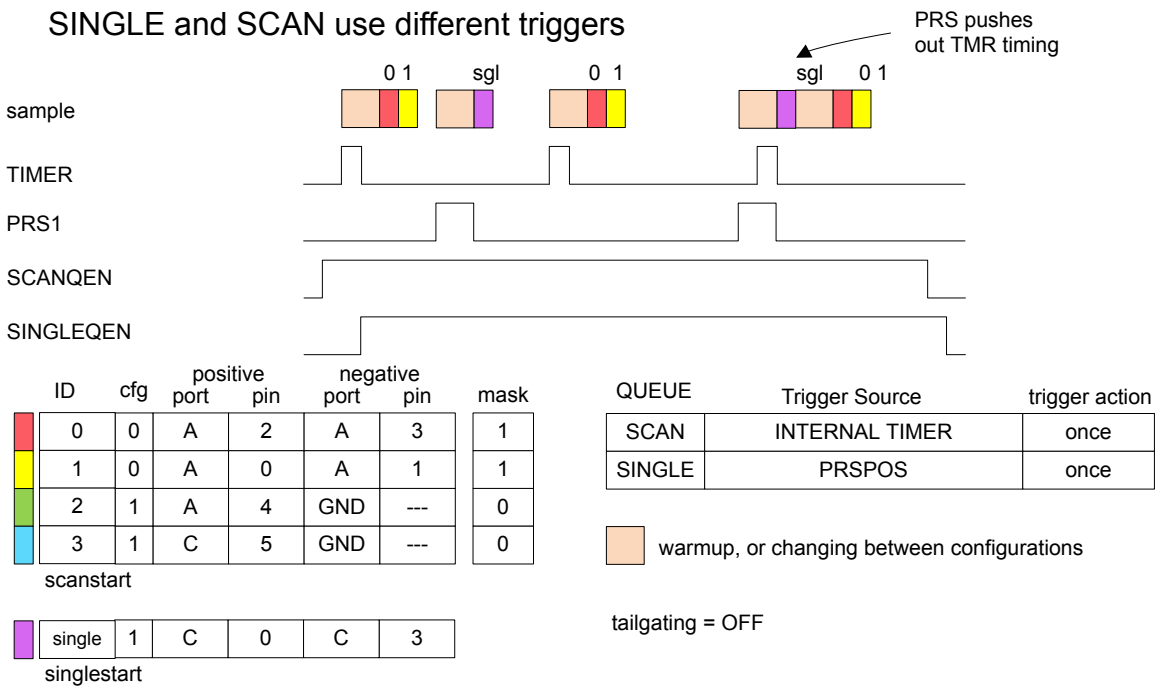


Figure 22.10. Conversions with Tailgating Disabled

Continuous Conversions

An example of continuous conversions triggered from the scan queue is shown in Figure 22.11 Continuous Conversions on page 725. In this example the SCANTRIGACTION field in IADC_TRIGGER is set to CONTINUOUS, and the conversion trigger source is software (SCANTRIGSEL = IMMEDIATE). When the scan queue is enabled with IADC_CMD_SCANSTART, the ADC warms up and then performs repeated back-to-back scans until software disables the scan queue using IADC_CMD_SCANSTOP. While this example shows only one channel converted continuously, it is possible to enable multiple channels for the scan sequence.

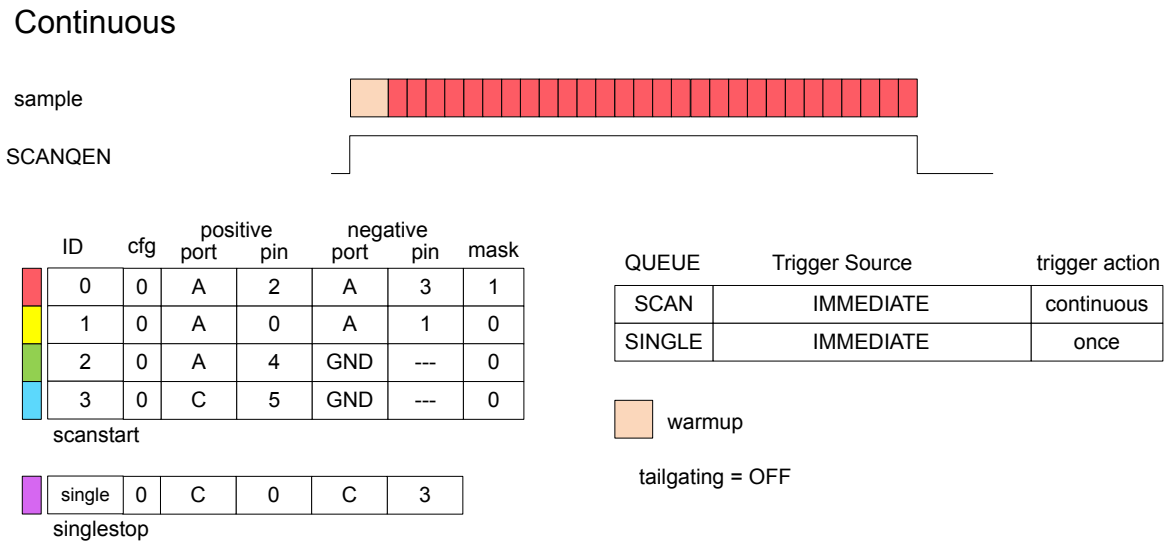


Figure 22.11. Continuous Conversions

22.3.4 Reference Selection and Analog Gain

The default IADC reference is to use the internal band gap circuit. The analog power supply voltage can also be used as a voltage reference. The reference voltage is selected using the REFSEL field in IADC_CFGx.

Table 22.2. Mode Settings

Reference	Description	Normal and High-Speed Mode Voltage Range	High-Accuracy Mode Voltage Range
VBGR	Internal	1.21V	1.21V
VDDX	Analog Power Supply	AVDD	AVDD
VREF	External	1.0V - AVDD (1.25V Nominal)	1.0V - 1.25V (1.25V Nominal)
VREF2P5	External	1.0V - AVDD (1.25V Nominal)	1.0V - 2.5V[1] (2.5V Nominal)

Note:
1. In high-accuracy mode with VREF2P5 selected, the AVDD supply must be at least 3V.

The IADC also has analog gain selection, controlled via the ANALOGGAIN field in IADC_CFGx. The analog gain can be set to 0.5x, 1x, 2x, 3x, or 4x. Note that 2x, 3x, and 4x gain modes may require slower ADC_CLK. The analog gain impacts where the full-scale input reading occurs. For example, with a 1.25 V external reference and ANALOGGAIN set to 2x, the analog input to the IADC is multiplied by a factor of 2, and a full-scale reading occurs at 1.25 V / 2 = 0.625 V. If ANALOGGAIN is set to 0.5x, the full-scale reading of the ADC will not occur until the input reaches 2.5 V. Note that the ADC is only capable of measuring inputs within the supply rails of the device. If the full scale is configured to be greater than the supply voltage, the maximum input will be limited to the supply.

The sampling capacitance (C_{sample}) is changed according to the analog gain setting.

Table 22.3. Input Sampling Capacitance vs. Analog Gain

Analog Gain Setting	Input Sampling Capacitance
0.5x	1 pF
1x	2 pF
2x	4 pF
3x	6 pF
4x	8 pF

Given the sampling capacitance and the front-end sampling rate (F_{sample}), the input impedance of the converter can be calculated as:

$Z_{in} = 1 / (C_{sample} * F_{sample})$

Note that the input is not sampled when the converter is inactive between conversions and operating with WARMUPMODE = NORMAL or KEEPINSTANDBY with longer intervals between conversion triggers can increase the effective input impedance of the converter.

22.3.5 Input and Configuration Selection

The IADC supports measurement on a number of internal and external signals. External signals are routed to GPIO through shared ABUS resources on the device, or (on some devices) through dedicated analog inputs available to the IADC block.

The single queue and the scan queue have separate registers available to select inputs and configurations. The IADC_SINGLE register is used to select the input and configuration for the single queue. The IADC_SCANx registers are used to select the inputs and configurations for each of the scan table entries. In both cases, the register contents and setup are similar. The PORTPOS and PINPOS fields are used to select a signal for the positive ADC input, while PORTNEG and PINNEG are used to select a signal for the negative ADC input. The CFG field selects which of the two configuration sets will be used with the input (i.e. configuration options specified in IADC_CFGx, IADC_SCALEx, and IADC_SCHEDx).

To perform single-ended conversions, the PORTNEG field should be set to GND. This indicates that the positive ADC input will be measured with reference to chip ground. PORTPOS and PINPOS should be used to select the desired input signal. The PINNEG field is not used for single-ended conversions.

To perform differential conversions, PORTPOS, PINPOS are used to select the positive input to the ADC, while PORTNEG and PINNEG are used to select the negative input. Note that there are two independent multiplexers in the ADC, and firmware cannot select two signals from the same multiplexer for a differential measurement. The "even" multiplexer consists of all EVEN ABUS selections, Supply voltage options, and GND. The "odd" multiplexer consists of all ODD ABUS selections and GND. One selection from each multiplexer is allowed on the positive and negative input. More detailed examples may be found in 22.3.5.3 ABUS Input Selection Examples.

The scan queue has one additional register, IADC_MASKREQ, to specify which of the 16 possible channel slots will be converted during a scan operation. Each channel in the scan queue is enabled by writing the corresponding bit in the IADC_MASKREQ register to 1. Enabled channels will be converted in sequence from lowest to highest, during a scan. See 22.3.5.4 Scan Queue for more details on using the scan queue.

22.3.5.1 External GPIO Inputs

GPIO input selections are routed through shared ABUS resources. In order for the IADC to use any GPIO as an input, the IADC must be allocated appropriate analog bus resources in the GPIO_ABUSALLOC, GPIO_BBUSALLOC, or GPIO_CDBUSALLOC registers. For example, if IADC0 will be using both odd and even numbered pins on GPIO port PA, then AEVEN0 and AODD0 in GPIO_ABUSALLOC could both be set to IADC0. This gives IADC0 access to these two buses. Generally, bus access is set to specific peripherals at configuration time and left alone - it is not normally required to change the bus allocation on the fly. If the IADC requests a pin from a bus that has not been allocated to the IADC, an error will be generated, the PORTALLOCERRIF in IADC_IF will be set, and any conversion result will be 0. For more details on analog bus structure and capabilities, refer to the GPIO section.

When the appropriate analog buses have been configured to route to the IADC, GPIO selection is a simple matter of programming the desired port and pin into the PORTPOS, PINPOS, PORTNEG, and PINNEG fields. For example, to configure a channel to convert the differential voltage between pins PA5 and PA4, PORTPOS = PORTA, PINPOS = 5, PORTNEG = PORTA, PINNEG = 4. If an invalid selection is made, a polarity error will be generated. More specific examples are described in 22.3.5.3 ABUS Input Selection Examples.

22.3.5.2 Internal and Dedicated Inputs

Internal signals and dedicated inputs are not routed through the shared ABUS resources. In general, these resources are selected directly by the settings of PORTPOS and PORTNEG, while the PINPOS and PINNEG fields are not used. When PORTPOS is set to SUPPLY, PINPOS is used to select which of the power supplies is connected. To facilitate power supply measurements using internal reference options, higher voltage supplies are attenuated by a factor of 4.

Table 22.4. Supply Selection (PORTPOS = SUPPLY)

PINPOS	Supply Connection	Voltage at Positive Input
0	AVDD	AVDD / 4
1	IOVDD	IOVDD / 4
2	VSS	VSS
3	VSS	VSS
4	DVDD	DVDD / 4
7	DECOUPLE	DECOUPLE

If an internal signal is selected for PORTPOS or PORTNEG, selecting GND on the opposite input will instruct the converter to perform a single-ended conversion. In the case where PORTPOS = GND, the IADC logic will automatically swap the direct input selected by PORTNEG to the positive input of the ADC. Otherwise, a differential conversion is performed with PORTPOS selecting the positive and PORTNEG selecting the negative input.

22.3.5.3 ABUS Input Selection Examples

When configuring to measure a single-ended signal through the ABUS, the positive input selection should always point to the desired input, and PORTNEG should be programmed to GND.

Correct configuration examples for single-ended conversions are shown in Figure 22.12 Single-Ended Port/Pin Selection Odd Channel on page 729 and Figure 22.13 Single-Ended Port/Pin Selection Even Channel on page 729. Note that the IADC logic will automatically swap the appropriate multiplexer to the positive input of the ADC.

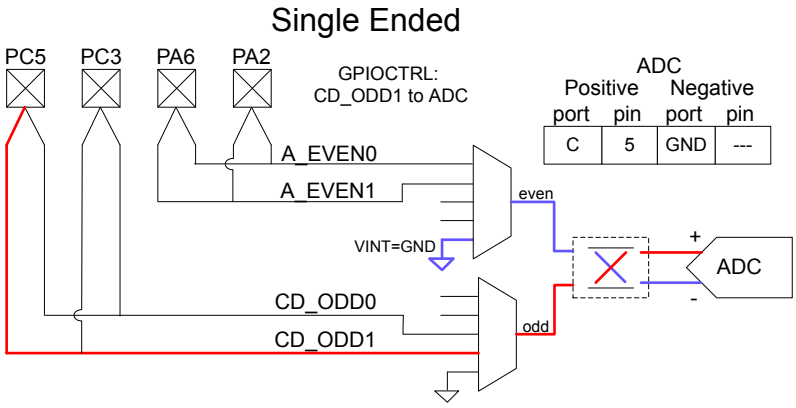


Figure 22.12. Single-Ended Port/Pin Selection Odd Channel

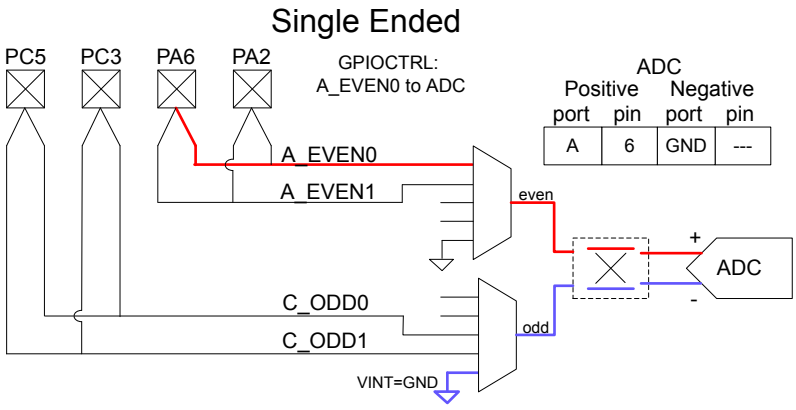


Figure 22.13. Single-Ended Port/Pin Selection Even Channel

Figure 22.14 Single-Ended Port/Pin Selection Polarity Error on page 730 shows an example where the PORTPOS input has been configured to GND, with PORTNEG and PINNEG configured for a GPIO pin. This will result in a polarity error (POLARITYERRIF in IADC_IF will be set) and any conversion result will be 0.

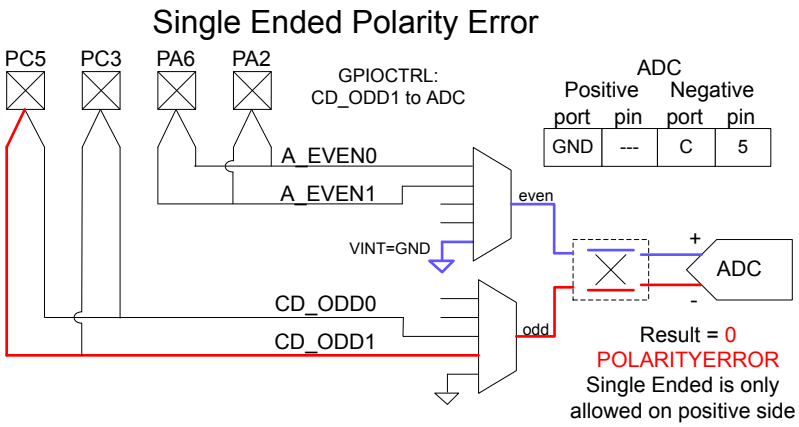


Figure 22.14. Single-Ended Port/Pin Selection Polarity Error

Correct configuration examples for differential conversions are shown in Figure 22.15 Differential Port/Pin Selection without Swap on page 730 and Figure 22.16 Differential Port/Pin Selection with Swap on page 731. In both these examples, the inputs were selected from one EVEN multiplexer channel and one ODD multiplexer channel. As with single-ended mode, the IADC logic will automatically swap the multiplexer connections to the IADC input if needed.

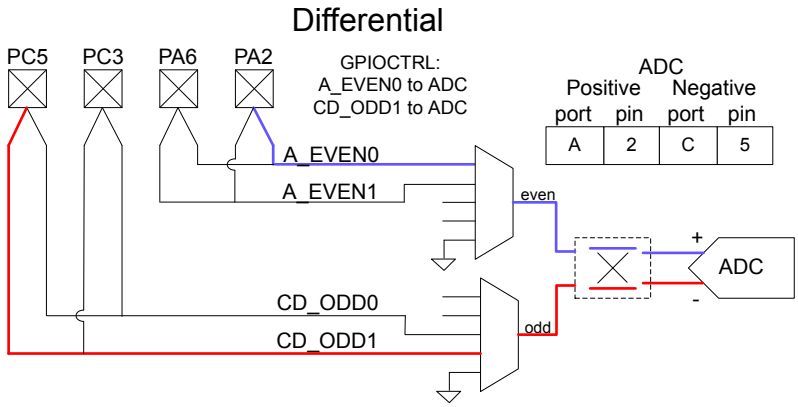


Figure 22.15. Differential Port/Pin Selection without Swap

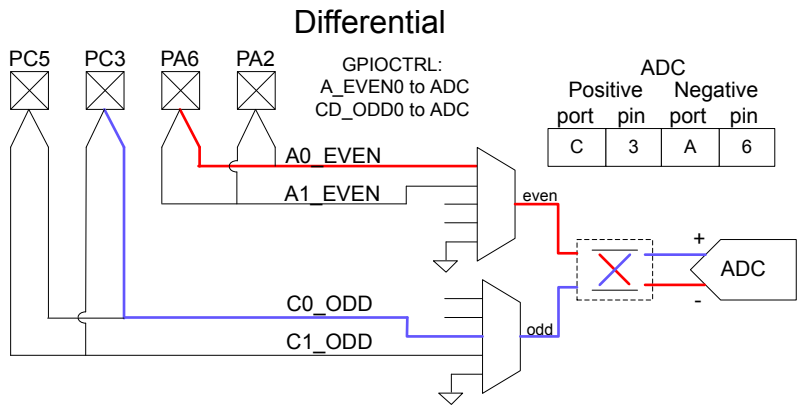


Figure 22.16. Differential Port/Pin Selection with Swap

Figure 22.17 Differential Port/Pin Selection Polarity Error on page 731 shows an example where the both the positive and the negative input selections point to ODD buses. Even though the IADC has been allocated both buses, they both route through the ODD input multiplexer and cannot be measured against one another. This will result in a polarity error (POLARITYERRIF in IADC_IF will be set) and any conversion result will be 0x7FFFF. Likewise, a polarity error will occur if both inputs are selected from EVEN buses.

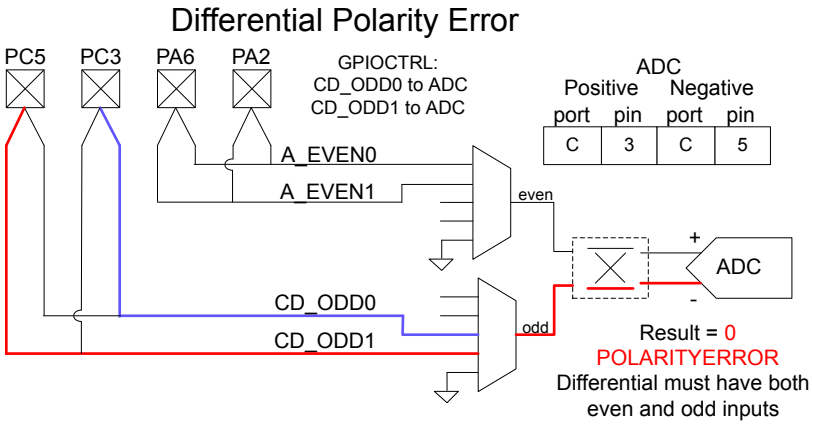


Figure 22.17. Differential Port/Pin Selection Polarity Error

22.3.5.4 Scan Queue

The scan queue allows the IADC to automatically convert up to 16 channels in sequence without CPU intervention. Input and configuration selection for each channel in the scan table is specified by the IADC_SCANx register for that channel (channel 0 is configured with IADC_SCAN0, channel 1 is configured with IADC_SCAN1, and so on). The IADC_MASKREQ register allows software to define which of the scan table entries (IADC_SCANx) to convert during a scan. For example, channels 0, 1, and 7 can be enabled by writing bits 0, 1, and 7 of IADC_MASKREQ to 1 (IADC_MASKREQ = 0x0083).

The IADC_SCANx registers must be configured when the IADC module is disabled (IADC_EN_EN = 0). IADC_MASKREQ can be written while IADC_EN_EN is set to 1. If a scan operation is in progress, MASKREQ will be synchronized and held until the current scan operation has completed. Then MASKREQ is copied into the STMASK register for the next scan operation. IADC_STMASK is the working copy of the MASKREQ used by the IADC during a scan. MASKREQ will only transfer to STMASK when the scan queue is not scanning and converting the scan table. IADC_STATUS_MASKWRITEPENDING can be used by software to see when the MASKREQ write has been transferred to STMASK. Writing a new MASKREQ in the middle of a scan will not corrupt the current scan. Software which writes to MASKREQ during a scan operation must ensure IADC_STATUS_MASKWRITEPENDING returns to 0 before updating IADC_MASKREQ again. Figure 22.18 MASKREQ Updates on page 732 shows a time line of when the MASKREQ write is updated.

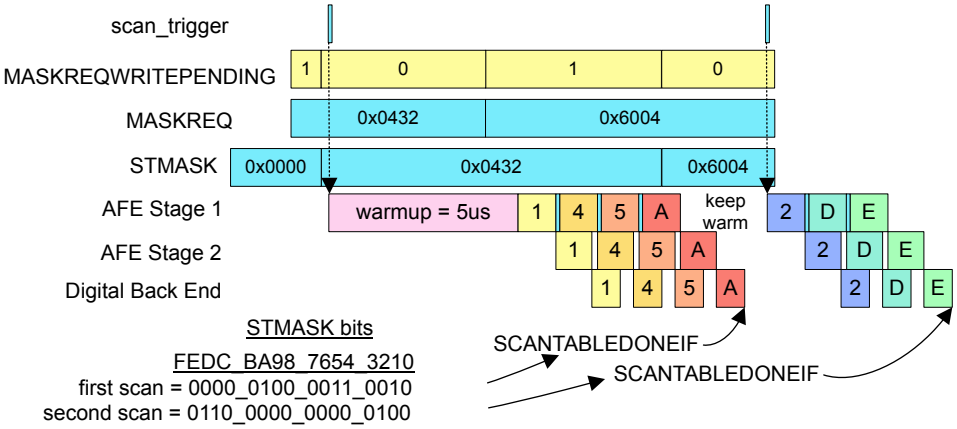


Figure 22.18. MASKREQ Updates

22.3.6 Gain and Offset Correction

The IADC has built in gain and offset correction capabilities. Each of the two configuration groups contains its own correction values stored in the IADC_SCALEx register, allowing the IADC to automatically apply the appropriate correction for the IADC configuration that is being used.

Gain correction is performed through a fixed-point 16-bit value with a range from 0.75x to 1.2499x. The 3 MSBs of the gain value are not directly writeable. The GAIN3MSB bit in IADC_SCALEx is used to select between 011 and 100 for the 3 MSBs, and the lower 13 bits are programmed directly into IADC_SCALEx_GAIN13LSB. Clearing GAIN3MSB to 0 selects the most significant bits of the gain as 011, representing a range from 0.75x to 0.9999x. Setting GAIN3MSB to 1 selects the most significant bits of the gain as 100, representing a range from 1.00x to 1.2499x.

Offset correction is controlled by the OFFSET field in IADC_SCALEx. It is important to note that the offset correction does not have a direct 1-to-1 relationship with the LSB of the IADC output, and depends on both the OSR and gain correction settings. The offset correction range is +/- 12.5% of full scale. OFFSET is encoded as a 2's complement, 18-bit number with the LSB representing $1 / 2^{20}$ of full scale. Thus, bit 8 of OFFSET aligns with bit 0 of the 12-bit IADC output word.

22.3.6.1 Using Production Calibration Parameters

IADC calibration is performed on every device during Silicon Labs production test and production calibration parameters are stored in the flash DI page. The production calibration values are useful for a wide variety of possible IADC configurations, but do not map directly to the offset and gain correction fields in the IADC_SCALEx registers. Software must calculate the actual offset and gain correction values from the factory calibration values.

22.3.6.1.1 Gain Correction

Gain error is measured during production test at various settings of ANALOGGAIN, and stored in the DEVINFO_IADC0GAIN0 and DEVINFO_IADC0GAIN1 locations. The GAINCANA1 field is used for 0.5x and 1x ANALOGGAIN settings, while GAINCANA2, GAINCANA3, and GANCANA4 are used for ANALOGGAIN settings of 2x, 3x, and 4x, respectively.

The GAINCANAn values are expressed as the full 16-bit fixed-point gain, and must be compressed before writing to the IADC_SCALEx register.

22.3.6.1.1.1 Gain Correction in Normal / High Speed Mode

When the IADC is operated in Normal / High Speed mode, a 1st order filter is employed in the decimation. The nominal gain value in these modes for all OSRHS settings is 1.0, or 0x8000 as expressed in the fixed-point 16-bit format. The IADC gain error is designed to be minimal with the digital gain correction set to 1.0 (GAIN3MSB = 1 and GAIN13LSB = 0). Tighter gain error is achieved by adjusting these values in IADC_SCALEx. Using this gain correction mechanism will result in a slight increase to the DNL of the converter, which is reduced by higher OSR settings.

To apply a factory-calibrated gain:

1. Read the appropriate GAINCANAn field from the DEVINFO locations for the selected ANALOGGAIN.
2. Write the MSB (bit 15) of GAINCANAn to GAIN3MSB in IADC_SCALEx.
3. Write the 13 LSBs (bits 12-0) of GAINCANAn to GAIN13LSB in IADC_SCALEx.

22.3.6.1.1.2 Gain Correction in High Accuracy Mode

When the IADC is operated in High Accuracy mode, a 2nd order filter is employed in the decimation. The nominal gain value of the filter is dependent on the OSRHA setting. The gain value stored in DEVINFO space must be adjusted before applying to the IADC_SCALEx register.

To apply a factory-calibrated gain:

1. Read the appropriate GAINCANAn field from the DEVINFO locations for the selected ANALOGGAIN.
2. Multiply the value by the OSR gain correction factor (ha_gain) found in Table 22.5 Ideal High Accuracy Gain Correction on page 733.
3. Write the MSB (bit 15) of the result to GAIN3MSB in IADC_SCALEx.
4. Write the 13 LSBs (bits 12-0) of the result to GAIN13LSB in IADC_SCALEx.

Table 22.5. Ideal High Accuracy Gain Correction

OSRHA Setting	OSR	16-bit OSR Gain Correction (ha_gain)
HIACC16	16 x	0x7879
HIACC32	32 x	0x7C1F
HIACC64	64 x	0x7E08
HIACC92	92 x	0x7A8E
HIACC128	128 x	0x7F02
HIACC256	256 x	0x7F80

22.3.6.1.2 Offset Correction

Offset is impacted by the selected ANALOGGAIN and OSR settings in IADC_CFGx, the GAIN3MSB and GAIN13LSB values in IADC_SCALEx, and the voltage reference. Offset is production calibrated for any combination of possibilities, but the OFFSET register value must be calculated for the given situation before it can be effectively used.

22.3.6.1.2.1 Offset Correction in Normal Mode

The production offset calibration consists of four 16-bit terms written to the DEVINFO space: OFFSETANA1NORM, OFFSETANA2NORM, OFFSETANA3NORM, and OFFSETANABASE. The following procedures will determine the setting for the OFFSET register based on production calibration values.

Step 1: Determine the offset gain adjustment term (off_gain) based on ANALOGGAIN.

For ANALOGGAIN set to 0.5x or 1x:

off_gain = 0

For ANALOGGAIN set to 2x, 3x, or 4x, off_gain is calculated as:

off_gain = OFFSETANA2NORM * (gain - 1)

The following table summarizes these equations:

Table 22.6. Offset Gain Adjustment

ANALOGGAIN Setting	Analog front-end gain	Offset Gain Adjustment Term (off_gain)
ANAGAIN0P5	0.5 x	0
ANAGAIN1	1 x	0
ANAGAIN2	2 x	OFFSETANA2NORM * 1
ANAGAIN3	3 x	OFFSETANA2NORM * 2
ANAGAIN4	4 x	OFFSETANA2NORM * 3

Step 2: Calculate the analog offset adjustment term (off_ana) based on OSR and off_gain.

For an OSR of 2x (OSRHS = 0):

off_ana = OFFSETANA1NORM + off_gain

For all other OSR settings, 4x - 64x:

off_ana = OFFSETANABASE + 2*(OFFSETANA3NORM - off_gain)/OSR

The following table expresses these equations:

Table 22.7. Analog Offset Adjustment

OSRHS Setting	OSR	Analog Offset Adjustment Term (off_ana)
HISPD2	2 x	OFFSETANA1NORM + off_gain
HISPD4	4 x	OFFSETANABASE + (OFFSETANA3NORM - off_gain)/2
HISPD8	8 x	OFFSETANABASE + (OFFSETANA3NORM - off_gain)/4
HISPD16	16 x	OFFSETANABASE + (OFFSETANA3NORM - off_gain)/8
HISPD32	32 x	OFFSETANABASE + (OFFSETANA3NORM - off_gain)/16
HISPD64	64 x	OFFSETANABASE + (OFFSETANA3NORM - off_gain)/32

Step 3: Compensate for reference voltage differences.

The off_ana term represents the offset at the input of the ADC, meaning that the reference voltage will have an impact on the magnitude of the offset at the output. Production calibration values are determined with a 1.25 V reference source. If a voltage significantly different than 1.25 V is used for V_{REF}, adjust the off_ana term by a factor of 1.25 / V_{REF}.

off_ana = off_ana * (1.25 / V_{REF})

Step 4: Calculate total offset by adding the analog offset to the systematic offset.

Systematic offset is a fixed number dependent on OSR, and calculated according to the following equation:

off_sys = 640*(256/OSR)

Total uncorrected offset (off_tot) is calculated by:

off_tot = (off_ana * 4 + off_sys)

Step 5: Apply gain error correction, if needed.

Before writing the OFFSET field, the total uncorrected offset must be multiplied by the gain calibration factor. If the gain calibration factor is equal to 1.0 (0x8000 in 16-bit hex, or GAIN3MSB = 1 and GAIN13LSB = 0), this step may be skipped. Otherwise, adjust off_tot according to the following equation:

off_tot = GAIN_FACTOR * (off_tot + 0x80000) - 0x80000

where GAIN_FACTOR = GAINCANAn / 32768.

Step 6: Write the offset correction value to the OFFSET field.

The OFFSET field holds an 18-bit 2's complement number, which should be the negation of the total offset, or -(off_tot). Before writing to the SCALE register, any leading sign bits should be masked off to avoid corrupting the programmed gain settings.

OFFSET = 0x3FFFF & (-off_tot)

22.3.6.1.2.2 Offset Correction in High Speed Mode

Offset correction for High Speed mode is identical to the procedure for Normal mode, with the exception of the calibration terms used in the DEVINFO space. The same OFFSETANABASE term is used, but OFFSETANA1HISPD, OFFSETANA2HISPD and OFFSETANA3HISPD are used instead of the OFFSETANAxNORM values. Refer to 22.3.6.1.2.1 Offset Correction in Normal Mode for the procedure, replacing OFFSETANAxNORM with OFFSETANAxHISPD.

22.3.6.1.2.3 Offset Correction in High Accuracy Mode

The production offset calibration for High Accuracy mode uses two 16-bit terms written to the DEVINFO space: OFFSETANA1HIACC and OFFSETANABASE. The following procedure will determine the setting for the OFFSET register based on production calibration values.

Step 1: Calculate the analog offset adjustment term (off_ana) based on the OSR setting (OSRHA in IADC_CFGn).

off_ana = OFFSETANABASE + OFFSETANA1HIACC / (2^{OSRHA})

The following table expresses this relationship:

Table 22.8. Analog Offset Adjustment

OSRHA Setting	OSR	Analog Offset Adjustment Term (off_ana)
HIACC16	16 x	OFFSETANABASE + OFFSETANA1HIACC
HIACC32	32 x	OFFSETANABASE + (OFFSETANA1HIACC / 2)
HIACC64	64 x	OFFSETANABASE + (OFFSETANA1HIACC / 4)
HIACC92	92 x	OFFSETANABASE + (OFFSETANA1HIACC / 8)
HIACC128	128 x	OFFSETANABASE + (OFFSETANA1HIACC / 16)
HIACC256	256 x	OFFSETANABASE + (OFFSETANA1HIACC / 32)

Step 2: Compensate for reference voltage differences.

The off_ana term represents the offset at the input of the ADC, meaning that the reference voltage will have an impact on the magnitude of the offset at the output. Production calibration values are determined with a 1.25 V reference source. If a voltage significantly different than 1.25 V is used for V_{REF}, adjust the off_ana term by a factor of 1.25 / V_{REF}.

off_ana = off_ana * (1.25 / V_{REF})

Step 3: Calculate total offset by adding the analog offset to the systematic offset.

Systematic offset is a fixed number dependent on OSR, and calculated according to the following equation:

off_sys = 0x40000 / (OSR / (OSR+1))

Total uncorrected offset (off_tot) is calculated by:

off_tot = (off_ana * 4 + off_sys)

Step 4: Apply gain error correction.

Before writing the OFFSET field, the total uncorrected offset must be multiplied by the gain calibration factor according to the following equation:

off_tot = GAIN_FACTOR * (off_tot + 0x80000) - 0x80000

where GAIN_FACTOR = GAINCANAn / 32768.

Step 5: Write the offset correction value to the OFFSET field.

The OFFSET field holds an 18-bit 2's complement number, which should be the negation of the total offset, or -(off_tot). Before writing to the SCALE register, any leading sign bits should be masked off to avoid corrupting the programmed gain settings.

OFFSET = 0x3FFFF & (-off_tot)

22.3.6.2 Calibration

Calibration can be performed in-system to correct for external errors and provide more accurate measurements. The general calibration procedure is as follows:

1. Configure the ADC to the desired mode, OSR, analog gain settings, reference source, etc.
2. Force the IADC to use bipolar output for the conversion: IADC_CFGx_TWOSCOMPL = FORCEBIPOLAR.
3. Set the initial offset to the maximum negative value (IADC_SCALEx_OFFSET = 0x20000), and the initial gain to 1.0 (GAIN3MSB = 1, GAIN13LSB = 0x0000). This will prevent output saturation when measuring full scale.
4. Apply a full-scale positive input to the IADC and perform a conversion (*result_fullscale*). Multiple conversions can be performed and averaged together to reduce any system-level noise.
5. Apply a zero input to the IADC and perform a conversion (*result_zero*). Multiple conversions can be performed and averaged together to reduce any system-level noise.
6. Calculate the gain correction factor: Divide the expected value by the difference in the measured values (*result_fullscale* - *result_zero*). Note that the offset adjustment in Step 3 will be canceled out by this calculation.
7. Write the gain correction factor to the IADC using the GAIN3MSB and GAIN13LSB fields in IADC_SCALEx.
8. Set IADC_SCALEx_OFFSET to 0x00000 in preparation for the offset calibration.
9. Apply the desired zero voltage to the IADC input and perform a conversion (*result_offset*). Multiple conversions can be performed and averaged together to reduce any system-level noise.
10. Multiply *result_offset* to convert to a 20-bit value (*result_offset_20*). For example, a 12-bit result should be multiplied by 256.
11. Negate *result_offset_20* and write the value to IADC_SCALEx_OFFSET.

    Note that the IADC_SCALEx_OFFSET field is 18 bits. If the result is greater than (2^{17} - 1) or less than (-2^{17}), the offset is too large to be corrected.

22.3.7 Output Data FIFOs

The single and scan queues each have a eight-word data FIFO. Conversions results are written to the output data FIFO associated with the queue. Single queue results are written to the single FIFO and scan queue results are written to the scan data FIFO. The two queues are identical in operation, but independent.

Conversion results are read from the single FIFO using IADC_SINGLEFIFODATA. Reading SINGLEFIFODATA will pop the oldest result from the FIFO. It is also possible to read the most recent valid data word using IADC_SINGLEDATA. Reading SINGLEDATA does not pop a conversion from the FIFO. Similarly, the scan FIFO results are read with IADC_SCANFIFODATA, which reads the oldest result and pops the FIFO. The most recent scan result can be read using IADC_SCANDATA.

When the single FIFO has valid data, the SINGLEFIFODV flag in IADC_STATUS is set to 1. When the scan FIFO has valid data SCANFIFODV in IADC_STATUS is set to 1. These data valid status bits are cleared automatically whenever the associated FIFO is empty. For more granular FIFO status, the number of data words present in the FIFO is indicated in IADC_SINGLEFIFOSTAT (for single FIFO) or IADC_SCANFIFOSTAT (for scan FIFO).

A programmable data level watermark is also available for the FIFOs, allowing hardware to trigger interrupts or DMA operations when a specified number of conversion results are available. The DVL field in register SINGLEFIFOCFG or SCANFIFOCFG sets the watermark level, between 1 and 4 conversions. If the number of valid entries in the FIFO reaches or exceeds the level set in DVL, the SINGLEFIFODVLIF (for single FIFO) or SCANFIFODVLIF (for scan FIFO) flag in the IADC_IF register will be set to 1. If enabled, an interrupt or DMA request will be triggered when the flag is set.

By default, DMA requests are turned off for operation in EM2 or EM3. However, the DMAWUFIFOSINGLE or DMAWUFIFOSCAN bits in SINGLEFIFOCFG or SCANFIFOCFG may be used to enable DMA operations in these lower energy modes.

Overflow and underflow status flags are also available in IADC_IF. An overflow condition occurs when an IADC conversion completes, but the associated FIFO is already full. In an overflow case the SINGLEFIFOOFIF or SCANFIFOOFIF flag will be set. The most recent conversion will still be available in the SINGLEDATA or SCANDATA register, but the FIFO will not be updated with the new data. An underflow condition occurs when software or hardware attempts to read from an empty FIFO. In an underflow case the SINGLEFIFOUFIF or SCANFIFOUFIF flag will be set.

22.3.7.1 Data Alignment and Channel ID

The IADC has data alignment options and the ability to include a channel ID along with the conversion data. For the single queue, alignment and channel ID are configured in the IADC_SINGLEFIFOCFG register. For the scan queue, alignment and channel ID are configured in the IADC_SCANFIFOCFG register.

The ALIGNMENT bit field specifies the data justification and the number of data bits as shown in Figure 22.19 Data Alignment on page 738. By default, the converter will produce 12-bit right-justified data, corresponding to ALIGNMENT = RIGHT12.

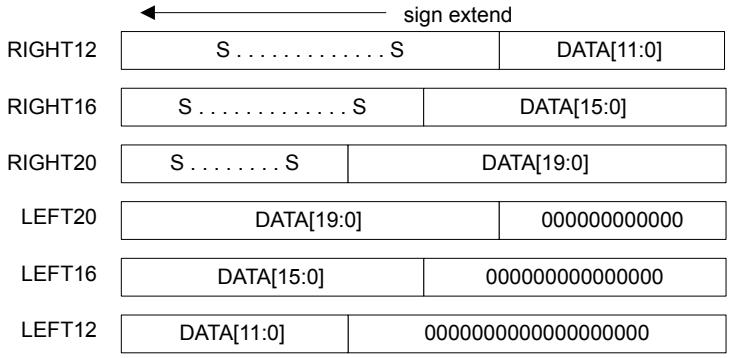


Figure 22.19. Data Alignment

The SHOWID bit controls whether the conversion channel ID is included in the output data word. This option is primarily used with the scan FIFO to help software determine which channel each conversion result came from. If SHOWID is enabled for single conversions, the ID will always be set to 0x20. Figure 22.20 Data Alignment With ID on page 738 shows output data formatting including the ID, when SHOWID = 1.

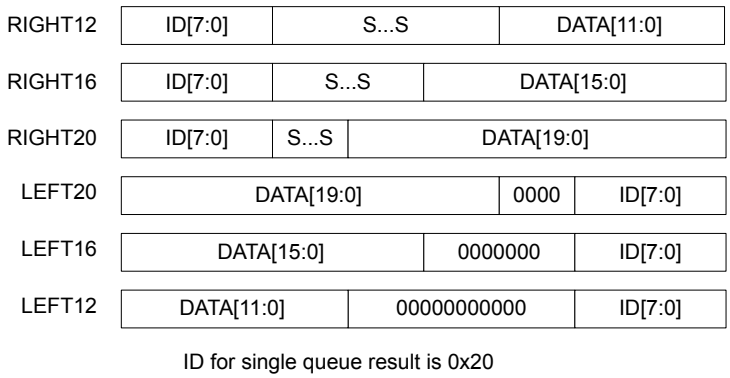


Figure 22.20. Data Alignment With ID

22.3.7.2 Output Polarity

The output polarity of the IADC is controlled by the TWOSCOMPL field in the IADC_CFGx register. The IADC supports unipolar and bipolar output formatting independent of the input configuration. By default, the TWOSCOMPL field is set to AUTO, meaning that single-ended conversions will produce unipolar output, and differential conversions will produce bipolar output. The polarity can be forced to unipolar or bipolar mode by setting TWOSCOMPL to FORCEUNIPOLAR or FORCEBIPOLAR, respectively.

Unipolar samples are unsigned integers representing zero to positive full-scale. Bipolar samples are two's-complement signed integers, representing negative full-scale to positive full-scale. Using unipolar mode on a differential input signal allows for more dynamic range when the signal is positive, but will saturate to zero when the signal is negative.

Note: If bipolar output is used with a single-ended input configuration, it is possible to see negative output values when the input is close to ground. However, the input voltage is still limited by the supply range of the device.

22.3.7.3 Digital Accumulation and Averaging

The IADC may optionally accumulate and average several conversion results before posting an output word to the FIFO. Digital averaging is controlled by the DIGAVG field in the IADC_CFGx register. It can be configured to average 1, 2, 4, 8, or 16 samples. The IADC will collect the number of samples specified by DIGAVG on the selected channel slot back-to-back, and produce only one averaged output word.

22.3.7.4 Output Resolution

The usable output resolution of the IADC is a minimum of 12 bits, when the oversampling ratio is set to 2 and no digital averaging is used (DIGAVG = AVG1). An extra bit of output resolution is produced for every power of 2 increase in either of these settings. In other words, the output resolution of the ADC can be determined as:

$$\text{Output Resolution} = 11 + \log_2(\text{OversamplingRatio} \times \text{DigitalAveraging})$$

The MSB is always left-aligned within the DATA field, and the output word will be truncated to 12, 16, or 20 bits, as shown in Figure 22.19 Data Alignment on page 738 and Figure 22.20 Data Alignment With ID on page 738. When using 16 or 20 bit alignment with lower oversampling ratio and digital averaging settings, LSBs of the output can contain residual effects of the offset and gain computation. These residual effects do not represent additional information about the input signal. Any extra LSBs can be masked to 0 by software.

Table 22.9. Output Resolution Masking Examples

Alignment Setting	Oversampling Ratio	Digital Averaging	Number of averaged samples	Output Resolution	Recommended Mask for DATA field
16-bit	2x	1x	2	12 bits	0xFFF0
16-bit	8x	2x	16	15 bits	0xFFFE
20-bit	2x	1x	2	12 bits	0xFFF00
20-bit	16x	4x	64	17 bits	0xFFFF8

22.3.7.5 Flushing the FIFOs

Each FIFO has a command bit in the IADC_CMD register that can be used to trigger a FIFO flush. The FIFO data may be flushed independently for each queue. To flush a FIFO:

1. The IADC must be enabled with the clock running.
2. Disable the queue associated with the FIFO using the SCANSTOP or SINGLESTOP bits in the IADC_CMD register.
3. Ensure the queue is disabled by reading the associated flag in the IADC_STATUS register (SINGLEQEN or SCANQEN).
4. Set the command bit to flush the desired FIFO (SINGLEFIFOFLUSH or SCANFIFOFLUSH) in the IADC_CMD register.
5. Wait for the corresponding status bit (SINGLEFIFOFLUSHING or SCANFIFOFLUSHING) in IADC_STATUS to go low.

22.3.8 Window Compare

The IADC has a window comparison unit that can trigger interrupts conditional on the output data of the converter. The window comparison unit has two thresholds - greater than or equal (ADGT), and less than or equal (ADLT), which are programmable through the IADC_CMPTHR register. The ADGT and ADLT thresholds always use a 16 bit, left-justified format, regardless of the format specified by the FIFO. The 12-bit conversion result will be compared against the upper 12 bits of the window comparator.

The window comparison unit is active on the ADC output on a conversion-by-conversion basis, and is shared between the two FIFOs. It is not possible to set different window comparison thresholds for different channels or for each FIFO. However, each channel specified in the IADC has a CMP bit field to enable the window comparison on results from that channel. For example, it is possible to only apply the window comparison and associated interrupt to scan channel #3 by setting the CMP field in IADC_SCAN3 to 1. When the CMP field associated with a channel is 0, the window comparator will not be active for results from that channel.

The window comparator supports conditional triggering on output results which are inside or outside a specified window. When ADLT is greater than or equal to ADGT, the comparator will trigger on an "inside" condition, or when DATA <= ADLT **and** DATA >= GT. When ADLT is less than ADGT, the comparator will trigger on an "outside" condition, or when DATA <= ADLT **or** DATA >= GT.

Figure 22.21 Window Comparison Examples on page 740 shows different configurations of the ADLT and ADGT values and the resulting windows. When the window comparator detects that the appropriate conditions are met (shown by the shaded region in the figure), it will generate an interrupt via the SINGLECMPIF flag for conversions on the single queue, or via the SCANCMPIF flag for conversions on the scan queue.

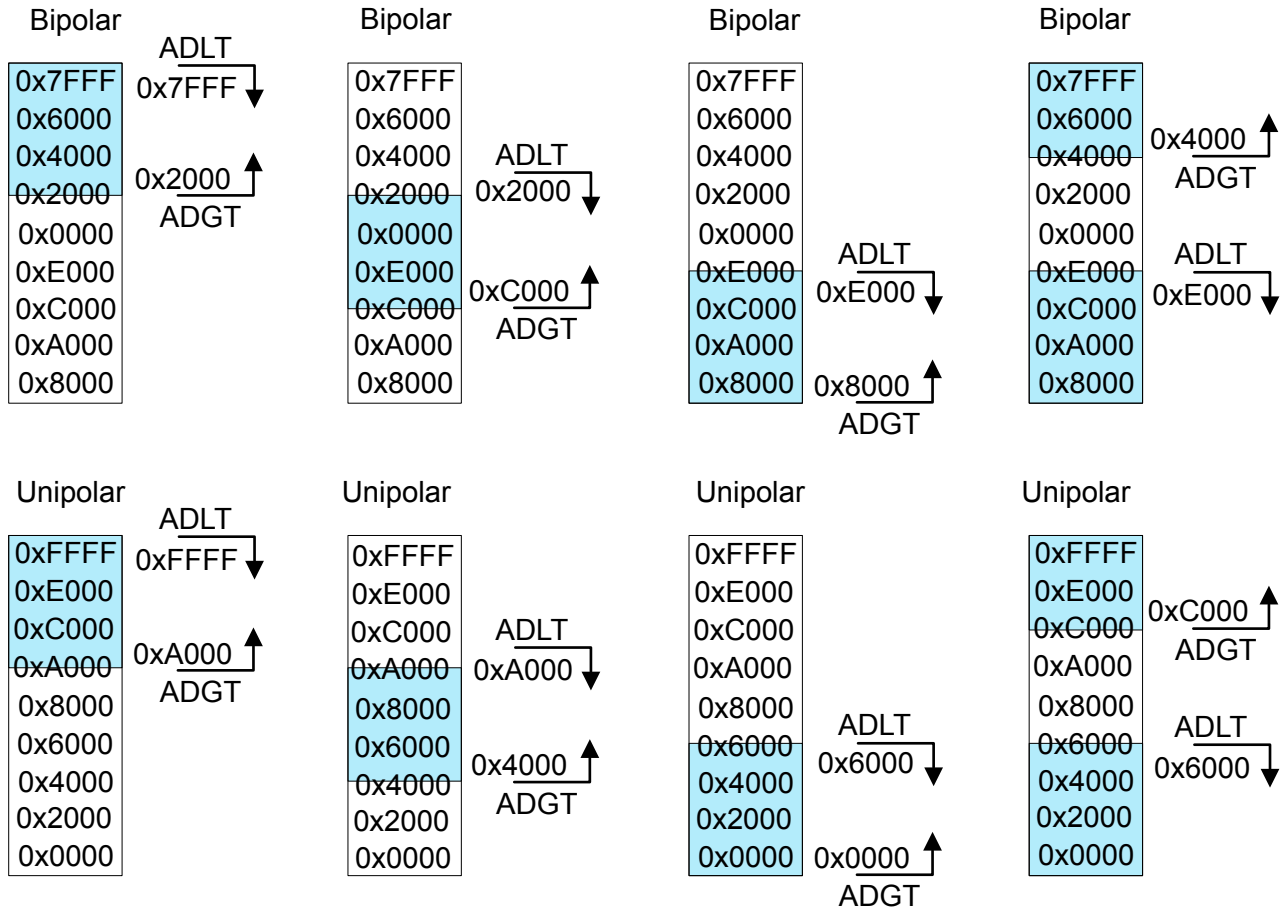


Figure 22.21. Window Comparison Examples

22.3.9 Interrupts

Interrupts are enabled in the IADC_IEN register, allowing interrupts to be generated on several different IADC conditions. Each of the flags in IADC_IF has a corresponding enable bit in the IADC_IEN register. A brief overview of the available interrupt sources is shown in the list below; more details can be found in the relevant sections of this chapter.

- SINGLEFIFODVLIF - The single FIFO watermark specified in SINGLEFIFOCFG_DVL has been reached or exceeded.
- SCANFIFODVLIF - The scan FIFO watermark specified in SCANFIFOCFG_DVL has been reached or exceeded.
- SINGLECMPIF - A conversion result from the single queue tripped the window comparator.
- SCANCMPIF - A conversion result from the scan queue tripped the window comparator.
- SCANENTRYDONEIF - A scan queue conversion has completed.
- SCANTABLEDONEIF - A scan queue operation has completed (all channels specified in the scan mask have been converted once).
- POLARITYERRIF - A channel polarity selection error has occurred (two channels from the EVEN multiplexer or two channels from the ODD multiplexer were selected for positive and negative inputs).
- PORTALLOCERRIF - A port allocation error has occurred (a pin not allocated to the IADC in the GPIO bus allocation registers was requested).
- SINGLEFIFOOFIF - A single FIFO overflow has occurred.
- SCANFIFOOFIF - A scan FIFO overflow has occurred.
- SINGLEFIFOUFIF - A single FIFO underflow has occurred.
- SCANFIFOUFIF - A scan FIFO underflow has occurred.
- EM23ABORTERRORIF - The system entered EM2 or EM3 while the IADC was converting and using a clock not supported in EM2 or EM3.

Hardware sets the interrupt flags in IADC_IF, and the flags remain set (sticky) until cleared by software. The interrupts flags should be cleared before enabling the IADC to remove any previous interrupt history. Clearing or setting interrupt bits can be done by writing to IADC_IF with a set or clear mask.

22.3.10 LESENSE Interface

The LESENSE peripheral can be set up to trigger IADC0 conversions and use data from IADC0 to evaluate sensor status. The channel scanner hardware is used by LESENSE in this mode and the IADC SCAN trigger must be set to LESENSE. The SCAN queue can only be used for LESENSE when operated in this mode, but the SINGLE queue may still be used independently.

When an LESENSE channel is active, the OFFSET field in LESENSE_CHx_INTERACT field is used to determine which of the ADC's 16 scanner channels is sampled. OFFSET = 0 corresponds to IADC_SCAN0, OFFSET = 1 corresponds to IADC_SCAN1, and so on.

The IADC sample is triggered when the sample delay configured in LESENSE_CHx_TIMING_SAMPLEDLY has expired. Results from the conversion are sent to the LESENSE result FIFO for further processing by LESENSE, and are not available in the SCAN FIFO.

22.4 IADC Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	IADC_IPVERSION	R	IPVERSION
0x004	IADC_EN	RW ENABLE	Enable
0x008	IADC_CTRL	RW CONFIG	Control
0x00C	IADC_CMD	W SYNC	Command
0x010	IADC_TIMER	RW CONFIG	Timer
0x014	IADC_STATUS	RH	Status
0x018	IADC_MASKREQ	RW SYNC	Mask Request
0x01C	IADC_STMASK	RH SYNC	Scan Table Mask
0x020	IADC_CMPTHR	RW CONFIG	Digital Window Comparator Threshold
0x024	IADC_IF	RWH INTFLAG	Interrupt Flags
0x028	IADC_IEN	RW	Interrupt Enable
0x02C	IADC_TRIGGER	RW CONFIG	Trigger
0x048	IADC_CFGx	RW CONFIG	Configuration
0x050	IADC_SCALEx	RW CONFIG	Scaling
0x054	IADC_SCHEDx	RW CONFIG	Scheduling
0x070	IADC_SINGLEFIFOCFG	RW CONFIG	Single FIFO Configuration
0x074	IADC_SINGLEFIFODATA	RH(r)	Single FIFO DATA
0x078	IADC_SINGLEFIFOSTAT	RH	Single FIFO Status
0x07C	IADC_SINGLEDATA	RH SYNC	Single Data
0x080	IADC_SCANFIFOCFG	RW CONFIG	Scan FIFO Configuration
0x084	IADC_SCANFIFODATA	RH(r)	Scan FIFO Read Data
0x088	IADC_SCANFIFOSTAT	RH	Scan FIFO Status
0x08C	IADC_SCANDATA	RH SYNC	Scan Data
0x098	IADC_SINGLE	RW SYNC	Single Queue Port Selection
0x0A0	IADC_SCANx	RW CONFIG	SCAN Entry
0x1000	IADC_IPVERSION_SET	R	IPVERSION
0x1004	IADC_EN_SET	RW ENABLE	Enable
0x1008	IADC_CTRL_SET	RW CONFIG	Control
0x100C	IADC_CMD_SET	W SYNC	Command
0x1010	IADC_TIMER_SET	RW CONFIG	Timer
0x1014	IADC_STATUS_SET	RH	Status
0x1018	IADC_MASKREQ_SET	RW SYNC	Mask Request
0x101C	IADC_STMASK_SET	RH SYNC	Scan Table Mask
0x1020	IADC_CMPTHR_SET	RW CONFIG	Digital Window Comparator Threshold
0x1024	IADC_IF_SET	RWH INTFLAG	Interrupt Flags

Offset	Name	Type	Description
0x1028	IADC_IEN_SET	RW	Interrupt Enable
0x102C	IADC_TRIGGER_SET	RW CONFIG	Trigger
0x1048	IADC_CFGx_SET	RW CONFIG	Configuration
0x1050	IADC_SCALEx_SET	RW CONFIG	Scaling
0x1054	IADC_SCHEDx_SET	RW CONFIG	Scheduling
0x1070	IADC_SINGLEFIFOCFG_SET	RW CONFIG	Single FIFO Configuration
0x1074	IADC_SINGLEFIFODATA_SET	RH(r)	Single FIFO DATA
0x1078	IADC_SINGLEFIFOSTAT_SET	RH	Single FIFO Status
0x107C	IADC_SINGLEDATA_SET	RH SYNC	Single Data
0x1080	IADC_SCANFIFOCFG_SET	RW CONFIG	Scan FIFO Configuration
0x1084	IADC_SCANFIFODATA_SET	RH(r)	Scan FIFO Read Data
0x1088	IADC_SCANFIFOSTAT_SET	RH	Scan FIFO Status
0x108C	IADC_SCANDATA_SET	RH SYNC	Scan Data
0x1098	IADC_SINGLE_SET	RW SYNC	Single Queue Port Selection
0x10A0	IADC_SCANx_SET	RW CONFIG	SCAN Entry
0x2000	IADC_IPVERSION_CLR	R	IPVERSION
0x2004	IADC_EN_CLR	RW ENABLE	Enable
0x2008	IADC_CTRL_CLR	RW CONFIG	Control
0x200C	IADC_CMD_CLR	W SYNC	Command
0x2010	IADC_TIMER_CLR	RW CONFIG	Timer
0x2014	IADC_STATUS_CLR	RH	Status
0x2018	IADC_MASKREQ_CLR	RW SYNC	Mask Request
0x201C	IADC_STMASK_CLR	RH SYNC	Scan Table Mask
0x2020	IADC_CMPTHR_CLR	RW CONFIG	Digital Window Comparator Threshold
0x2024	IADC_IF_CLR	RWH INTFLAG	Interrupt Flags
0x2028	IADC_IEN_CLR	RW	Interrupt Enable
0x202C	IADC_TRIGGER_CLR	RW CONFIG	Trigger
0x2048	IADC_CFGx_CLR	RW CONFIG	Configuration
0x2050	IADC_SCALEx_CLR	RW CONFIG	Scaling
0x2054	IADC_SCHEDx_CLR	RW CONFIG	Scheduling
0x2070	IADC_SINGLEFIFOCFG_CLR	RW CONFIG	Single FIFO Configuration
0x2074	IADC_SINGLEFIFODATA_CLR	RH(r)	Single FIFO DATA
0x2078	IADC_SINGLEFIFOSTAT_CLR	RH	Single FIFO Status
0x207C	IADC_SINGLEDATA_CLR	RH SYNC	Single Data
0x2080	IADC_SCANFIFOCFG_CLR	RW CONFIG	Scan FIFO Configuration
0x2084	IADC_SCANFIFODATA_CLR	RH(r)	Scan FIFO Read Data
0x2088	IADC_SCANFIFOSTAT_CLR	RH	Scan FIFO Status

Offset	Name	Type	Description
0x208C	IADC_SCANDATA_CLR	RH SYNC	Scan Data
0x2098	IADC_SINGLE_CLR	RW SYNC	Single Queue Port Selection
0x20A0	IADC_SCANx_CLR	RW CONFIG	SCAN Entry
0x3000	IADC_IPVERSION_TGL	R	IPVERSION
0x3004	IADC_EN_TGL	RW ENABLE	Enable
0x3008	IADC_CTRL_TGL	RW CONFIG	Control
0x300C	IADC_CMD_TGL	W SYNC	Command
0x3010	IADC_TIMER_TGL	RW CONFIG	Timer
0x3014	IADC_STATUS_TGL	RH	Status
0x3018	IADC_MASKREQ_TGL	RW SYNC	Mask Request
0x301C	IADC_STMASK_TGL	RH SYNC	Scan Table Mask
0x3020	IADC_CMPTHR_TGL	RW CONFIG	Digital Window Comparator Threshold
0x3024	IADC_IF_TGL	RWH INTFLAG	Interrupt Flags
0x3028	IADC_IEN_TGL	RW	Interrupt Enable
0x302C	IADC_TRIGGER_TGL	RW CONFIG	Trigger
0x3048	IADC_CFGx_TGL	RW CONFIG	Configuration
0x3050	IADC_SCALEx_TGL	RW CONFIG	Scaling
0x3054	IADC_SCHEDx_TGL	RW CONFIG	Scheduling
0x3070	IADC_SINGLEFIFOCFG_TGL	RW CONFIG	Single FIFO Configuration
0x3074	IADC_SINGLEFIFODATA_TGL	RH(r)	Single FIFO DATA
0x3078	IADC_SINGLEFIFOSTAT_TGL	RH	Single FIFO Status
0x307C	IADC_SINGLEDATA_TGL	RH SYNC	Single Data
0x3080	IADC_SCANFIFOCFG_TGL	RW CONFIG	Scan FIFO Configuration
0x3084	IADC_SCANFIFODATA_TGL	RH(r)	Scan FIFO Read Data
0x3088	IADC_SCANFIFOSTAT_TGL	RH	Scan FIFO Status
0x308C	IADC_SCANDATA_TGL	RH SYNC	Scan Data
0x3098	IADC_SINGLE_TGL	RW SYNC	Single Queue Port Selection
0x30A0	IADC_SCANx_TGL	RW CONFIG	SCAN Entry

22.5 IADC Register Description

22.5.1 IADC_IPVERSION - IPVERSION

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x3																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x3	R	**IP version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

22.5.2 IADC_EN - Enable

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status**
	When EN is cleared, DISABLING status is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGS and FIFOs.			
0	EN	0x0	RW	**Enable IADC Module**
	The EN bit enables the module. Software should write to CONFIG type registers before setting the EN bit. Software should write to SYNC type registers only after setting the EN bit.			

Value	Mode	Description
0	DISABLE	Disable
1	ENABLE	Enable

22.5.3 IADC_CTRL - Control

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0x0								0x0																	0x0		0x0	0x0	0x0	0x0
Access		RW								RW																	RW		RW	RW	RW	RW
Name		HSCLKRATE								TIMEBASE																	WARMUPMODE		DBGHALT	ADCCLKSUSPEND1	ADCCLKSUSPEND0	EM23WUCONVERT

Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
30:28	HSCLKRATE	0x0	RW	**High Speed Clock Rate**

Ratio to divide incoming CLK_CMU_ADC clock by. The resulting clock (CLK_SRC_ADC) must be 40 MHz or less.

Value	Mode	Description
0	DIV1	Use CMU_CLK_ADC directly. The source clock must be 40 MHz or less.
1	DIV2	Divide CMU_CLK_ADC by 2 before using it. The resulting CLK_SRC_ADC must be 40 MHz or less.
2	DIV3	Divide CMU_CLK_ADC by 3 before using it. The resulting CLK_SRC_ADC must be 40 MHz or less.
3	DIV4	Divide CMU_CLK_ADC by 4 before using it. The resulting CLK_SRC_ADC must be 40 MHz or less.

27:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:16	TIMEBASE	0x0	RW	**Time Base**

ADC clock cycles (TIMEBASE + 1) needed to generate a 1 us interval for warm up and start up timing. Does not allow less than 2 cycles. A setting of 0x0 (1 cycle) is replaced with 0x1 (2 cycles).

15:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	WARMUPMODE	0x0	RW	**Warmup Mode**

Select the warmup mode for the ADC.

Value	Mode	Description
0	NORMAL	Shut down the IADC after conversions have completed.
1	KEEPINSTANDBY	Switch to standby mode after conversions have completed. The next warmup time will require 1us.
2	KEEPWARM	Keep IADC fully powered after conversions have completed.

3	DBGHALT	0x0	RW	**Debug Halt**

Bit	Name	Reset	Access	Description
	ADC behavior when halted by debugger.			

Value	Mode	Description
0	NORMAL	Continue operation as normal during debug mode
1	HALT	Complete the current conversion and then halt during debug mode

Bit	Name	Reset	Access	Description
2	ADCCLKSUSPEND1	0x0	RW	**ADC_CLK Suspend - PRS1**
	This only functions with single trigger select set to PRSPOS or PRSNEG. In EM0 and EM1, this gates the local clock while clock source remains running. In EM2 and EM3, this disables the clock source until the PRSPOS or PRSNEG event is detected. This bit has no effect if the local IADC timer is running.			

Value	Mode	Description
0	PRSWUDIS	Normal mode which does not disable the ADC_CLK.
1	PRSWUEN	ADCCLKWUEN will gate off ADC_CLK until the trigger is detected provided the internal timer is not selected as the trigger. Once the trigger is detected the ADC_CLK will be started, the band gap will be started, the ADC will be warmed up, and the SCAN Table and the Single entry will be converted. Once the conversions are done, the ADC_CLK will be gated off.

Bit	Name	Reset	Access	Description
1	ADCCLKSUSPEND0	0x0	RW	**ADC_CLK Suspend - PRS0**
	This only functions with scan trigger select set to PRSPOS or PRSNEG. In EM0 and EM1, this gates the local clock while clock source remains running. In EM2 and EM3, this disables the clock source until the PRSPOS or PRSNEG event is detected. This bit has no effect if the local IADC timer is running.			

Value	Mode	Description
0	PRSWUDIS	Normal mode which does not disable the ADC_CLK.
1	PRSWUEN	ADCCLKWUEN will gate off ADC_CLK until the trigger is detected provided the internal timer is not selected as the trigger. Once the trigger is detected the ADC_CLK will be started, the band gap will be started, the ADC will be warmed up, and the SCAN Table and the Single entry will be converted. Once the conversions are done, the ADC_CLK will be gated off.

Bit	Name	Reset	Access	Description
0	EM23WUCONVERT	0x0	RW	**EM23 Wakeup on Conversion**
	EM23 wake up on conversion			

Value	Mode	Description
0	WUDVL	When using suspend mode, conversions performed in EM2 or EM3 should not wake up the DMA until the FIFO's DVL setting is reached. This saves more power for large OSR settings or infrequent sampling.
1	WUCONVERT	When using suspend mode, conversions performed in EM2 or EM3 will wake up the DMA and keep it awake until the conversions are done, regardless of the DVL setting. This mode burns more power, but it is useful when the conversion rate is faster than the time for the DMA to cycle through wake up and going back to sleep as it converts more than 4 scan table entries. Without using the wake up on conversion mode, the FIFO may overflow while the DMA is going in and out of sleep.

22.5.4 IADC_CMD - Command

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset							0x0	0x0							0x0	0x0												0x0	0x0		0x0	0x0
Access							W(nB)	W(nB)							W(nB)	W(nB)												W(nB)	W(nB)		W(nB)	W(nB)
Name							SCANFIFOFLUSH	SINGLEFIFOFLUSH							TIMERDIS	TIMEREN												SCANSTOP	SCANSTART		SINGLESTOP	SINGLESTART

Bit	Name	Reset	Access	Description
31:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25	SCANFIFOFLUSH	0x0	W(nB)	**Flush the Scan FIFO**
	Flush the Scan FIFO. The IADC must be enabled, not suspended, and the IADC clock must be running. Operation has completed when STATUS.SCANFIFOFLUSHING has gone low. The scan queue should be disabled. Any incoming scan queue data will be discarded during the flush.			
24	SINGLEFIFOFLUSH	0x0	W(nB)	**Flush the Single FIFO**
	Flush the Single FIFO. The IADC must be enabled, not suspended, and the IADC clock must be running. Operation has completed when STATUS.SINGLEFIFOFLUSHING has gone low. The Single queue should be disabled. Any incoming single queue data will be discarded during the flush.			
23:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	TIMERDIS	0x0	W(nB)	**Timer Disable**
	Disable the local timer and reset the counter to timer reload value.			
16	TIMEREN	0x0	W(nB)	**Timer Enable**
	Enable the local timer.			
15:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	SCANSTOP	0x0	W(nB)	**Scan Queue Stop**
	Stop the Scan queue. Disables Scan triggers and clears pending conversions in the Scan queue. Any conversion that has already started will continue until it is complete. If the scan queue is stopped before all entries of the scan table have completed, the remaining entries will not be converted.			
3	SCANSTART	0x0	W(nB)	**Scan Queue Start**
	Start the Scan queue. Enables triggering of the Scan queue.			
2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	SINGLESTOP	0x0	W(nB)	**Single Queue Stop**
	Stop the Single queue. Disables Single queue triggers and clears pending conversions in the Single queue. Any conversion that has already started will continue until it is complete.			
0	SINGLESTART	0x0	W(nB)	**Single Queue Start**

Bit	Name	Reset	Access	Description
	Start the Single queue. Enables triggering of the Single queue.			

22.5.5 IADC_TIMER - Timer

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	RW															
Name																	TIMER															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	TIMER	0x0	RW	**Timer Period**
	Number of CLK_SRC_ADC cycles per timer event.			

22.5.6 IADC_STATUS - Status

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0x0						0x0			0x0	0x0				0x0	0x0	0x0					0x0	0x0		0x0		0x0	0x0		0x0	0x0
Access		R						R			R	R				R	R	R					R	R		R		R	R		R	R
Name		ADCWARM						SYNCBUSY			MASKREQWRITEPENDING	SINGLEWRITEPENDING				TIMERACTIVE	SCANFIFOFLUSHING	SINGLEFIFOFLUSHING					SCANFIFODV	SINGLEFIFODV		CONVERTING		SCANQUEUEPENDING	SCANQEN		SINGLEQUEUEPENDING	SINGLEQEN

Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
30	ADCWARM	0x0	R	**ADCWARM**
	The ADC analog front end and reference require a delay before converting when coming from a powered down or standby state. This status bit indicates that the analog front end and reference are ready.			
29:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24	SYNCBUSY	0x0	R	**SYNCBUSY**
	Indicates synchronization ongoing			
23:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	MASKREQWRITE-PENDING	0x0	R	**MASKREQ write pending**
	A write to MASKREQ is pending. The ADC converts using a local working mask register, and only transfers MASKREQ to the local working version when the SCAN queue is not converting.			
20	SINGLEWRITEPEND-ING	0x0	R	**SINGLE write pending**
	The SINGLE register write is pending. The ADC converts using a local working version of the SINGLE register, and only transfer SINGLE to the local working version when the SINGLE queue is not being converted.			
19:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	TIMERACTIVE	0x0	R	**Timer Active**
	The local timer is running.			
15	SCANFIFOFLUSHING	0x0	R	**The Scan FIFO is flushing**
	A scan data FIFO flush operation is in progress.			
14	SINGLEFIFOFLUSHING	0x0	R	**The Single FIFO is flushing**
	A single data FIFO flush operation is in progress.			

Bit	Name	Reset	Access	Description
13:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	SCANFIFODV	0x0	R	**SCANFIFO Data Valid**
	At least one result in the scan FIFO is ready to read.			
8	SINGLEFIFODV	0x0	R	**SINGLEFIFO Data Valid**
	At least one result in the single FIFO is ready to read.			
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	CONVERTING	0x0	R	**Converting**
	The ADC is warmed up and in the process of performing a conversion.			
5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	SCANQUEUEPENDING	0x0	R	**Scan Queue Pending**
	The Scan queue has been triggered and is waiting to start conversion.			
3	SCANQEN	0x0	R	**Scan Queued Enabled**
	The Scan queue is enabled.			
2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	SINGLEQUEUEPEND-ING	0x0	R	**Single Queue Pending**
	The Single queue has been triggered and is waiting to start conversion. When tailgating is used, SINGLEQUEUEPEND-ING will remain high until the a scan operation has completed.			
0	SINGLEQEN	0x0	R	**Single Queue Enabled**
	The Single queue is enabled.			

22.5.7 IADC_MASKREQ - Mask Request

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	RW															
Name																	MASKREQ															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	MASKREQ	0x0	RW	**Scan Queue Mask Request**
	Allows software to specify which entries in the Scan table should be converted. For example MASKREQ = 0x8014 means that scan table entries 15, 4, and 2 will be converted. The other entries will not be converted.			

22.5.8 IADC_STMASK - Scan Table Mask

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	R															
Name																	STMASK															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	STMASK	0x0	R	**Scan Table Mask**
	This is the active / working copy of the MASKREQ register that the ADC uses. It will only be updated at the end of a scan sequence or when no scan is in progress.			

22.5.9 IADC_CMPTHR - Digital Window Comparator Threshold

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset								0x0									0x0															
Access								RW									RW															
Name								ADGT									ADLT															

Bit	Name	Reset	Access	Description
31:16	ADGT	0x0	RW	**ADC Greater Than or Equal to Threshold**
	Compare threshold value for greater-than or equal to comparison. ADGT should be specified in a left-justified, 16-bit format regardless of the FIFO ALIGNMENT setting. Comparisons with 12-bit formats will ignore the 4 LSBs of the ADGT value. Comparisons with 20-bit formats will ignore the 4 LSBs of the 20-bit result. Unipolar or bipolar mode is considered in the comparison. When ADGT is greater than ADLT, the comparison is true if the result is either greater than ADGT or less than ADLT, but false if the result falls between the values.			
15:0	ADLT	0x0	RW	**ADC Less Than or Equal to Threshold**
	Compare threshold value for less-than or equal to comparison. ADLT should be specified in a left-justified, 16-bit format regardless of the FIFO ALIGNMENT setting. Comparisons with 12-bit formats will ignore the 4 LSBs of the ADLT value. Comparisons with 20-bit formats will ignore the 4 LSBs of the 20-bit result. Unipolar or bipolar mode is considered in the comparison. When ADGT is greater than ADLT, the comparison is true if the result is either greater than ADGT or less than ADLT, but false if the result falls between the values.			

22.5.10 IADC_IF - Interrupt Flags

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0												0x0	0x0	0x0	0x0			0x0	0x0			0x0	0x0	0x0				0x0	0x0	0x0	0x0
Access	RW												RW	RW	RW	RW			RW	RW			RW	RW	RW				RW	RW	RW	RW
Name	EM23ABORTERROR												SCANFIFOUF	SINGLEFIFOUF	SCANFIFOOF	SINGLEFIFOOF			PORTALLOCERR	POLARITYERR			SINGLEDONE	SCANTABLEDONE	SCANENTRYDONE				SCANCMP	SINGLECMP	SCANFIFODVL	SINGLEFIFODVL

Bit	Name	Reset	Access	Description
31	EM23ABORTERROR	0x0	RW	**EM2/3 Abort Error**
	The system entered EM2 or EM3 during a conversion with an unsupported clock. Conversion results may be corrupted.			
30:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19	SCANFIFOUF	0x0	RW	**Scan FIFO Underflow**
	A scan FIFO underflow has occured.			
18	SINGLEFIFOUF	0x0	RW	**Single FIFO Underflow**
	A single FIFO underflow has occured.			
17	SCANFIFOOF	0x0	RW	**Scan FIFO Overflow**
	A scan FIFO overflow has occured.			
16	SINGLEFIFOOF	0x0	RW	**Single FIFO Overflow**
	A single FIFO overflow has occured.			
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	PORTALLOCERR	0x0	RW	**Port Allocation Error**
	A pin was selected on a port which has not been allocated to the IADC in GPIO control.			
12	POLARITYERR	0x0	RW	**Polarity Error**
	Either two even channels or two odd channels were programmed into the channel mux selection. The ADC result will be set to 0xFFFF.			
11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	SINGLEDONE	0x0	RW	**Single Conversion Done**
	A single conversion has completed.			
8	SCANTABLEDONE	0x0	RW	**Scan Table Done**
	A scan sequence completed. Set at the end of a scan sequence after all valid entries of the scan table have completed.			
7	SCANENTRYDONE	0x0	RW	**Scan Entry Done**
	A scan table conversion completed. Set at the completion of each valid entry of the scan table.			

Bit	Name	Reset	Access	Description
6:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	SCANCMP	0x0	RW	**Scan Result Window Compare**
	Scan digital compare window tripped.			
2	SINGLECMP	0x0	RW	**Single Result Window Compare**
	Single digital compare window tripped.			
1	SCANFIFODVL	0x0	RW	**Scan FIFO Data Valid Level**
	A minimum of (DVL+1) entries are ready to be read from the Scan FIFO.			
0	SINGLEFIFODVL	0x0	RW	**Single FIFO Data Valid Level**
	A minimum of (DVL+1) entries are ready to be read from the Single FIFO.			

22.5.11 IADC_IEN - Interrupt Enable

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0												0x0	0x0	0x0	0x0			0x0	0x0			0x0	0x0	0x0				0x0	0x0	0x0	0x0
Access	RW												RW	RW	RW	RW			RW	RW			RW	RW	RW				RW	RW	RW	RW
Name	EM23ABORTERROR												SCANFIFOUF	SINGLEFIFOUF	SCANFIFOOF	SINGLEFIFOOF			PORTALLOCERR	POLARITYERR			SINGLEDONE	SCANTABLEDONE	SCANENTRYDONE				SCANCMP	SINGLECMP	SCANFIFODVL	SINGLEFIFODVL

Bit	Name	Reset	Access	Description
31	EM23ABORTERROR	0x0	RW	**EM2/3 Abort Error Enable**
	EM2/3 Abort Error Enable			
30:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19	SCANFIFOUF	0x0	RW	**Scan FIFO Underflow Enable**
	Scan FIFO underflow interrupt enable			
18	SINGLEFIFOUF	0x0	RW	**Single FIFO Underflow Enable**
	Single FIFO underflow interrupt enable			
17	SCANFIFOOF	0x0	RW	**Scan FIFO Overflow Enable**
	Scan FIFO overflow interrupt enable			
16	SINGLEFIFOOF	0x0	RW	**Single FIFO Overflow Enable**
	Single FIFO overflow interrupt enable			
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	PORTALLOCERR	0x0	RW	**Port Allocation Error Enable**
	Port Allocation Error Enable			
12	POLARITYERR	0x0	RW	**Polarity Error Enable**
	Polarity Error Enable			
11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	SINGLEDONE	0x0	RW	**Single Conversion Done Enable**
	Single Conversion Done interrupt enable			
8	SCANTABLEDONE	0x0	RW	**Scan Table Done Enable**
	Scan Table Done interrupt enable			
7	SCANENTRYDONE	0x0	RW	**Scan Entry Done Enable**
	Scan Entry Done interrupt enable			

Bit	Name	Reset	Access	Description
6:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	SCANCMP	0x0	RW	**Scan Result Window Compare Enable**
	Scan Result Window Compare Enable			
2	SINGLECMP	0x0	RW	**Single Result Window Compare Enable**
	Single Result Window Compare Enable			
1	SCANFIFODVL	0x0	RW	**Scan FIFO Data Valid Level Enable**
	Scan FIFO Data Valid Level interrupt enable			
0	SINGLEFIFODVL	0x0	RW	**Single FIFO Data Valid Level Enable**
	Single FIFO Data Valid Level interrupt enable			

22.5.12 IADC_TRIGGER - Trigger

Offset	Bit Position																															
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x0				0x0			0x0					0x0			0x0	
Access																RW				RW			RW					RW			RW	
Name																SINGLETAILGATE				SINGLETRIGACTION			SINGLETRIGSEL					SCANTRIGACTION			SCANTRIGSEL	

Bit	Name	Reset	Access	Description
31:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	SINGLETAILGATE	0x0	RW	**Single Tailgate Enable**
	Enables tailgating.			
	Value	Mode		Description
	0	TAILGATEOFF		The single queue is ready to start warming up and converting once the trigger had been detected.
	1	TAILGATEON		After the single queue's trigger is detected, it must wait until the end of a scan operation before the Single queue can be converted.
15:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	SINGLETRIGACTION	0x0	RW	**Single Trigger Action**
	Selects the trigger action for the single queue.			
	Value	Mode		Description
	0	ONCE		For TRIGSEL=IMMEDIATE, converts the single queue once and disables queue. For TRIGSEL = TIMER, PRSCLKGRP, PRSPOS, PRSNEG, converts the single queue once per trigger.
	1	CONTINUOUS		Converts the single queue, then checks for a pending scan queue before converting the single queue again continuously. The queues are first come first serve. If both queues are continuous, the IADC alternates between them.
11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10:8	SINGLETRIGSEL	0x0	RW	**Single Trigger Select**
	Selects the trigger source for the single queue.			
	Value	Mode		Description

Bit	Name	Reset	Access	Description
	0	IMMEDIATE		Immediate triggering. The single queue will be disabled once the conversion is complete, unless TRIGGERACTION is set to continuous.
	1	TIMER		Triggers when the local timer count reaches zero.
	2	PRSCLKGRP		Triggers on PRS1 from a timer module that is using the same clock group as the ADC and has been programmed to use the same clock source as the ADC. The prescale may be different between the ADC and the timer module.
	3	PRSPOS		Triggers on asynchronous PRS1 positive edge. Requires PRS1 to go low for 3 ADC_CLKs before another positive edge can be detected. Generates an additional delay of 1 to 2 CLK_SRC_ADC cycles for synchronization.
	4	PRSNEG		Triggers on asynchronous PRS1 negative edge. Requires PRS1 to go high for 3 ADC_CLKs before another negative edge can be detected. Generates an additional delay of 1 to 2 CLK_SRC_ADC cycles for synchronization. PRSNEG should only be used when the trigger source is from a module that remains powered during EM23. For modules (ie: TIMER) that power down during EM23, PRSPOS should be used for an asynchronous trigger, and PRSCLKGRP should be used for a synchronous trigger.
7:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	SCANTRIGACTION	0x0	RW	**Scan Trigger Action** Selects the trigger action for the scan queue.
	Value	Mode		Description
	0	ONCE		For TRIGSEL=IMMEDIATE, goes through the scan table once and disables queue. For TRIGSEL = TIMER, PRSCLKGRP, PRSPOS, PRSNEG, goes through the scan table once per trigger.
	1	CONTINUOUS		Goes through the scan table, converts each entry with a mask bit set, and puts it back into the scan queue to repeat again continuously. The queues are first come first serve. If both queues are triggered, the single queue will get to convert after each scan table completes. The scan queue will get to convert after each single conversion completes.
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	SCANTRIGSEL	0x0	RW	**Scan Trigger Select** Selects the trigger source for the scan queue.
	Value	Mode		Description
	0	IMMEDIATE		Immediate triggering. The scan queue will be disabled once all conversions in the scan table are complete, unless TRIGGERACTION is set to continuous.
	1	TIMER		Triggers when the local timer count reaches zero.
	2	PRSCLKGRP		Triggers on PRS0 from a timer module that is using the same clock group as the ADC and has been programmed to use the same clock source as the ADC. The prescale may be different between the ADC and the timer module.

Bit	Name	Reset	Access	Description
	3	PRSPOS		Triggers on asynchronous PRS0 positive edge. Requires PRS0 to go low for 3 ADC_CLKs before another positive edge can be detected. Generates an additional delay of 1 to 2 CLK_SRC_ADC cycles for synchronization.
	4	PRSNEG		Triggers on asynchronous PRS0 negative edge. Requires PRS0 to go high for 3 ADC_CLKs before another negative edge can be detected. Generates an additional delay of 1 to 2 CLK_SRC_ADC cycles for synchronization. PRSNEG should only be used when the trigger source is from a module that remains powered during EM23. For modules (ie: TIMER) that power down during EM23, PRSPOS should be used for an asynchronous trigger, and PRSCLKGRP should be used for a synchronous trigger.
	5	LESENSE		Triggers on LESENSE convert request. When using the LESENSE for the SCAN Table, only one entry is converted per LESENSE convert request.

22.5.13 IADC_CFGx - Configuration

Offset	Bit Position																													
0x048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset			0x0						0x0					0x0				0x2							0x3			0x0			0x0	
Access			RW						RW					RW				RW							RW			RW			RW	
Name			TWOSCOMPL						DIGAVG					REFSEL				ANALOGGAIN							OSRHA			OSRHS			ADCMODE	

Bit	Name	Reset	Access	Description
31:30	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
29:28	TWOSCOMPL	0x0	RW	**Two's Complement**
	Selects output word polarity.			

Value	Mode	Description
0	AUTO	Automatic: Single ended measurements are reported as unipolar and differential measurements are reported as bipolar.
1	FORCEUNIPOLAR	Force all measurements to result in unipolar output. Negative differential numbers will saturate to 0.
2	FORCEBIPOLAR	Force all measurements to result in bipolar output. Single ended measurements are half the range, but allow for small negative measurements.

Bit	Name	Reset	Access	Description
27:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:21	DIGAVG	0x0	RW	**Digital Averaging**
	Number of output words to convert and average.			

Value	Mode	Description
0	AVG1	Collect one output word (no digital averaging).
1	AVG2	Collect and average 2 digital output words.
2	AVG4	Collect and average 4 digital output words.
3	AVG8	Collect and average 8 digital output words.
4	AVG16	Collect and average 16 digital output words.

Bit	Name	Reset	Access	Description
20:19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18:16	REFSEL	0x0	RW	**Reference Select**
	Selects voltage reference.			

Value	Mode	Description
0	VBGR	Internal 1.21 V reference.

Bit	Name	Reset	Access	Description
	1	VREF		External Reference. (Calibrated for 1.25V nominal.)
	2	VREF2P5		External Reference. Supports 2.5V in high accuracy mode.
	3	VDDX		AVDD (unbuffered)
	4	VDDX0P8BUF		AVDD (buffered) * 0.8
15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14:12	ANALOGGAIN	0x2	RW	**Analog Gain**
	Sets analog front end gain.			
	Value	Mode		Description
	1	ANAGAIN0P5		Analog gain of 0.5x.
	2	ANAGAIN1		Analog gain of 1x.
	3	ANAGAIN2		Analog gain of 2x.
	4	ANAGAIN3		Analog gain of 3x.
	5	ANAGAIN4		Analog gain of 4x.
11:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:5	OSRHA	0x3	RW	**High Accuracy OSR**
	Over sampling ratio for high accuracy conversions.			
	Value	Mode		Description
	0	HIACC16		High accuracy over sampling of 16x.
	1	HIACC32		High accuracy over sampling of 32x.
	2	HIACC64		High accuracy over sampling of 64x.
	3	HIACC92		High accuracy over sampling of 92x.
	4	HIACC128		High accuracy over sampling of 128x.
	5	HIACC256		High accuracy over sampling of 256x.
4:2	OSRHS	0x0	RW	**High Speed OSR**
	Over sampling ratio for high speed conversions.			
	Value	Mode		Description
	0	HISPD2		High speed over sampling of 2x.
	1	HISPD4		High speed over sampling of 4x.
	2	HISPD8		High speed over sampling of 8x.
	3	HISPD16		High speed over sampling of 16x.
	4	HISPD32		HIgh speed over sampling of 32x.
	5	HISPD64		High speed over sampling of 64x.
1:0	ADCMODE	0x0	RW	**ADC Mode**
	Selects ADC conversion mode.			

Bit	Name	Reset	Access	Description

Value	Mode	Description
0	NORMAL	High speed mode with a maximum ADC_CLK of 10 MHz.
1	HIGHSPEED	Double high speed mode with a maximum ADC_CLK of 20 MHz. Power consumption is boosted to allow faster conversions.
2	HIGHACCURACY	High accuracy mode with maximum ADC_CLK of 5 MHz.

22.5.14 IADC_SCALEx - Scaling

Offset	Bit Position																														
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1	0x0													0x2C000																
Access	RW	RW													RW																
Name	GAIN3MSB	GAIN13LSB													OFFSET																

Bit	Name	Reset	Access	Description
31	GAIN3MSB	0x1	RW	**Gain 3 MSBs**
	3 MSBs of the 16-bit gain value (0=011 or 0.75; 1=1xx or 1.00). Example {GAIN3MSB, GAIN13LSB} = {100, 0_1001_0000_0000} = 1.07031x. Example {GAIN3MSB, GAIN13LSB} = {011, 0_0000_1010_0010} = 0.75494x.			

Value	Mode	Description
0	GAIN011	Upper 3 bits of gain = 011 (0.75x)
1	GAIN100	Upper 3 bits of gain = 100 (1.00x)

30:18	GAIN13LSB	0x0	RW	**Gain 13 LSBs**
	13 LSBs of the 16-bit gain value.			
17:0	OFFSET	0x2C000	RW	**Offset**
	Offset			

22.5.15 IADC_SCHEDx - Scheduling

Offset	Bit Position																															
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0					
Access																											RW					
Name																											PRESCALE					

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:0	PRESCALE	0x0	RW	**Prescale** Second level prescaler - divides the CLK_SRC_ADC by (PRESCALE + 1) to generate ADC_CLK. PRESCALE=0 should only be used with HSCLKRATE=0. (See text.)

22.5.16 IADC_SINGLEFIFOCFG - Single FIFO Configuration

Offset	Bit Position																															
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0		0x3			0x0		0x0	
Access																								RW		RW			RW		RW	
Name																								DMAWUFIFOSINGLE		DVL			SHOWID		ALIGNMENT	

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	DMAWUFIFOSINGLE	0x0	RW	**Single FIFO DMA wakeup.**

Enables single FIFO to wake DMA in EM2 or EM3.

Value	Mode	Description
0	DISABLED	While in EM2 or EM3, the DMA controller will not be requested.
1	ENABLED	While in EM2 or EM3, the DMA controller will be requested when the single FIFO reaches its Data Valid Level. [DVL must be set to 0 (VALID1).]

Bit	Name	Reset	Access	Description
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	DVL	0x3	RW	**Data Valid Level**

Data valid level before requesting DMA transfer. If the number of words in the FIFO reaches or exceeds DVL+1, DMA requests will be generated.

Value	Mode	Description
0	VALID1	When 1 entry in the single FIFO is valid, set the SINGLEFIFODVL interrupt and request DMA.
1	VALID2	When 2 entries in the single FIFO are valid, set the SINGLEFIFODVL interrupt and request DMA.
2	VALID3	When 3 entries in the single FIFO are valid, set the SINGLEFIFODVL interrupt and request DMA.
3	VALID4	When 4 entries in the single FIFO are valid, set the SINGLEFIFODVL interrupt and request DMA.
4	VALID5	When 5 entries in the single FIFO are valid, set the SINGLEFIFODVL interrupt and request DMA.
5	VALID6	When 6 entries in the single FIFO are valid, set the SINGLEFIFODVL interrupt and request DMA.
6	VALID7	When 7 entries in the single FIFO are valid, set the SINGLEFIFODVL interrupt and request DMA.

Bit	Name	Reset	Access	Description
	7	VALID8		When 8 entries in the single FIFO are valid, set the SINGLEFIFODVL interrupt and request DMA.
3	SHOWID	0x0	RW	**Show ID**
	ID of 0x20 will be applied in the output words.			
2:0	ALIGNMENT	0x0	RW	**Alignment**
	Alignment of output data written into FIFO.			

Value	Mode	Description
0	RIGHT12	ID[7:0], SIGN_EXT, DATA[11:0]
1	RIGHT16	ID[7:0], SIGN_EXT, DATA[15:0]
2	RIGHT20	ID[7:0], SIGN_EXT, DATA[19:0]
3	LEFT12	DATA[11:0], 000000000000, ID[7:0]
4	LEFT16	DATA[15:0], 00000000, ID[7:0]
5	LEFT20	DATA[19:0], 0000, ID[7:0]

22.5.17 IADC_SINGLEFIFODATA - Single FIFO DATA

Offset	Bit Position																													
0x074	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R(r)																															
Name	DATA																															

Bit	Name	Reset	Access	Description
31:0	DATA	0x0	R(r)	**Single FIFO Read Data**
	Reads and pops the oldest value from the single FIFO.			

22.5.18 IADC_SINGLEFIFOSTAT - Single FIFO Status

Offset	Bit Position																															
0x078	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													R		
Name																													FIFOREADCNT		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	FIFOREADCNT	0x0	R	**FIFO Read Count**
	Number of valid entries available to read.			

22.5.19 IADC_SINGLEDATA - Single Data

Offset	Bit Position																															
0x07C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x0																
Access																R																
Name																DATA																

Bit	Name	Reset	Access	Description
31:0	DATA	0x0	R	**Data**
	Reads the most recent data word from the single FIFO, but does not pop a value. Even if the FIFO has overflowed and stopped updating, the most recent conversion will continue to overwrite SINGLEDATA.			

22.5.20 IADC_SCANFIFOCFG - Scan FIFO Configuration

Offset	Bit Position																															
0x080	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0			0x3		0x0		0x0	
Access																								RW			RW		RW		RW	
Name																								DMAWUFIFOSCAN			DVL		SHOWID		ALIGNMENT	

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	DMAWUFIFOSCAN	0x0	RW	**Scan FIFO DMA Wakeup**

Enables scan FIFO to wake DMA in EM2 or EM3.

Value	Mode	Description
0	DISABLED	While in EM2 or EM3, the DMA controller will not be requested.
1	ENABLED	While in EM2 or EM3, the DMA controller will be requested when the scan FIFO reaches its Data Valid Level. [DVL must be set to 0 (VALID1).]

7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	DVL	0x3	RW	**Data Valid Level**

Data valid level before requesting DMA transfer. If the number of words in the FIFO reaches or exceeds DVL+1, DMA requests will be generated.

Value	Mode	Description
0	VALID1	When 1 entry in the scan FIFO is valid, set the SCANFIFODVL interrupt and request DMA.
1	VALID2	When 2 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
2	VALID3	When 3 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
3	VALID4	When 4 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
4	VALID5	When 5 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
5	VALID6	When 6 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
6	VALID7	When 7 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.

Bit	Name	Reset	Access	Description
	7	VALID8		When 8 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
3	SHOWID	0x0	RW	**Show ID**
	Enable ID in output words.			
2:0	ALIGNMENT	0x0	RW	**Alignment**
	Alignment of output data written into FIFO.			

Value	Mode	Description
0	RIGHT12	ID[7:0], SIGN_EXT, DATA[11:0]
1	RIGHT16	ID[7:0], SIGN_EXT, DATA[15:0]
2	RIGHT20	ID[7:0], SIGN_EXT, DATA[19:0]
3	LEFT12	DATA[11:0], 000000000000, ID[7:0]
4	LEFT16	DATA[15:0], 00000000, ID[7:0]
5	LEFT20	DATA[19:0], 0000, ID[7:0]

22.5.21 IADC_SCANFIFODATA - Scan FIFO Read Data

Offset	Bit Position																													
0x084	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R(r)																															
Name	DATA																															

Bit	Name	Reset	Access	Description
31:0	DATA	0x0	R(r)	**Data**
	Reads and pops the oldest value from the scan FIFO.			

22.5.22 IADC_SCANFIFOSTAT - Scan FIFO Status

Offset	Bit Position																															
0x088	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													R		
Name																													FIFOREADCNT		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	FIFOREADCNT	0x0	R	**FIFO Read Count**
	Number of valid entries available to read.			

22.5.23 IADC_SCANDATA - Scan Data

Offset	Bit Position																															
0x08C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x0																
Access																R																
Name																DATA																

Bit	Name	Reset	Access	Description
31:0	DATA	0x0	R	**Data**
	Reads the most recent data word from the scan FIFO, but does not pop a value. Even if the FIFO has overflowed and stopped updating, the most recent conversion will continue to overwrite SCANDATA.			

22.5.24 IADC_SINGLE - Single Queue Port Selection

Offset	Bit Position																													
0x098	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0	0x0	0x0				0x0				0x0				0x0			
Access															RW	RW	RW				RW				RW				RW			
Name															CMP	CFG	PORTPOS				PINPOS				PORTNEG				PINNEG			

Bit	Name	Reset	Access	Description
31:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	CMP	0x0	RW	**Comparison Enable**
	Enable digital window comparison for this entry.			
16	CFG	0x0	RW	**Configuration Group Select**
	Select which configuration group (CFGx, SCALEx, SCHEDx registers) is used with this entry.			

Value	Mode	Description
0	CONFIG0	Use configuration group 0
1	CONFIG1	Use configuration group 1

Bit	Name	Reset	Access	Description
15:12	PORTPOS	0x0	RW	**Positive Port Select**
	Port (A, B, C, D) or special signal assigned to the positve input of the ADC			

Value	Mode	Description
0	GND	Ground
1	SUPPLY	Supply Pin - Select specific supply using PINPOS
2	DAC0	Direct connection to DAC0_CH0
4	PADANA0	Direct connection to AIN0 input pin
5	PADANA2	Direct connection to AIN2 input pin
8	PORTA	Port A - Select pin number using PINPOS
9	PORTB	Port B - Select pin number using PINPOS
10	PORTC	Port C - Select pin number using PINPOS
11	PORTD	Port D - Select pin number using PINPOS

Bit	Name	Reset	Access	Description
11:8	PINPOS	0x0	RW	**Positive Pin Select**
	Pin number for the positive input of the ADC.			
7:4	PORTNEG	0x0	RW	**Negative Port Select**
	Port (A, B, C, D) or special signal assigned to the negative input of the ADC			

Value	Mode	Description
0	GND	Ground (single-ended)

Bit	Name	Reset	Access	Description
	2	DAC1		Direct connection to DAC0_CH1
	4	PADANA1		Direct connection to AIN1 input pin
	5	PADANA3		Direct connection to AIN3 input pin
	8	PORTA		Port A - Select pin number using PINNEG
	9	PORTB		Port B - Select pin number using PINNEG
	10	PORTC		Port C - Select pin number using PINNEG
	11	PORTD		Port D - Select pin number using PINNEG
3:0	PINNEG	0x0	RW	**Negative Pin Select**
	Pin number for the negative input of the ADC.			

22.5.25 IADC_SCANx - SCAN Entry

Offset	Bit Position																													
0x0A0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0	0x0	0x0				0x0				0x0				0x0			
Access															RW	RW	RW				RW				RW				RW			
Name															CMP	CFG	PORTPOS				PINPOS				PORTNEG				PINNEG			

Bit	Name	Reset	Access	Description
31:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	CMP	0x0	RW	**Comparison Enable**
	Enable digital window comparison for this entry.			
16	CFG	0x0	RW	**Configuration Group Select**
	Select which configuration group (CFGx, SCALEx, SCHEDx registers) is used with this entry.			

Value	Mode	Description
0	CONFIG0	Use configuration group 0
1	CONFIG1	Use configuration group 1

15:12	PORTPOS	0x0	RW	**Positive Port Select**
	Port (A, B, C, D) or special signal assigned to the positve input of the ADC			

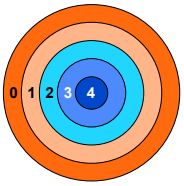
Value	Mode	Description
0	GND	Ground
1	SUPPLY	Supply Pin - Select specific supply using PINPOS
2	DAC0	Direct connection to DAC0_CH0
4	PADANA0	Direct connection to AIN0 input pin
5	PADANA2	Direct connection to AIN2 input pin
8	PORTA	Port A - Select pin number using PINPOS
9	PORTB	Port B - Select pin number using PINPOS
10	PORTC	Port C - Select pin number using PINPOS
11	PORTD	Port D - Select pin number using PINPOS

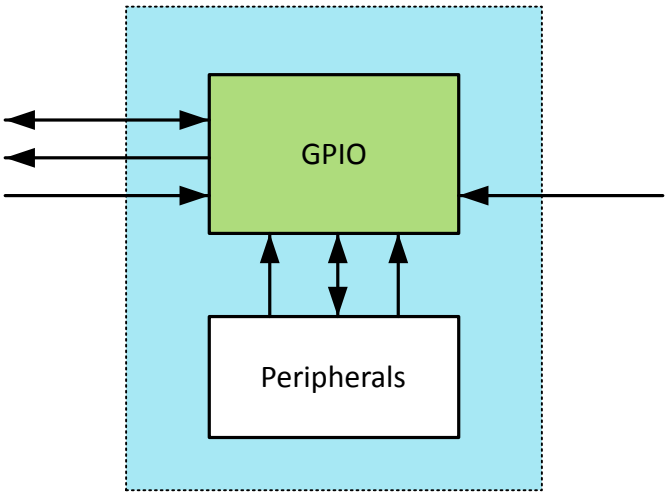
11:8	PINPOS	0x0	RW	**Positive Pin Select**
	Pin number for the positive input of the ADC.			
7:4	PORTNEG	0x0	RW	**Negative Port Select**
	Port (A, B, C, D) or special signal assigned to the negative input of the ADC			

Value	Mode	Description
0	GND	Ground (single-ended)

Bit	Name	Reset	Access	Description
	2	DAC1		Direct connection to DAC0_CH1
	4	PADANA1		Direct connection to AIN1 input pin
	5	PADANA3		Direct connection to AIN3 input pin
	8	PORTA		Port A - Select pin number using PINNEG
	9	PORTB		Port B - Select pin number using PINNEG
	10	PORTC		Port C - Select pin number using PINNEG
	11	PORTD		Port D - Select pin number using PINNEG
3:0	PINNEG	0x0	RW	**Negative Pin Select**
	Pin number for the negative input of the ADC.			

23. GPIO - General Purpose Input/Output





Quick Facts

What?

The General Purpose Input/Output (GPIO) is used for pin configurations as well as routing for peripheral pin connections.

Why?

Easy to use and highly configurable input/output pins are important to fit many communication protocols as well as minimizing software control overhead. Flexible routing of peripheral functions helps to ease PCB layout.

How?

Each pin on the device can be individually configured as either an input or an output with several different drive modes. Also, individual bit manipulation registers minimizes control overhead. Peripheral connections to pins can be routed to pins as desired solving congestion and contention issues that may arise with limited routing flexibility. Fully asynchronous interrupts can also be generated from any pin.

23.1 Introduction

In the EFM32PG28 devices the General Purpose Input/Output (GPIO) pins are organized into ports with up to 16 pins each. These GPIO pins can be individually configured as either an output or input. More advanced configurations like open-drain, open-source, and glitch filtering can be configured for each individual GPIO pin. Peripheral resources, like Timer PWM outputs or USART RX/TX can be routed to the GPIO pins as desired by the user. Finally, the input value of a pin can be routed through the Peripheral Reflex System to other peripherals or used to trigger an external interrupt.

23.2 Features

- Individual configuration for each pin
  - Tristate (reset state)
  - Push-pull
  - Open-drain
  - Pull-up resistor
  - Pull-down resistor
  - Programable Slewrate Control
- EM4 IO pin retention
  - Output enable
  - Output value
  - Pull enable
  - Pull direction
- EM4 wake-up on selected GPIO pins
- Glitch suppression input filter
- Extremely flexible analog and digital resource routing
- Toggle register for output data
- Dedicated data input register (read-only)
- Interrupts
  - Two independent interrupt vectors
  - All GPIO pins are selectable as interrupts in EM0 and EM1
  - All PA and PB GPIO pins are also selectable as interrupts down to EM2 and EM3
  - All EM4 wake-up pins are also available as interrupts in EM0/1/2/3
  - Separate enable, status, set and clear registers
  - Asynchronous sensing
  - Rising, falling or both edges
- Peripheral Reflex System producer
  - All GPIO pins are selectable

23.3 Functional Description

An overview of the GPIO module is shown in Figure 23.1 Pin Configuration on page 776. The GPIO pins are grouped into 16-pin ports. Each individual GPIO pin is called Pxn where x indicates the port (A, B, C ...) and n indicates the pin number (0,1,....,15). Fewer than 16 pins may be available on some ports depending on the total number of I/O pins on the package. After a reset, both input and output are disabled for all pins on the device, except for the Serial Wire Debug pins.

To use a pin, the Mode Register (GPIO_Px_MODEL/GPIO_Px_MODEH) must be configured for the pin to make it an input or output. These registers can also do more advanced configuration, which is covered in 23.3.1 Pin Configuration. When the port is configured as an input or an output, the Data In Register (GPIO_Px_DIN) can be used to read the level of each pin in the port (bit n in the register is connected to pin n on the port). When configured as an output, the value of the Data Out Register (GPIO_Px_DOUT) will be driven to the pin.

The DOUT value can be changed in 4 different ways:

- Writing to the GPIO_Px_DOUT register
- Writing the SET address of the GPIO_Px_DOUT register sets the DOUT bits
- Writing the CLEAR address of the GPIO_Px_DOUT register clears the DOUT bits
- Writing the GPIO_Px_DOUTTGL register toggles the corresponding DOUT bits

Reading the GPIO_Px_DOUT register will return its contents. Reading the GPIO_Px_DOUTTGL register will return 0.

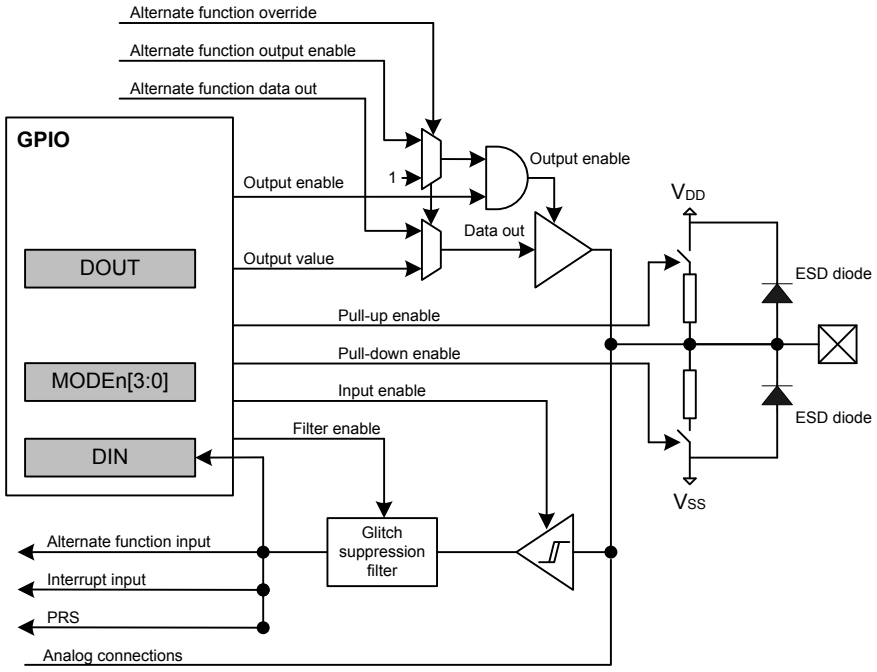


Figure 23.1. Pin Configuration

23.3.1 Pin Configuration

In addition to setting the pins as either outputs or inputs, the GPIO_Px_MODEL and GPIO_Px_MODEH registers can be used for more advanced configurations. GPIO_Px_MODEL contains 8 bit fields named MODEn (n=0,1,..7) which control pins 0-7, while GPIO_Px_MODEH contains 8 bit fields named MODEn (n=8,9,..15) which control pins 8-15. In some modes GPIO_Px_DOUT is also used for extra configurations like pull-up/down and glitch suppression filter enable. Table 23.1 Pin Configuration on page 777 shows the available configurations.

Table 23.1. Pin Configuration

MODEn	Input	Output	DOUT	Pull-down	Pull-up	Alt Port Ctrl	Input Filter	Description
DISABLED	Disabled	Disabled	0					Input disabled
			1		On			Input disabled with pull-up
INPUT	Enabled if not DINDIS		0					Input enabled
			1				On	Input enabled with filter
INPUTPULL			0	On				Input enabled with pull-down
			1		On			Input enabled with pull-up
INPUTPULLFILTER			0	On			On	Input enabled with pull-down and filter
			1		On		On	Input enabled with pull-up and filter
PUSHPULL		Push-pull	x					Push-pull
PUSHPULLALT			x			On		Push-pull with alternate port control values
WIREDOR		Open Source (Wired-OR)	x					Open-source
WIREDORPULLDOWN			x	On				Open-source with pull-down
WIREDAND		Open Drain (Wired-AND)	x					Open-drain
WIREDANDFILTER			x				On	Open-drain with filter
WIREDANDPULLUP			x		On			Open-drain with pull-up
WIREDANDPULLUPFILTER			x		On		On	Open-drain with pull-up and filter
WIREDANDALT			x			On		Open-drain with alternate port control values
WIREDANDALTFILTER			x			On	On	Open-drain with alternate port control values and filter
WIREDANDALTPULLUP			x		On	On		Open-drain with alternate port control values and pull-up
WIREDANDALTPULLUPFILTER			x		On	On	On	Open-drain with alternate port control values, pull-up and filter

MODEn determines which mode the pin is in at a given time. Setting MODEn to DISABLED disables the pin, reducing power consumption to a minimum. When the output driver and input driver are disabled, the pin can be used as a connection for an analog module. An input is enabled by setting MODEn to any value other than DISABLED while DINDIS for the given port is cleared. Set DINDIS to disable

the input of a GPIO port. The pull-up, pull-down and glitch filter function can optionally be applied to the input, see Figure 23.2 Tristated Output with Optional Pull-up or Pull-down on page 778.

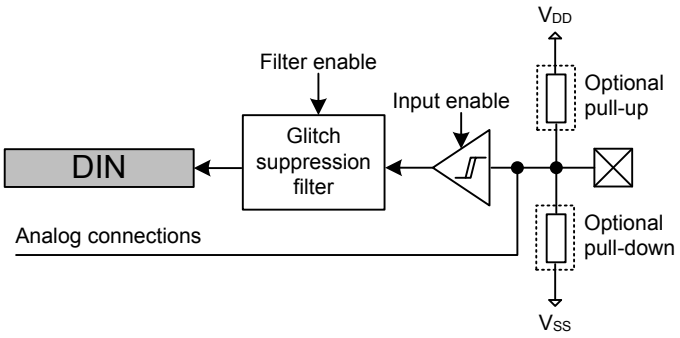


Figure 23.2. Tristated Output with Optional Pull-up or Pull-down

When MODEn is PUSHPULL or PUSHPULLALT, the pin operates in push-pull mode. In this mode, the pin can have alternate port control values and can be driven either high or low, dependent on the value of GPIO_Px_DOUT. The push-pull configuration is shown in Figure 23.3 Push-Pull Configuration on page 778.

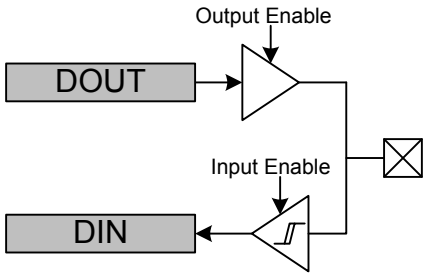


Figure 23.3. Push-Pull Configuration

When MODEn is WIREDOR or WIREDORPULLDOWN, the pin operates in open-source mode (with a pull-down resistor for WIREDORPULLDOWN). When driving a high value in open-source mode, the pull-down is disconnected to save power.

When the mode is prefixed with WIREDAND, the pin operates in open-drain mode as shown in Figure 23.4 Open-drain on page 778. In open-drain mode, the pin can have an input filter, a pull-up, alternate port control values or any combination of these. When driving a low value in open-drain mode, the pull-up is disconnected to save power.

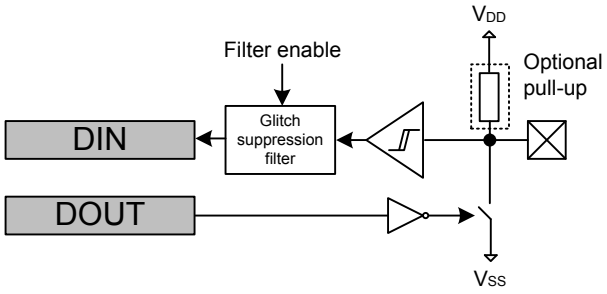


Figure 23.4. Open-drain

23.3.2 Alternate Port Control

The Alternate Port Control allows for additional flexibility of port level settings. A user may setup two different port configurations (normal and alternate modes) and select which is applied on a pin by pin bases. For example you may configure half of port A to use the slowest slew rate while the other half uses a faster slew rate.

Alternate port control is enabled when MODEn is set to any of the ALT enumerated modes (i.e.. PUSHPULLALT). When MODEn is an alternate mode, the pin uses the alternate port control values specified in the DINDISALT and SLEWRATEALT fields in GPIO_Px_CTRL. In all other modes, the port control values are used from the DINDIS and SLEWRATE fields in GPIO_Px_CTRL.

23.3.3 Slew Rate

The slewrate can be applied to pins on a port-by-port basis. The slew rate applied to pins configured using normal MODEn settings can be controlled using the SLEWRATE fields in GPIO_Px_CTRL. The slewrate applied to pins configured using the alternate MODEn settings can be controlled using the SLEWRATEALT field.

The lowest slew rate setting has limited drive strength. That is the current is limited to about 1 mA. This setting provides slow switching and limited drive. A slew rate setting of 1 provides the slowest switching with full drive capability. The maximum recommended setting for most digital I/O is 6. A slew rate setting of 7 should only be used for high-speed clock signals, above 10 MHz. A setting of 7 should not be used on more than one pin per port. Please refer to the datasheet for GPIO rise and fall times.

23.3.4 Input Disable

The pin inputs can be disabled on a port-by-port basis. The input of pins configured using the normal MODEn settings can be disabled by setting DINDIS in GPIO_Px_CTRL. The input of pins configured using the alternate MODEn settings can be disabled by setting DINDISALT.

23.3.5 Configuration Lock

The GPIO configuration registers (GPIO_Px_CTRL, PIO_Px_MODEL, GPIO_xBUSALLOC, GPIO_EXTIPSELL, GPIO_EXTIPINSEL, GPIO_x_yROUTE, and GPIO_xROUTEEN) can be locked by writing any value other than 0xA534 to GPIO_LOCK. Writing the value 0xA534 to the GPIOx_LOCK register unlocks the configuration registers.

23.3.6 EM2 Functionality

While all GPIO pins retain their state in EM2, only pins on port A and B remain fully functional in EM2. Digital peripherals which are active in EM2 must have their resources routed to pins on port A or B to function correctly in EM2. Analog peripherals may use any GPIO pin while in EM2 provided that the ABUS was configured prior to entering EM2. However, analog peripherals that are configured to scan multiple pins while in EM2 (such as the ADC) dynamically reconfigure the ABUS while in EM2 and thus must use only pins on port A and B.

23.3.7 EM4 Functionality

By default GPIO pins revert back to their reset state when EM4 is entered. The GPIO pins can be configured to retain the settings for output enable, output value, pull enable, and pull direction while in EM4.

EM4 GPIO retention is controlled with the EM4IORETMODE field in the EMU_EM4CTRL register:

- Setting EM4IORETMODE to EM4EXIT will cause GPIO retention to persist while in EM4. GPIO state will be reset during wakeup.
- Setting EM4IORETMODE to SWUNLATCH will cause the GPIO retention to persist through EM4 and wakeup, until the EM4UNLATCH bit is written by software. When using SWUNLATCH, the GPIO register values are still reset on wakeup. To ensure the GPIO state does not change, software must re-write the GPIO registers before setting EM4UNLATCH and ending EM4 GPIO retention. Note that the GPIO state cannot be retained through an EM4 wakeup due to a reset (e.g., pin reset or POR reset) - only non-reset methods of EM4 wakeup are supported (e.g., EM4WU IRQ or BURTC IRQ).

See the EMU chapter for additional documentation on EM4IORETMODE and the EM4UNLATCH bit.

23.3.8 EM4 Wakeup

It is possible to trigger a wake-up from EM4 using any of the selectable EM4WU GPIO pins. The wake-up request can be triggered through the pins by enabling the corresponding bit in the GPIO_EM4WUEN register. When EM4 wake-up is enabled for the pin, the input filter is enabled during EM4. This is done to avoid false wake-up caused by glitches. In addition, the polarity of the EM4 wake-up request can be selected using the GPIO_EM4WUPOL register.

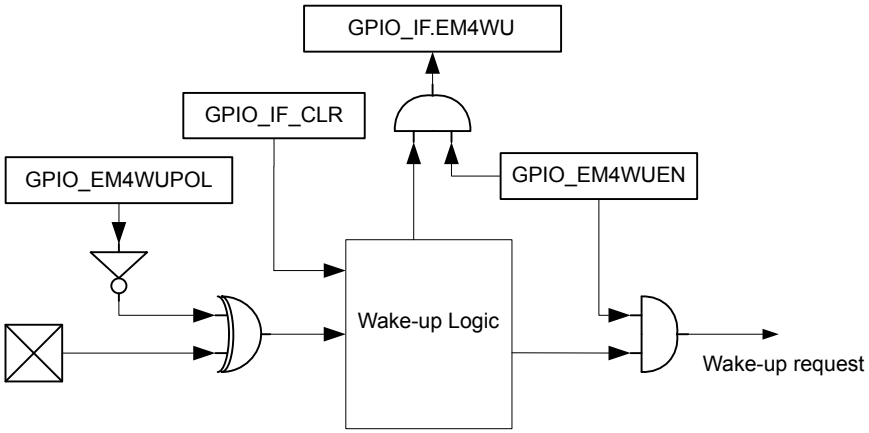


Figure 23.5. EM4 Wake-up Logic

The pins used for EM4 wake-up must be configured as inputs with glitch filters using the GPIO_Px_MODEL register. If the input is disabled and the wakeup polarity is low, a false wakeup will occur when entering EM4. If the input is enabled, the glitch filtered is disabled, and the polarity is set low, a glitch will occur when going into EM4 that will cause an immediate wake-up. Before going down to EM4, it is important to clear the wake-up logic by setting the GPIO_IF_CLR bits, which clear the wake-up logic, including the GPIO_IF register. It is possible to determine which pin caused the EM4WU by reading the GPIO_IF register.

Each EM4WU signal is connected to a fixed pin. Refer to the Alternate Function Table in the device Datasheet for the location of each EM4 wakeup signal.

23.3.9 Debug Connections

23.3.9.1 JTAG Debug Connection

The JTAG Debug Port is a fixed location resource connected directly to specific GPIO pins. Refer to the Alternate Function Table in the device Datasheet for the location of the JTAG signals. By default TMS, TCK, TDO, and TDI pin connections are enabled with internal pull up, pull down, no pull, and pull up resistors, respectively. It is possible to disable these pin connections (and disable the pull resistors) by setting the SWDIOTMSPEN, SWCLKTCKPEN, TDOPEN, and TDIPEN bits in GPIO_DEBUGROUTEPEN to 0.

23.3.9.2 Serial Wire Debug Connection

The SW Debug Port is a fixed location resource connected directly to specific GPIO pins. Refer to the Alternate Function Table in the device Datasheet for the location of the SW Debug port signals. The SWDIO and SWCLK pin connections are enabled by default with internal pull up and pull down resistors, respectively. It is possible to disable these pin connections (and disable the pull resistors) by setting the SWDIOTMSPEN and SWCLKTCKPEN bits in GPIO_DEBUGROUTEPEN to 0.

The Serial Wire Viewer pin, SWV, can be enabled by setting the SWVPEN bit in GPIO_TRACEROUTEPEN.

Note: The SWV pin is not affected by debug lock, so the SWV pin should not be enabled for production devices.

23.3.9.3 Disabling Debug Connections

When the debug pins are disabled, the device can no longer be accessed by a debugger. A reset will set the debug pins back to their enabled default state. The GPIO_DBGROUTEPEN register can only be updated when the debugger is disconnected from the system. Any attempts to modify GPIO_DBGROUTEPEN when the debugger is connected will not occur. If you do disable the debug pins, make sure you have at least a 3 second timeout at the start of your program code before you disable the debug pins. This way the debugger will have time to connect to the device after a reset and before the pins are disabled.

23.3.9.4 ETM Trace Connections

The device includes ETM trace pins. The trace clock can be enabled by setting the TRACECLKPEN bit-field in GPIO_TRACEROUTEPEN. The data pin(s) can be enabled individually by setting TRACEDATAxPEN in GPIO_TRACEROUTEPEN. The trace pins are fixed location resources connected to specific pins. Refer to the Alternate Function Table in the device Datasheet for the location of the ETM trace port signals.

23.3.10 Interrupt Generation

23.3.10.1 Standard Interrupt Generation

The GPIO can generate an interrupt from any edge of the input of any GPIO pin on the device. The standard interrupts have asynchronous sense capability, enabling wake-up from energy modes as low as EM3, see Figure 23.6 Pin n Interrupt Generation on page 781.

Note: In EM2 and EM3, only signals on Port A and Port B are available as standard interrupts. Standard interrutps are available to all pins in EM0 and EM1.

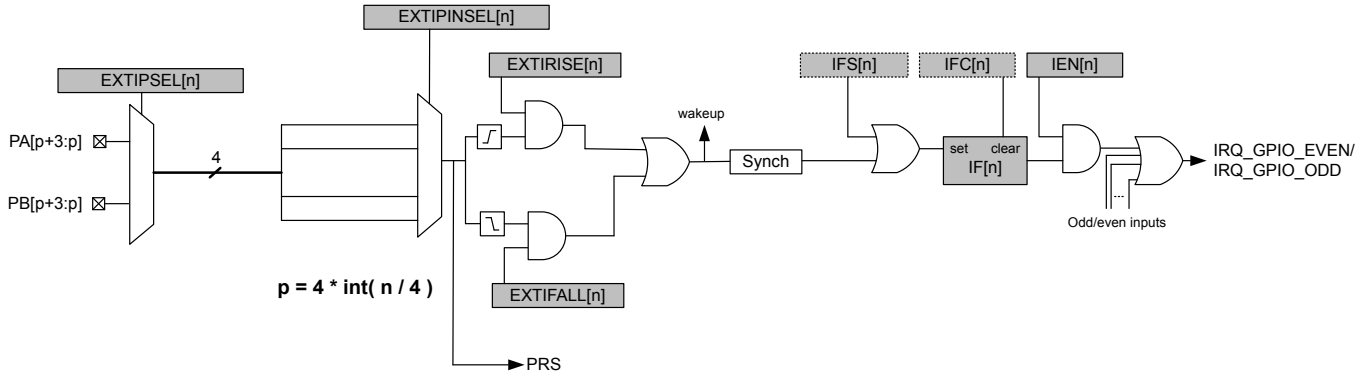


Figure 23.6. Pin n Interrupt Generation

The standard external pin interrupts are numbered starting with 0. Each interrupt has a corresponding enable bit in the GPIO_IEN register and an interrupt flag bit in the GPIO_IF register. Each interrupt may be used with one of four possible pins on any available port. First select the desired port for each interrupt using the corresponding EXTIPSELx field in the GPIO_EXTIPSELL register. (Some devices with many pins may also have a GPIO_EXTIPSELH register.)

Each interrupt can be mapped to one of four possible pins on the selected port. External interrupts EXTI0 through EXTI3 may be mapped to pins 0,1,2, or 3 on any available port. External interrupts EXTI4 through EXTI7 may be mapped to pins 4,5,6 or 7 on any available port.

Note: Note that while the EXTIEN field in the GPIO_IEN register has 15 bits, the number of useful bits is limited by the number of pins available in the widest port. If the widest port is 8 bits wide, only the first 8 external interrupts are useful.

The selected pin for each interrupt is the base plus the offset. The base for EXTI0 through EXTI3 is 0, while the base for interrupts EXTI4 through EXTI7 is 4. The base may be calculated by taking the interrupt number, dividing by four, then using only the integer portion of the quotient. (BASE = Integer(N/4)

The offset is selected using the corresponding field in the GPIO_EXTIPINSELL register, (Some devices with many pins may also have a GPIO_EXTIPINSELH register.) Subtract the base from the desired pin number to get the offset. For example, to map EXTI5 to pin 7 of PORTA, the base is 4 and the offset will be 3.

The GPIO_EXTIRISE[n] and GPIO_EXTIFALL[n] registers enable sensing of rising and falling edges. By setting the EXT[n] bit in GPIO_IEN, a high interrupt flag n, will trigger one of two interrupt lines. The even interrupt line is triggered by any enabled even numbered interrupt flag index, while the odd interrupt line is triggered by odd flag indexes. The interrupt flags can be set and cleared by software when writing the GPIO_IF_SET and GPIO_IF_CLR register locations. Since the external interrupts are asynchronous, they are sensitive to noise. To increase noise tolerance, the MODEx field(s) in the GPIO_Px_MODEL register, should be set to include glitch filtering for pins that have external interrupts enabled.

23.3.10.2 Interrupt Generation on EM4WU Pins

In addition to being an EM4 wake source, any of the dedicated EM4WU (EM4 wake-up) signals on PA, PB, PC or PD may be used to generate edge-sensitive interrupts in EM0, EM1, EM2, and EM3.

In order to enable an EM4WU pin as an interrupt, set the EM4WUIENn field in the GPIO_IEN register and the EM4WUENn field in the EM4WUEN register. The EM4WUPOLn field in the GPIO_EM4WUPOL register is used to set the desired polarity for the interrupt (0 for a falling edge, and 1 for a rising edge).

Upon an interrupt occurring, the corresponding EM4WU index in the GPIO_IF register will be set along with the odd or even interrupt line depending on the index inside of GPIO_IF. For example, by setting the. EM4WU8 in GPIO_EM4WUPOL and EM4WU[8] in GPIO_IEN, the interrupt flag EM4WU[8] in GPIO_IF will be triggered by a rising edge on pin EM4WU8 and a interrupt request will be sent on IRQ_GPIO_EVEN.

The wake-up granularity of the EM4WU interrupts is based on the settings of the EM4WU field in the GPIO_IEN register and the EM4WUEN field in the GPIO_EM4WUEN register (see Table 23.2 EM4WU Interrupt Energy Mode Wakeup on page 782).

Table 23.2. EM4WU Interrupt Energy Mode Wakeup

EM4WUIENn in GPIO_IEN	EM4WUENn in GPIO_EM4WUEN	Energy Mode Wakeup	Interrupt
x	0	No Wake	No Interrupt
0	1	Wake from EM4	No Interrupt
1	1	Wake from EM1, EM2,EM3, or EM4	Interrupt from EM0, EM1, EM2, or EM3

For example, to configure the device to wake up and generate an interrupt when PD02 (EM4WU9) sees a falling edge:

1. Set bit 9 of EM4WUEN in the GPIO_EM4WUEN register to '1'. This enables the asynchronous wake logic.
2. Set bit 9 of EM4WUIEN in the GPIO_IEN register to '1'. This enables routing of the wake signal to the GPIO_ODD IRQ.
3. Clear bit 9 of EM4WUPOL in the GPIO_EM4WUPOL register to '0'. This indicates that the interrupt should occur when a falling edge is detected at the pin.
4. Enable the GPIO.ODD IRQ. The ODD interrupt is used because the bit index of EM4WUIF in GPIO_IF is odd.

23.3.11 Output to PRS

All pins within a group of four(0-3,4-7,8-11,12-15) from all ports are grouped together to form one PRS producer which outputs to the PRS. The pin from which the output should be taken is selected in the same fashion as the edge interrupts.

PRS output is not affected by the interrupt edge detection logic or gated by the IEN bits. See 23.3.10 Interrupt Generation for an illustration of where the PRS output signal is generated.

23.3.12 Peripheral Resource Routing

Most peripherals have resources that need to be connected to GPIO pins to function. For example, the I2C has SDA and SCL which need to be connected to pins for the I2C to communicate with other ICs. Resources come in three types. Fixed resources are hard-wired to a pin and can only be accessed in that location. For example the LFXO LFXTAL_I and LFXTAL_O resources are only available on one pin each. Digital route-able resources are connected to pins through the 23.3.12.1 Digital Bus (DBUS) which allows for extremely flexible resource placement. Analog route-able resources are connected to pins though the 23.3.12.2 Analog Bus (ABUS) which provides extremely flexible resource placement.

The locations of fixed resources and the limitations of ABUS and DBUS on each device can be found in the device data sheet.

23.3.12.1 Digital Bus (DBUS)

The Digital Bus (DBUS) is an any-to-any switch matrix between peripheral resources and GPIO pins as shown in Figure 23.7 Digital Bus Interconnect on page 783. There are two DBUSes on the EFM32PG28 - DBUSAB serves ports A and B, while DBUSCD serves ports C and D. Not all peripherals have access to both DBUSes.

To connect a resource to a pin, first select the desired PORT and PIN in the GPIO_x_yROUTE register, where x is the peripheral name and y is the resource name. The PORT field is encoded as PA = 0, PB = 1, PC = 2, etc. Once the pin is selected, the resource must be enabled by setting its enable bit in the appropriate GPIO_x_ROUTEEN register. For example, to route the SDA resource of I2C0 to PB03, set PORT to 0x1 and PIN to 0x3 in GPIO_I2C0_SDAROUTE. Then set the GPIO_I2C0_ROUTEN.SDAPEN bit.

Any pin connected to a digital resource should be properly configured for that resource (refer to 23.3.1 Pin Configuration). For example, an I2C SDA should be configured as open-drain, a USART (or EUSART) TX should be configured as push-pull, and a USART (or EUSART) RX should be configured as an input.

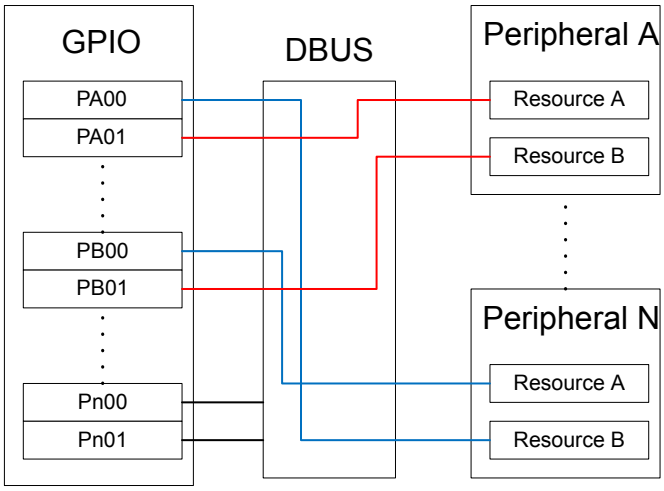


Figure 23.7. Digital Bus Interconnect

23.3.12.2 Analog Bus (ABUS)

Analog peripherals may be connected to any pins on port A, B, C, or D via the Analog Bus. There are three analog buses on the EFM32PG28: one dedicated to Port A (ABUSA), one dedicated to port B (ABUSB), and one that serves both ports C and D (ABUSCD). The specific pin and port selection for analog resources are configured in the analog peripherals. Refer to the respective analog peripheral chapter for this information. However, the GPIO block must be configured to grant the peripheral access to an ABUS before any connection can be made.

Note: The analog signals on ABUSes will be voltage limited by the lowest supply voltage of IOVDD and AVDD.

Up to two analog peripherals may be given access to an ABUS at any one time and the even/odd pins of each bus are configured independently. This means that a single bus may have up to four different analog peripherals connected to it: two on the even pins and two on the odd pins. The GPIO_ABUSxALLOC register, where x is the port, determines which peripherals have access to the bus. To grant a peripheral access to the bus even pins select it in either the EVEN0 or EVEN1 field. To grant a peripheral access to the bus odd pins select it in either the ODD0 or ODD1 fields.

When a differential connection is being used, positive inputs are restricted to the EVEN pins and negative inputs are restricted to the ODD pins. When a single ended connection is being used, the positive input is avaliable on all pins.

Peripherals may be given access to as many buses as desired. For example the ADC may be given access to ABUSA, ABUSB, and ABUSCD allowing it to select any pin on ports A-D. If two peripherals select the same port and pin the ABUS will make both connections simultaneously, effectively connecting the two peripherals together.

Any pin connected to an analog resource should be configured to input DISABLED as described in 23.3.1 Pin Configuration

The process for configuring an analog peripheral to access a pin through the ABUS is as follows:

- Configure the desired analog port pins to input DISABLED mode in the corresponding GPIO_PORTx_MODEL/H register.
- Configure the corresponding GPIO_xBUSALLOC field to grant access to the desired peripheral on the desired ABUS.
- Configure the analog peripheral to select the desired port and channel as described in the peripheral chapter.

23.3.12.3 Pin Function Tables

This section details the functions and GPIO pins available on the most fully-featured devices in the EFM32PG28 family. Availability of GPIO and signals varies. Refer to the device datasheet for specific peripheral and GPIO availability. Fixed-pin peripheral resources are shown in Table 23.3 GPIO Alternate Function Table on page 785, ABUS routing options are listed in Table 23.4 ABUS Routing Table on page 787, and DBUS routing options are listed in Table 23.5 DBUS Routing Table on page 788

Table 23.3. GPIO Alternate Function Table

GPIO	Alternate Functions
PA00	LCD.SEG8
PA01	GPIO.SWCLK
	LCD.SEG9
PA02	GPIO.SWDIO
PA03	GPIO.SWV
	GPIO.TDO
	GPIO.TRACEDATA0
	LESENSE.EN_0
PA04	GPIO.TDI
	GPIO.TRACECLK
	LCD.SEG10
	LESENSE.EN_1
PA05	GPIO.TRACEDATA1
	GPIO.EM4WU0
	LCD.SEG11
	LESENSE.EN_2
PA06	GPIO.TRACEDATA2
	LCD.LCD_CP
PA07	GPIO.TRACEDATA3
	LCD.SEG12
PA08	LCD.SEG13
PA09	LCD.SEG20
PA10	LCD.SEG21
PA11	LCD.SEG22
PA12	LCD.SEG23
PA13	LCD.COM4
	LCD.SEG24
PA14	LCD.COM5
	LCD.SEG25
PB00	LCD.SEG14
	VDAC0.CH0_MAIN_OUT

GPIO	Alternate Functions
PB01	GPIO.EM4WU3
	LCD.SEG15
	VDAC0.CH1_MAIN_OUT
PB02	LCD.SEG16
PB03	GPIO.EM4WU4
	LCD.SEG17
PB04	LCD.COM6
	LCD.SEG26
PB05	LCD.COM7
	LCD.SEG27
PC00	GPIO.EM4WU6
	LCD.SEG0
PC01	GPIO.EFP_TX_SDA
	LCD.SEG1
PC02	GPIO.EFP_TX_SCL
	LCD.SEG2
PC03	LCD.SEG3
PC04	LCD.SEG4
PC05	GPIO.EFP_INT
	GPIO.EM4WU7
	LCD.SEG5
PC06	LCD.SEG6
PC07	GPIO.EM4WU8
	LCD.SEG7
PC08	LCD.SEG18
PC09	LCD.SEG19
PC11	GPIO.THMSW_EN
	GPIO.THMSW_HALFSWITCH
PD00	LFXO.LFXTAL_O
PD01	LFXO.LFXTAL_I
	LFXO.LF_EXTCLK
PD02	GPIO.EM4WU9
	LCD.COM0
PD03	LCD.COM1
PD04	LCD.COM2
PD05	GPIO.EM4WU10
	LCD.COM3

Table 23.4. ABUS Routing Table

Peripheral	Signal	PA		PB		PC		PD	
		EVEN	ODD	EVEN	ODD	EVEN	ODD	EVEN	ODD
ACMP0	ana_neg	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	ana_pos	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ACMP1	ana_neg	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	ana_pos	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
IADC0	ana_neg	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	ana_pos	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VDAC0	ch0_abus_out	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	ch1_abus_out	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 23.5. DBUS Routing Table

Peripheral.Resource	PORT			
	PA	PB	PC	PD
ACMP0.DIGOUT	Available	Available	Available	Available
ACMP1.DIGOUT	Available	Available	Available	Available
CMU.CLKIN0			Available	Available
CMU.CLKOUT0			Available	Available
CMU.CLKOUT1			Available	Available
CMU.CLKOUT2	Available	Available		
EUSART0.CS	Available	Available		
EUSART0.CTS	Available	Available		
EUSART0.RTS	Available	Available		
EUSART0.RX	Available	Available		
EUSART0.SCLK	Available	Available		
EUSART0.TX	Available	Available		
EUSART1.CS	Available	Available	Available	Available
EUSART1.CTS	Available	Available	Available	Available
EUSART1.RTS	Available	Available	Available	Available
EUSART1.RX	Available	Available	Available	Available
EUSART1.SCLK	Available	Available	Available	Available
EUSART1.TX	Available	Available	Available	Available
EUSART2.CS			Available	Available
EUSART2.CTS			Available	Available
EUSART2.RTS			Available	Available
EUSART2.RX			Available	Available
EUSART2.SCLK			Available	Available
EUSART2.TX			Available	Available
I2C0.SCL	Available	Available	Available	Available
I2C0.SDA	Available	Available	Available	Available
I2C1.SCL			Available	Available
I2C1.SDA			Available	Available
KEYSCAN.COL_OUT_0	Available	Available	Available	Available
KEYSCAN.COL_OUT_1	Available	Available	Available	Available
KEYSCAN.COL_OUT_2	Available	Available	Available	Available
KEYSCAN.COL_OUT_3	Available	Available	Available	Available
KEYSCAN.COL_OUT_4	Available	Available	Available	Available
KEYSCAN.COL_OUT_5	Available	Available	Available	Available

Peripheral.Resource	PORT			
	PA	PB	PC	PD
KEYSCAN.COL_OUT_6	Available	Available	Available	Available
KEYSCAN.COL_OUT_7	Available	Available	Available	Available
KEYSCAN.ROW_SENSE_0	Available	Available		
KEYSCAN.ROW_SENSE_1	Available	Available		
KEYSCAN.ROW_SENSE_2	Available	Available		
KEYSCAN.ROW_SENSE_3	Available	Available		
KEYSCAN.ROW_SENSE_4	Available	Available		
KEYSCAN.ROW_SENSE_5	Available	Available		
LESENSE.CH0OUT	Available	Available		
LESENSE.CH10OUT	Available	Available		
LESENSE.CH11OUT	Available	Available		
LESENSE.CH12OUT	Available	Available		
LESENSE.CH13OUT	Available	Available		
LESENSE.CH14OUT	Available	Available		
LESENSE.CH15OUT	Available	Available		
LESENSE.CH1OUT	Available	Available		
LESENSE.CH2OUT	Available	Available		
LESENSE.CH3OUT	Available	Available		
LESENSE.CH4OUT	Available	Available		
LESENSE.CH5OUT	Available	Available		
LESENSE.CH6OUT	Available	Available		
LESENSE.CH7OUT	Available	Available		
LESENSE.CH8OUT	Available	Available		
LESENSE.CH9OUT	Available	Available		
LETIMER0.OUT0	Available	Available		
LETIMER0.OUT1	Available	Available		
PCNT0.S0IN	Available	Available		
PCNT0.S1IN	Available	Available		
PRS.ASYNCH0	Available	Available		
PRS.ASYNCH1	Available	Available		
PRS.ASYNCH10			Available	Available
PRS.ASYNCH11			Available	Available
PRS.ASYNCH2	Available	Available		
PRS.ASYNCH3	Available	Available		
PRS.ASYNCH4	Available	Available		
PRS.ASYNCH5	Available	Available		

Peripheral.Resource	PORT			
	PA	PB	PC	PD
PRS.ASYNCH6			Available	Available
PRS.ASYNCH7			Available	Available
PRS.ASYNCH8			Available	Available
PRS.ASYNCH9			Available	Available
PRS.SYNCH0	Available	Available	Available	Available
PRS.SYNCH1	Available	Available	Available	Available
PRS.SYNCH2	Available	Available	Available	Available
PRS.SYNCH3	Available	Available	Available	Available
TIMER0.CC0	Available	Available	Available	Available
TIMER0.CC1	Available	Available	Available	Available
TIMER0.CC2	Available	Available	Available	Available
TIMER0.CDTI0	Available	Available	Available	Available
TIMER0.CDTI1	Available	Available	Available	Available
TIMER0.CDTI2	Available	Available	Available	Available
TIMER1.CC0	Available	Available	Available	Available
TIMER1.CC1	Available	Available	Available	Available
TIMER1.CC2	Available	Available	Available	Available
TIMER1.CDTI0	Available	Available	Available	Available
TIMER1.CDTI1	Available	Available	Available	Available
TIMER1.CDTI2	Available	Available	Available	Available
TIMER2.CC0	Available	Available		
TIMER2.CC1	Available	Available		
TIMER2.CC2	Available	Available		
TIMER2.CDTI0	Available	Available		
TIMER2.CDTI1	Available	Available		
TIMER2.CDTI2	Available	Available		
TIMER3.CC0			Available	Available
TIMER3.CC1			Available	Available
TIMER3.CC2			Available	Available
TIMER3.CDTI0			Available	Available
TIMER3.CDTI1			Available	Available
TIMER3.CDTI2			Available	Available
TIMER4.CC0	Available	Available		
TIMER4.CC1	Available	Available		
TIMER4.CC2	Available	Available		
TIMER4.CDTI0	Available	Available		

Peripheral.Resource	PORT			
	PA	PB	PC	PD
TIMER4.CDTI1	Available	Available		
TIMER4.CDTI2	Available	Available		
USART0.CLK	Available	Available	Available	Available
USART0.CS	Available	Available	Available	Available
USART0.CTS	Available	Available	Available	Available
USART0.RTS	Available	Available	Available	Available
USART0.RX	Available	Available	Available	Available
USART0.TX	Available	Available	Available	Available

23.4 Synchronization

To avoid metastability in synchronous logic connected to the pins, all inputs are synchronized with double flip-flops. The flip-flops for the input data run on the selected APB clock for the GPIO module (PCLK). Consequently, when a pin changes state, the change will propagate to GPIO_Px_DIN after two 2 PCLK cycles. Synchronization (also running on the PCLK) is also added for interrupt input. To save power when the external interrupts are not used, the synchronization flip-flops for these can be turned off by clearing the EXTINT field in the GPIO_IEN register.

23.5 GPIO Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	GPIO_IPVERSION	R	Main
0x030	GPIO_PORTA_CTRL	RW	Port Control
0x034	GPIO_PORTA_MODEL	RW	Mode Low
0x03C	GPIO_PORTA_MODEH	RW	Mode High
0x040	GPIO_PORTA_DOUT	RW	Data Out
0x044	GPIO_PORTA_DIN	RH	Data in
0x060	GPIO_PORTB_CTRL	RW	Port Control
0x064	GPIO_PORTB_MODEL	RW	Mode Low
0x070	GPIO_PORTB_DOUT	RW	Data Out
0x074	GPIO_PORTB_DIN	RH	Data in
0x090	GPIO_PORTC_CTRL	RW	Port Control
0x094	GPIO_PORTC_MODEL	RW	Mode Low
0x09C	GPIO_PORTC_MODEH	RW	Mode High
0x0A0	GPIO_PORTC_DOUT	RW	Data Out
0x0A4	GPIO_PORTC_DIN	RH	Data in
0x0C0	GPIO_PORTD_CTRL	RW	Port Control
0x0C4	GPIO_PORTD_MODEL	RW	Mode Low
0x0CC	GPIO_PORTD_MODEH	RW	Mode High
0x0D0	GPIO_PORTD_DOUT	RW	Data Out
0x0D4	GPIO_PORTD_DIN	RH	Data in
0x300	GPIO_LOCK	W	Lock Register
0x310	GPIO_GPIOLOCKSTATUS	RH	Lock Status
0x320	GPIO_ABUSALLOC	RW	A Bus Allocation
0x324	GPIO_BBUSALLOC	RW	B Bus Allocation
0x328	GPIO_CDBUSALLOC	RW	CD Bus Allocation
0x400	GPIO_EXTIPSELL	RW	External Interrupt Port Select Low
0x404	GPIO_EXTIPSELH	RW	External Interrupt Port Select High
0x408	GPIO_EXTIPINSELL	RW	External Interrupt Pin Select Low
0x40C	GPIO_EXTIPINSELH	RW	External Interrupt Pin Select High
0x410	GPIO_EXTIRISE	RW	External Interrupt Rising Edge Trigger
0x414	GPIO_EXTIFALL	RW	External Interrupt Falling Edge Trigger
0x420	GPIO_IF	RWH INTFLAG	Interrupt Flag
0x424	GPIO_IEN	RW	Interrupt Enable
0x42C	GPIO_EM4WUEN	RW	EM4 Wakeup Enable
0x430	GPIO_EM4WUPOL	RW	EM4 Wakeup Polarity

Offset	Name	Type	Description
0x440	GPIO_DBGROUTEPEN	RW	Debugger Route Pin Enable
0x444	GPIO_TRACEROUTEPEN	RW	Trace Route Pin Enable
0x460	GPIO_LCDSEG	RW	LCD Segment Enable
0x470	GPIO_LCDCOM	RW	LCD Common Enable
0x480	GPIO_ACMP0_ROUTEEN	RW	ACMP0 Pin Enable
0x484	GPIO_ACMP0_ACMPOUT-ROUTE	RW	ACMPOUT Port/Pin Select
0x48C	GPIO_ACMP1_ROUTEEN	RW	ACMP1 Pin Enable
0x490	GPIO_ACMP1_ACMPOUT-ROUTE	RW	ACMPOUT Port/Pin Select
0x498	GPIO_CMU_ROUTEEN	RW	CMU Pin Enable
0x49C	GPIO_CMU_CLKIN0ROUTE	RW	CLKIN0 Port/Pin Select
0x4A0	GPIO_CMU_CLKOUT0ROUTE	RW	CLKOUT0 Port/Pin Select
0x4A4	GPIO_CMU_CLKOUT1ROUTE	RW	CLKOUT1 Port/Pin Select
0x4A8	GPIO_CMU_CLKOUT2ROUTE	RW	CLKOUT2 Port/Pin Select
0x4C4	GPIO_EUSART0_ROUTEEN	RW	EUSART0 Pin Enable
0x4C8	GPIO_EUSART0_CSROUTE	RW	CS Port/Pin Select
0x4CC	GPIO_EUSART0_CTSROUTE	RW	CTS Port/Pin Select
0x4D0	GPIO_EUSART0_RTSROUTE	RW	RTS Port/Pin Select
0x4D4	GPIO_EUSART0_RXROUTE	RW	RX Port/Pin Select
0x4D8	GPIO_EUSART0_SCLKROUTE	RW	SCLK Port/Pin Select
0x4DC	GPIO_EUSART0_TXROUTE	RW	TX Port/Pin Select
0x4E4	GPIO_EUSART1_ROUTEEN	RW	EUSART1 Pin Enable
0x4E8	GPIO_EUSART1_CSROUTE	RW	CS Port/Pin Select
0x4EC	GPIO_EUSART1_CTSROUTE	RW	CTS Port/Pin Select
0x4F0	GPIO_EUSART1_RTSROUTE	RW	RTS Port/Pin Select
0x4F4	GPIO_EUSART1_RXROUTE	RW	RX Port/Pin Select
0x4F8	GPIO_EUSART1_SCLKROUTE	RW	SCLK Port/Pin Select
0x4FC	GPIO_EUSART1_TXROUTE	RW	TX Port/Pin Select
0x504	GPIO_EUSART2_ROUTEEN	RW	EUSART2 Pin Enable
0x508	GPIO_EUSART2_CSROUTE	RW	CS Port/Pin Select
0x50C	GPIO_EUSART2_CTSROUTE	RW	CTS Port/Pin Select
0x510	GPIO_EUSART2_RTSROUTE	RW	RTS Port/Pin Select
0x514	GPIO_EUSART2_RXROUTE	RW	RX Port/Pin Select
0x518	GPIO_EUSART2_SCLKROUTE	RW	SCLK Port/Pin Select
0x51C	GPIO_EUSART2_TXROUTE	RW	TX Port/Pin Select
0x538	GPIO_I2C0_ROUTEEN	RW	I2C0 Pin Enable
0x53C	GPIO_I2C0_SCLROUTE	RW	SCL Port/Pin Select

Offset	Name	Type	Description
0x540	GPIO_I2C0_SDAROUTE	RW	SDA Port/Pin Select
0x548	GPIO_I2C1_ROUTEEN	RW	I2C1 Pin Enable
0x54C	GPIO_I2C1_SCLROUTE	RW	SCL Port/Pin Select
0x550	GPIO_I2C1_SDAROUTE	RW	SDA Port/Pin Select
0x558	GPIO_KEYSCAN_ROUTEEN	RW	KEYSCAN Pin Enable
0x55C	GPIO_KEYSCAN_COL-OUT0ROUTE	RW	COLOUT0 Port/Pin Select
0x560	GPIO_KEYSCAN_COL-OUT1ROUTE	RW	COLOUT1 Port/Pin Select
0x564	GPIO_KEYSCAN_COL-OUT2ROUTE	RW	COLOUT2 Port/Pin Select
0x568	GPIO_KEYSCAN_COL-OUT3ROUTE	RW	COLOUT3 Port/Pin Select
0x56C	GPIO_KEYSCAN_COL-OUT4ROUTE	RW	COLOUT4 Port/Pin Select
0x570	GPIO_KEYSCAN_COL-OUT5ROUTE	RW	COLOUT5 Port/Pin Select
0x574	GPIO_KEYSCAN_COL-OUT6ROUTE	RW	COLOUT6 Port/Pin Select
0x578	GPIO_KEYSCAN_COL-OUT7ROUTE	RW	COLOUT7 Port/Pin Select
0x57C	GPIO_KEYSCAN_ROW-SENSE0ROUTE	RW	ROWSENSE0 Port/Pin Select
0x580	GPIO_KEYSCAN_ROW-SENSE1ROUTE	RW	ROWSENSE1 Port/Pin Select
0x584	GPIO_KEYSCAN_ROW-SENSE2ROUTE	RW	ROWSENSE2 Port/Pin Select
0x588	GPIO_KEYSCAN_ROW-SENSE3ROUTE	RW	ROWSENSE3 Port/Pin Select
0x58C	GPIO_KEYSCAN_ROW-SENSE4ROUTE	RW	ROWSENSE4 Port/Pin Select
0x590	GPIO_KEYSCAN_ROW-SENSE5ROUTE	RW	ROWSENSE5 Port/Pin Select
0x598	GPIO_LESENSE_ROUTEEN	RW	LESENSE Pin Enable
0x59C	GPIO_LESENSE_CH0OUT-ROUTE	RW	CH0OUT Port/Pin Select
0x5A0	GPIO_LESENSE_CH1OUT-ROUTE	RW	CH1OUT Port/Pin Select
0x5A4	GPIO_LESENSE_CH2OUT-ROUTE	RW	CH2OUT Port/Pin Select
0x5A8	GPIO_LESENSE_CH3OUT-ROUTE	RW	CH3OUT Port/Pin Select
0x5AC	GPIO_LESENSE_CH4OUT-ROUTE	RW	CH4OUT Port/Pin Select

Offset	Name	Type	Description
0x5B0	GPIO_LESENSE_CH5OUT-ROUTE	RW	CH5OUT Port/Pin Select
0x5B4	GPIO_LESENSE_CH6OUT-ROUTE	RW	CH6OUT Port/Pin Select
0x5B8	GPIO_LESENSE_CH7OUT-ROUTE	RW	CH7OUT Port/Pin Select
0x5BC	GPIO_LESENSE_CH8OUT-ROUTE	RW	CH8OUT Port/Pin Select
0x5C0	GPIO_LESENSE_CH9OUT-ROUTE	RW	CH9OUT Port/Pin Select
0x5C4	GPIO_LESENSE_CH10OUT-ROUTE	RW	CH10OUT Port/Pin Select
0x5C8	GPIO_LESENSE_CH11OUT-ROUTE	RW	CH11OUT Port/Pin Select
0x5CC	GPIO_LESENSE_CH12OUT-ROUTE	RW	CH12OUT Port/Pin Select
0x5D0	GPIO_LESENSE_CH13OUT-ROUTE	RW	CH13OUT Port/Pin Select
0x5D4	GPIO_LESENSE_CH14OUT-ROUTE	RW	CH14OUT Port/Pin Select
0x5D8	GPIO_LESENSE_CH15OUT-ROUTE	RW	CH15OUT Port/Pin Select
0x5E0	GPIO_LETIMER_ROUTEEN	RW	LETIMER Pin Enable
0x5E4	GPIO_LETIMER_OUT0ROUTE	RW	OUT0 Port/Pin Select
0x5E8	GPIO_LETIMER_OUT1ROUTE	RW	OUT1 Port/Pin Select
0x5F0	GPIO_MODEM_ROUTEEN	RW	MODEM Pin Enable
0x5F4	GPIO_MODEM_ANT0ROUTE	RW	ANT0 Port/Pin Select
0x5F8	GPIO_MODEM_ANT1ROUTE	RW	ANT1 Port/Pin Select
0x5FC	GPIO_MODEM_ANTROLLO-VERROUTE	RW	ANTROLLOVER Port/Pin Select
0x600	GPIO_MO-DEM_ANTRR0ROUTE	RW	ANTRR0 Port/Pin Select
0x604	GPIO_MO-DEM_ANTRR1ROUTE	RW	ANTRR1 Port/Pin Select
0x608	GPIO_MO-DEM_ANTRR2ROUTE	RW	ANTRR2 Port/Pin Select
0x60C	GPIO_MO-DEM_ANTRR3ROUTE	RW	ANTRR3 Port/Pin Select
0x610	GPIO_MO-DEM_ANTRR4ROUTE	RW	ANTRR4 Port/Pin Select
0x614	GPIO_MO-DEM_ANTRR5ROUTE	RW	ANTRR5 Port/Pin Select
0x618	GPIO_MODEM_ANTSWEN-ROUTE	RW	ANTSWEN Port/Pin Select

Offset	Name	Type	Description
0x61C	GPIO_MODEM_ANTSWUS-ROUTE	RW	ANTSWUS Port/Pin Select
0x620	GPIO_MODEM_ANTTRIG-ROUTE	RW	ANTTRIG Port/Pin Select
0x624	GPIO_MODEM_ANTTRIGSTOP-ROUTE	RW	ANTTRIGSTOP Port/Pin Select
0x628	GPIO_MODEM_DCLKROUTE	RW	DCLK Port/Pin Select
0x62C	GPIO_MODEM_DINROUTE	RW	DIN Port/Pin Select
0x630	GPIO_MODEM_DOUTROUTE	RW	DOUT Port/Pin Select
0x63C	GPIO_PCNT0_S0INROUTE	RW	S0IN Port/Pin Select
0x640	GPIO_PCNT0_S1INROUTE	RW	S1IN Port/Pin Select
0x648	GPIO_PRS0_ROUTEEN	RW	PRS0 Pin Enable
0x64C	GPIO_PRS0_ASYNCH0ROUTE	RW	ASYNCH0 Port/Pin Select
0x650	GPIO_PRS0_ASYNCH1ROUTE	RW	ASYNCH1 Port/Pin Select
0x654	GPIO_PRS0_ASYNCH2ROUTE	RW	ASYNCH2 Port/Pin Select
0x658	GPIO_PRS0_ASYNCH3ROUTE	RW	ASYNCH3 Port/Pin Select
0x65C	GPIO_PRS0_ASYNCH4ROUTE	RW	ASYNCH4 Port/Pin Select
0x660	GPIO_PRS0_ASYNCH5ROUTE	RW	ASYNCH5 Port/Pin Select
0x664	GPIO_PRS0_ASYNCH6ROUTE	RW	ASYNCH6 Port/Pin Select
0x668	GPIO_PRS0_ASYNCH7ROUTE	RW	ASYNCH7 Port/Pin Select
0x66C	GPIO_PRS0_ASYNCH8ROUTE	RW	ASYNCH8 Port/Pin Select
0x670	GPIO_PRS0_ASYNCH9ROUTE	RW	ASYNCH9 Port/Pin Select
0x674	GPIO_PRS0_ASYNCH10ROUTE	RW	ASYNCH10 Port/Pin Select
0x678	GPIO_PRS0_ASYNCH11ROUTE	RW	ASYNCH11 Port/Pin Select
0x67C	GPIO_PRS0_SYNCH0ROUTE	RW	SYNCH0 Port/Pin Select
0x680	GPIO_PRS0_SYNCH1ROUTE	RW	SYNCH1 Port/Pin Select
0x684	GPIO_PRS0_SYNCH2ROUTE	RW	SYNCH2 Port/Pin Select
0x688	GPIO_PRS0_SYNCH3ROUTE	RW	SYNCH3 Port/Pin Select
0x6F0	GPIO_SYXO0_BUFOUTREQI-NASYNCROUTE	RW	BUFOUTREQINASYNC Port/Pin Select
0x6F8	GPIO_TIMER0_ROUTEEN	RW	TIMER0 Pin Enable
0x6FC	GPIO_TIMER0_CC0ROUTE	RW	CC0 Port/Pin Select
0x700	GPIO_TIMER0_CC1ROUTE	RW	CC1 Port/Pin Select
0x704	GPIO_TIMER0_CC2ROUTE	RW	CC2 Port/Pin Select
0x708	GPIO_TIMER0_CDTI0ROUTE	RW	CDTI0 Port/Pin Select
0x70C	GPIO_TIMER0_CDTI1ROUTE	RW	CDTI1 Port/Pin Select
0x710	GPIO_TIMER0_CDTI2ROUTE	RW	CDTI2 Port/Pin Select

Offset	Name	Type	Description
0x718	GPIO_TIMER1_ROUTEEN	RW	TIMER1 Pin Enable
0x71C	GPIO_TIMER1_CC0ROUTE	RW	CC0 Port/Pin Select
0x720	GPIO_TIMER1_CC1ROUTE	RW	CC1 Port/Pin Select
0x724	GPIO_TIMER1_CC2ROUTE	RW	CC2 Port/Pin Select
0x728	GPIO_TIMER1_CDTI0ROUTE	RW	CDTI0 Port/Pin Select
0x72C	GPIO_TIMER1_CDTI1ROUTE	RW	CDTI1 Port/Pin Select
0x730	GPIO_TIMER1_CDTI2ROUTE	RW	CDTI2 Port/Pin Select
0x738	GPIO_TIMER2_ROUTEEN	RW	TIMER2 Pin Enable
0x73C	GPIO_TIMER2_CC0ROUTE	RW	CC0 Port/Pin Select
0x740	GPIO_TIMER2_CC1ROUTE	RW	CC1 Port/Pin Select
0x744	GPIO_TIMER2_CC2ROUTE	RW	CC2 Port/Pin Select
0x748	GPIO_TIMER2_CDTI0ROUTE	RW	CDTI0 Port/Pin Select
0x74C	GPIO_TIMER2_CDTI1ROUTE	RW	CDTI1 Port/Pin Select
0x750	GPIO_TIMER2_CDTI2ROUTE	RW	CDTI2 Port/Pin Select
0x758	GPIO_TIMER3_ROUTEEN	RW	TIMER3 Pin Enable
0x75C	GPIO_TIMER3_CC0ROUTE	RW	CC0 Port/Pin Select
0x760	GPIO_TIMER3_CC1ROUTE	RW	CC1 Port/Pin Select
0x764	GPIO_TIMER3_CC2ROUTE	RW	CC2 Port/Pin Select
0x768	GPIO_TIMER3_CDTI0ROUTE	RW	CDTI0 Port/Pin Select
0x76C	GPIO_TIMER3_CDTI1ROUTE	RW	CDTI1 Port/Pin Select
0x770	GPIO_TIMER3_CDTI2ROUTE	RW	CDTI2 Port/Pin Select
0x778	GPIO_TIMER4_ROUTEEN	RW	TIMER4 Pin Enable
0x77C	GPIO_TIMER4_CC0ROUTE	RW	CC0 Port/Pin Select
0x780	GPIO_TIMER4_CC1ROUTE	RW	CC1 Port/Pin Select
0x784	GPIO_TIMER4_CC2ROUTE	RW	CC2 Port/Pin Select
0x788	GPIO_TIMER4_CDTI0ROUTE	RW	CDTI0 Port/Pin Select
0x78C	GPIO_TIMER4_CDTI1ROUTE	RW	CDTI1 Port/Pin Select
0x790	GPIO_TIMER4_CDTI2ROUTE	RW	CDTI2 Port/Pin Select
0x798	GPIO_USART0_ROUTEEN	RW	USART0 Pin Enable
0x79C	GPIO_USART0_CSROUTE	RW	CS Port/Pin Select
0x7A0	GPIO_USART0_CTSROUTE	RW	CTS Port/Pin Select
0x7A4	GPIO_USART0_RTSROUTE	RW	RTS Port/Pin Select
0x7A8	GPIO_USART0_RXROUTE	RW	RX Port/Pin Select
0x7AC	GPIO_USART0_CLKROUTE	RW	SCLK Port/Pin Select
0x7B0	GPIO_USART0_TXROUTE	RW	TX Port/Pin Select
0x1000	GPIO_IPVERSION_SET	R	Main
0x1030	GPIO_PORTA_CTRL_SET	RW	Port Control

Offset	Name	Type	Description
0x1034	GPIO_PORTA_MODEL_SET	RW	Mode Low
0x103C	GPIO_PORTA_MODEH_SET	RW	Mode High
0x1040	GPIO_PORTA_DOUT_SET	RW	Data Out
0x1044	GPIO_PORTA_DIN_SET	RH	Data in
0x1060	GPIO_PORTB_CTRL_SET	RW	Port Control
0x1064	GPIO_PORTB_MODEL_SET	RW	Mode Low
0x1070	GPIO_PORTB_DOUT_SET	RW	Data Out
0x1074	GPIO_PORTB_DIN_SET	RH	Data in
0x1090	GPIO_PORTC_CTRL_SET	RW	Port Control
0x1094	GPIO_PORTC_MODEL_SET	RW	Mode Low
0x109C	GPIO_PORTC_MODEH_SET	RW	Mode High
0x10A0	GPIO_PORTC_DOUT_SET	RW	Data Out
0x10A4	GPIO_PORTC_DIN_SET	RH	Data in
0x10C0	GPIO_PORTD_CTRL_SET	RW	Port Control
0x10C4	GPIO_PORTD_MODEL_SET	RW	Mode Low
0x10CC	GPIO_PORTD_MODEH_SET	RW	Mode High
0x10D0	GPIO_PORTD_DOUT_SET	RW	Data Out
0x10D4	GPIO_PORTD_DIN_SET	RH	Data in
0x1300	GPIO_LOCK_SET	W	Lock Register
0x1310	GPIO_GPIOLOCKSTATUS_SET	RH	Lock Status
0x1320	GPIO_ABUSALLOC_SET	RW	A Bus Allocation
0x1324	GPIO_BBUSALLOC_SET	RW	B Bus Allocation
0x1328	GPIO_CDBUSALLOC_SET	RW	CD Bus Allocation
0x1400	GPIO_EXTIPSELL_SET	RW	External Interrupt Port Select Low
0x1404	GPIO_EXTIPSELH_SET	RW	External Interrupt Port Select High
0x1408	GPIO_EXTIPINSELL_SET	RW	External Interrupt Pin Select Low
0x140C	GPIO_EXTIPINSELH_SET	RW	External Interrupt Pin Select High
0x1410	GPIO_EXTIRISE_SET	RW	External Interrupt Rising Edge Trigger
0x1414	GPIO_EXTIFALL_SET	RW	External Interrupt Falling Edge Trigger
0x1420	GPIO_IF_SET	RWH INTFLAG	Interrupt Flag
0x1424	GPIO_IEN_SET	RW	Interrupt Enable
0x142C	GPIO_EM4WUEN_SET	RW	EM4 Wakeup Enable
0x1430	GPIO_EM4WUPOL_SET	RW	EM4 Wakeup Polarity
0x1440	GPIO_DBGROUTEPEN_SET	RW	Debugger Route Pin Enable
0x1444	GPIO_TRACEROUTEPEN_SET	RW	Trace Route Pin Enable
0x1460	GPIO_LCDSEG_SET	RW	LCD Segment Enable
0x1470	GPIO_LCDCOM_SET	RW	LCD Common Enable

Offset	Name	Type	Description
0x1480	GPIO_ACMP0_ROUTEEN_SET	RW	ACMP0 Pin Enable
0x1484	GPIO_ACMP0_ACMPOUT-ROUTE_SET	RW	ACMPOUT Port/Pin Select
0x148C	GPIO_ACMP1_ROUTEEN_SET	RW	ACMP1 Pin Enable
0x1490	GPIO_ACMP1_ACMPOUT-ROUTE_SET	RW	ACMPOUT Port/Pin Select
0x1498	GPIO_CMU_ROUTEEN_SET	RW	CMU Pin Enable
0x149C	GPIO_CMU_CLKIN0ROUTE_SET	RW	CLKIN0 Port/Pin Select
0x14A0	GPIO_CMU_CLKOUT0ROUTE_SET	RW	CLKOUT0 Port/Pin Select
0x14A4	GPIO_CMU_CLKOUT1ROUTE_SET	RW	CLKOUT1 Port/Pin Select
0x14A8	GPIO_CMU_CLKOUT2ROUTE_SET	RW	CLKOUT2 Port/Pin Select
0x14C4	GPIO_EUSART0_ROU-TEEN_SET	RW	EUSART0 Pin Enable
0x14C8	GPIO_EU-SART0_CSROUTE_SET	RW	CS Port/Pin Select
0x14CC	GPIO_EU-SART0_CTSROUTE_SET	RW	CTS Port/Pin Select
0x14D0	GPIO_EU-SART0_RTSROUTE_SET	RW	RTS Port/Pin Select
0x14D4	GPIO_EU-SART0_RXROUTE_SET	RW	RX Port/Pin Select
0x14D8	GPIO_EU-SART0_SCLKROUTE_SET	RW	SCLK Port/Pin Select
0x14DC	GPIO_EU-SART0_TXROUTE_SET	RW	TX Port/Pin Select
0x14E4	GPIO_EUSART1_ROU-TEEN_SET	RW	EUSART1 Pin Enable
0x14E8	GPIO_EU-SART1_CSROUTE_SET	RW	CS Port/Pin Select
0x14EC	GPIO_EU-SART1_CTSROUTE_SET	RW	CTS Port/Pin Select
0x14F0	GPIO_EU-SART1_RTSROUTE_SET	RW	RTS Port/Pin Select
0x14F4	GPIO_EU-SART1_RXROUTE_SET	RW	RX Port/Pin Select
0x14F8	GPIO_EU-SART1_SCLKROUTE_SET	RW	SCLK Port/Pin Select
0x14FC	GPIO_EU-SART1_TXROUTE_SET	RW	TX Port/Pin Select
0x1504	GPIO_EUSART2_ROU-TEEN_SET	RW	EUSART2 Pin Enable

Offset	Name	Type	Description
0x1508	GPIO_EUSART2_CSROUTE_SET	RW	CS Port/Pin Select
0x150C	GPIO_EUSART2_CTSROUTE_SET	RW	CTS Port/Pin Select
0x1510	GPIO_EUSART2_RTSROUTE_SET	RW	RTS Port/Pin Select
0x1514	GPIO_EUSART2_RXROUTE_SET	RW	RX Port/Pin Select
0x1518	GPIO_EUSART2_SCLKROUTE_SET	RW	SCLK Port/Pin Select
0x151C	GPIO_EUSART2_TXROUTE_SET	RW	TX Port/Pin Select
0x1538	GPIO_I2C0_ROUTEEN_SET	RW	I2C0 Pin Enable
0x153C	GPIO_I2C0_SCLROUTE_SET	RW	SCL Port/Pin Select
0x1540	GPIO_I2C0_SDAROUTE_SET	RW	SDA Port/Pin Select
0x1548	GPIO_I2C1_ROUTEEN_SET	RW	I2C1 Pin Enable
0x154C	GPIO_I2C1_SCLROUTE_SET	RW	SCL Port/Pin Select
0x1550	GPIO_I2C1_SDAROUTE_SET	RW	SDA Port/Pin Select
0x1558	GPIO_KEYSCAN_ROUTEEN_SET	RW	KEYSCAN Pin Enable
0x155C	GPIO_KEYSCAN_COLOUT0ROUTE_SET	RW	COLOUT0 Port/Pin Select
0x1560	GPIO_KEYSCAN_COLOUT1ROUTE_SET	RW	COLOUT1 Port/Pin Select
0x1564	GPIO_KEYSCAN_COLOUT2ROUTE_SET	RW	COLOUT2 Port/Pin Select
0x1568	GPIO_KEYSCAN_COLOUT3ROUTE_SET	RW	COLOUT3 Port/Pin Select
0x156C	GPIO_KEYSCAN_COLOUT4ROUTE_SET	RW	COLOUT4 Port/Pin Select
0x1570	GPIO_KEYSCAN_COLOUT5ROUTE_SET	RW	COLOUT5 Port/Pin Select
0x1574	GPIO_KEYSCAN_COLOUT6ROUTE_SET	RW	COLOUT6 Port/Pin Select
0x1578	GPIO_KEYSCAN_COLOUT7ROUTE_SET	RW	COLOUT7 Port/Pin Select
0x157C	GPIO_KEYSCAN_ROWSENSE0ROUTE_SET	RW	ROWSENSE0 Port/Pin Select
0x1580	GPIO_KEYSCAN_ROWSENSE1ROUTE_SET	RW	ROWSENSE1 Port/Pin Select
0x1584	GPIO_KEYSCAN_ROWSENSE2ROUTE_SET	RW	ROWSENSE2 Port/Pin Select
0x1588	GPIO_KEYSCAN_ROWSENSE3ROUTE_SET	RW	ROWSENSE3 Port/Pin Select

Offset	Name	Type	Description
0x158C	GPIO_KEYSCAN_ROWSENSE4ROUTE_SET	RW	ROWSENSE4 Port/Pin Select
0x1590	GPIO_KEYSCAN_ROWSENSE5ROUTE_SET	RW	ROWSENSE5 Port/Pin Select
0x1598	GPIO_LESENSE_ROUTEEN_SET	RW	LESENSE Pin Enable
0x159C	GPIO_LESENSE_CH0OUTROUTE_SET	RW	CH0OUT Port/Pin Select
0x15A0	GPIO_LESENSE_CH1OUTROUTE_SET	RW	CH1OUT Port/Pin Select
0x15A4	GPIO_LESENSE_CH2OUTROUTE_SET	RW	CH2OUT Port/Pin Select
0x15A8	GPIO_LESENSE_CH3OUTROUTE_SET	RW	CH3OUT Port/Pin Select
0x15AC	GPIO_LESENSE_CH4OUTROUTE_SET	RW	CH4OUT Port/Pin Select
0x15B0	GPIO_LESENSE_CH5OUTROUTE_SET	RW	CH5OUT Port/Pin Select
0x15B4	GPIO_LESENSE_CH6OUTROUTE_SET	RW	CH6OUT Port/Pin Select
0x15B8	GPIO_LESENSE_CH7OUTROUTE_SET	RW	CH7OUT Port/Pin Select
0x15BC	GPIO_LESENSE_CH8OUTROUTE_SET	RW	CH8OUT Port/Pin Select
0x15C0	GPIO_LESENSE_CH9OUTROUTE_SET	RW	CH9OUT Port/Pin Select
0x15C4	GPIO_LESENSE_CH10OUTROUTE_SET	RW	CH10OUT Port/Pin Select
0x15C8	GPIO_LESENSE_CH11OUTROUTE_SET	RW	CH11OUT Port/Pin Select
0x15CC	GPIO_LESENSE_CH12OUTROUTE_SET	RW	CH12OUT Port/Pin Select
0x15D0	GPIO_LESENSE_CH13OUTROUTE_SET	RW	CH13OUT Port/Pin Select
0x15D4	GPIO_LESENSE_CH14OUTROUTE_SET	RW	CH14OUT Port/Pin Select
0x15D8	GPIO_LESENSE_CH15OUTROUTE_SET	RW	CH15OUT Port/Pin Select
0x15E0	GPIO_LETIMER_ROUTEEN_SET	RW	LETIMER Pin Enable
0x15E4	GPIO_LETIMER_OUT0ROUTE_SET	RW	OUT0 Port/Pin Select
0x15E8	GPIO_LETIMER_OUT1ROUTE_SET	RW	OUT1 Port/Pin Select
0x15F0	GPIO_MODEM_ROUTEEN_SET	RW	MODEM Pin Enable
0x15F4	GPIO_MODEM_ANT0ROUTE_SET	RW	ANT0 Port/Pin Select

Offset	Name	Type	Description
0x15F8	GPIO_MODEM_ANT1ROUTE_SET	RW	ANT1 Port/Pin Select
0x15FC	GPIO_MODEM_ANTROLLOVERROUTE_SET	RW	ANTROLLOVER Port/Pin Select
0x1600	GPIO_MODEM_ANTRR0ROUTE_SET	RW	ANTRR0 Port/Pin Select
0x1604	GPIO_MODEM_ANTRR1ROUTE_SET	RW	ANTRR1 Port/Pin Select
0x1608	GPIO_MODEM_ANTRR2ROUTE_SET	RW	ANTRR2 Port/Pin Select
0x160C	GPIO_MODEM_ANTRR3ROUTE_SET	RW	ANTRR3 Port/Pin Select
0x1610	GPIO_MODEM_ANTRR4ROUTE_SET	RW	ANTRR4 Port/Pin Select
0x1614	GPIO_MODEM_ANTRR5ROUTE_SET	RW	ANTRR5 Port/Pin Select
0x1618	GPIO_MODEM_ANTSWENROUTE_SET	RW	ANTSWEN Port/Pin Select
0x161C	GPIO_MODEM_ANTSWUSROUTE_SET	RW	ANTSWUS Port/Pin Select
0x1620	GPIO_MODEM_ANTTRIGROUTE_SET	RW	ANTTRIG Port/Pin Select
0x1624	GPIO_MODEM_ANTTRIGSTOPROUTE_SET	RW	ANTTRIGSTOP Port/Pin Select
0x1628	GPIO_MODEM_DCLKROUTE_SET	RW	DCLK Port/Pin Select
0x162C	GPIO_MODEM_DINROUTE_SET	RW	DIN Port/Pin Select
0x1630	GPIO_MODEM_DOUTROUTE_SET	RW	DOUT Port/Pin Select
0x163C	GPIO_PCNT0_S0INROUTE_SET	RW	S0IN Port/Pin Select
0x1640	GPIO_PCNT0_S1INROUTE_SET	RW	S1IN Port/Pin Select
0x1648	GPIO_PRS0_ROUTEEN_SET	RW	PRS0 Pin Enable
0x164C	GPIO_PRS0_ASYNCH0ROUTE_SET	RW	ASYNCH0 Port/Pin Select
0x1650	GPIO_PRS0_ASYNCH1ROUTE_SET	RW	ASYNCH1 Port/Pin Select
0x1654	GPIO_PRS0_ASYNCH2ROUTE_SET	RW	ASYNCH2 Port/Pin Select
0x1658	GPIO_PRS0_ASYNCH3ROUTE_SET	RW	ASYNCH3 Port/Pin Select
0x165C	GPIO_PRS0_ASYNCH4ROUTE_SET	RW	ASYNCH4 Port/Pin Select
0x1660	GPIO_PRS0_ASYNCH5ROUTE_SET	RW	ASYNCH5 Port/Pin Select

Offset	Name	Type	Description
0x1664	GPIO_PRS0_ASYNCH6ROUTE_SET	RW	ASYNCH6 Port/Pin Select
0x1668	GPIO_PRS0_ASYNCH7ROUTE_SET	RW	ASYNCH7 Port/Pin Select
0x166C	GPIO_PRS0_ASYNCH8ROUTE_SET	RW	ASYNCH8 Port/Pin Select
0x1670	GPIO_PRS0_ASYNCH9ROUTE_SET	RW	ASYNCH9 Port/Pin Select
0x1674	GPIO_PRS0_ASYNCH10ROUTE_SET	RW	ASYNCH10 Port/Pin Select
0x1678	GPIO_PRS0_ASYNCH11ROUTE_SET	RW	ASYNCH11 Port/Pin Select
0x167C	GPIO_PRS0_SYNCH0ROUTE_SET	RW	SYNCH0 Port/Pin Select
0x1680	GPIO_PRS0_SYNCH1ROUTE_SET	RW	SYNCH1 Port/Pin Select
0x1684	GPIO_PRS0_SYNCH2ROUTE_SET	RW	SYNCH2 Port/Pin Select
0x1688	GPIO_PRS0_SYNCH3ROUTE_SET	RW	SYNCH3 Port/Pin Select
0x16F0	GPIO_SYXO0_BUFOUTREQINASYNCROUTE_SET	RW	BUFOUTREQINASYNC Port/Pin Select
0x16F8	GPIO_TIMER0_ROUTEEN_SET	RW	TIMER0 Pin Enable
0x16FC	GPIO_TIMER0_CC0ROUTE_SET	RW	CC0 Port/Pin Select
0x1700	GPIO_TIMER0_CC1ROUTE_SET	RW	CC1 Port/Pin Select
0x1704	GPIO_TIMER0_CC2ROUTE_SET	RW	CC2 Port/Pin Select
0x1708	GPIO_TIMER0_CDTI0ROUTE_SET	RW	CDTI0 Port/Pin Select
0x170C	GPIO_TIMER0_CDTI1ROUTE_SET	RW	CDTI1 Port/Pin Select
0x1710	GPIO_TIMER0_CDTI2ROUTE_SET	RW	CDTI2 Port/Pin Select
0x1718	GPIO_TIMER1_ROUTEEN_SET	RW	TIMER1 Pin Enable
0x171C	GPIO_TIMER1_CC0ROUTE_SET	RW	CC0 Port/Pin Select
0x1720	GPIO_TIMER1_CC1ROUTE_SET	RW	CC1 Port/Pin Select
0x1724	GPIO_TIMER1_CC2ROUTE_SET	RW	CC2 Port/Pin Select
0x1728	GPIO_TIMER1_CDTI0ROUTE_SET	RW	CDTI0 Port/Pin Select
0x172C	GPIO_TIMER1_CDTI1ROUTE_SET	RW	CDTI1 Port/Pin Select

Offset	Name	Type	Description
0x1730	GPIO_TIMER1_CDTI2ROUTE_SET	RW	CDTI2 Port/Pin Select
0x1738	GPIO_TIMER2_ROUTEEN_SET	RW	TIMER2 Pin Enable
0x173C	GPIO_TIMER2_CC0ROUTE_SET	RW	CC0 Port/Pin Select
0x1740	GPIO_TIMER2_CC1ROUTE_SET	RW	CC1 Port/Pin Select
0x1744	GPIO_TIMER2_CC2ROUTE_SET	RW	CC2 Port/Pin Select
0x1748	GPIO_TIMER2_CDTI0ROUTE_SET	RW	CDTI0 Port/Pin Select
0x174C	GPIO_TIMER2_CDTI1ROUTE_SET	RW	CDTI1 Port/Pin Select
0x1750	GPIO_TIMER2_CDTI2ROUTE_SET	RW	CDTI2 Port/Pin Select
0x1758	GPIO_TIMER3_ROUTEEN_SET	RW	TIMER3 Pin Enable
0x175C	GPIO_TIMER3_CC0ROUTE_SET	RW	CC0 Port/Pin Select
0x1760	GPIO_TIMER3_CC1ROUTE_SET	RW	CC1 Port/Pin Select
0x1764	GPIO_TIMER3_CC2ROUTE_SET	RW	CC2 Port/Pin Select
0x1768	GPIO_TIMER3_CDTI0ROUTE_SET	RW	CDTI0 Port/Pin Select
0x176C	GPIO_TIMER3_CDTI1ROUTE_SET	RW	CDTI1 Port/Pin Select
0x1770	GPIO_TIMER3_CDTI2ROUTE_SET	RW	CDTI2 Port/Pin Select
0x1778	GPIO_TIMER4_ROUTEEN_SET	RW	TIMER4 Pin Enable
0x177C	GPIO_TIMER4_CC0ROUTE_SET	RW	CC0 Port/Pin Select
0x1780	GPIO_TIMER4_CC1ROUTE_SET	RW	CC1 Port/Pin Select
0x1784	GPIO_TIMER4_CC2ROUTE_SET	RW	CC2 Port/Pin Select
0x1788	GPIO_TIMER4_CDTI0ROUTE_SET	RW	CDTI0 Port/Pin Select
0x178C	GPIO_TIMER4_CDTI1ROUTE_SET	RW	CDTI1 Port/Pin Select
0x1790	GPIO_TIMER4_CDTI2ROUTE_SET	RW	CDTI2 Port/Pin Select
0x1798	GPIO_USART0_ROUTEEN_SET	RW	USART0 Pin Enable
0x179C	GPIO_USART0_CSROUTE_SET	RW	CS Port/Pin Select

Offset	Name	Type	Description
0x17A0	GPIO_USART0_CTSROUTE_SET	RW	CTS Port/Pin Select
0x17A4	GPIO_USART0_RTSROUTE_SET	RW	RTS Port/Pin Select
0x17A8	GPIO_USART0_RXROUTE_SET	RW	RX Port/Pin Select
0x17AC	GPIO_USART0_CLKROUTE_SET	RW	SCLK Port/Pin Select
0x17B0	GPIO_USART0_TXROUTE_SET	RW	TX Port/Pin Select
0x2000	GPIO_IPVERSION_CLR	R	Main
0x2030	GPIO_PORTA_CTRL_CLR	RW	Port Control
0x2034	GPIO_PORTA_MODEL_CLR	RW	Mode Low
0x203C	GPIO_PORTA_MODEH_CLR	RW	Mode High
0x2040	GPIO_PORTA_DOUT_CLR	RW	Data Out
0x2044	GPIO_PORTA_DIN_CLR	RH	Data in
0x2060	GPIO_PORTB_CTRL_CLR	RW	Port Control
0x2064	GPIO_PORTB_MODEL_CLR	RW	Mode Low
0x2070	GPIO_PORTB_DOUT_CLR	RW	Data Out
0x2074	GPIO_PORTB_DIN_CLR	RH	Data in
0x2090	GPIO_PORTC_CTRL_CLR	RW	Port Control
0x2094	GPIO_PORTC_MODEL_CLR	RW	Mode Low
0x209C	GPIO_PORTC_MODEH_CLR	RW	Mode High
0x20A0	GPIO_PORTC_DOUT_CLR	RW	Data Out
0x20A4	GPIO_PORTC_DIN_CLR	RH	Data in
0x20C0	GPIO_PORTD_CTRL_CLR	RW	Port Control
0x20C4	GPIO_PORTD_MODEL_CLR	RW	Mode Low
0x20CC	GPIO_PORTD_MODEH_CLR	RW	Mode High
0x20D0	GPIO_PORTD_DOUT_CLR	RW	Data Out
0x20D4	GPIO_PORTD_DIN_CLR	RH	Data in
0x2300	GPIO_LOCK_CLR	W	Lock Register
0x2310	GPIO_GPIOLOCKSTATUS_CLR	RH	Lock Status
0x2320	GPIO_ABUSALLOC_CLR	RW	A Bus Allocation
0x2324	GPIO_BBUSALLOC_CLR	RW	B Bus Allocation
0x2328	GPIO_CDBUSALLOC_CLR	RW	CD Bus Allocation
0x2400	GPIO_EXTIPSELL_CLR	RW	External Interrupt Port Select Low
0x2404	GPIO_EXTIPSELH_CLR	RW	External Interrupt Port Select High
0x2408	GPIO_EXTIPINSELL_CLR	RW	External Interrupt Pin Select Low
0x240C	GPIO_EXTIPINSELH_CLR	RW	External Interrupt Pin Select High
0x2410	GPIO_EXTIRISE_CLR	RW	External Interrupt Rising Edge Trigger

Offset	Name	Type	Description
0x2414	GPIO_EXTIFALL_CLR	RW	External Interrupt Falling Edge Trigger
0x2420	GPIO_IF_CLR	RWH INTFLAG	Interrupt Flag
0x2424	GPIO_IEN_CLR	RW	Interrupt Enable
0x242C	GPIO_EM4WUEN_CLR	RW	EM4 Wakeup Enable
0x2430	GPIO_EM4WUPOL_CLR	RW	EM4 Wakeup Polarity
0x2440	GPIO_DBGROUTEPEN_CLR	RW	Debugger Route Pin Enable
0x2444	GPIO_TRACEROUTEPEN_CLR	RW	Trace Route Pin Enable
0x2460	GPIO_LCDSEG_CLR	RW	LCD Segment Enable
0x2470	GPIO_LCDCOM_CLR	RW	LCD Common Enable
0x2480	GPIO_ACMP0_ROUTEEN_CLR	RW	ACMP0 Pin Enable
0x2484	GPIO_ACMP0_ACMPOUT-ROUTE_CLR	RW	ACMPOUT Port/Pin Select
0x248C	GPIO_ACMP1_ROUTEEN_CLR	RW	ACMP1 Pin Enable
0x2490	GPIO_ACMP1_ACMPOUT-ROUTE_CLR	RW	ACMPOUT Port/Pin Select
0x2498	GPIO_CMU_ROUTEEN_CLR	RW	CMU Pin Enable
0x249C	GPIO_CMU_CLKIN0ROUTE_CLR	RW	CLKIN0 Port/Pin Select
0x24A0	GPIO_CMU_CLKOUT0ROUTE_CLR	RW	CLKOUT0 Port/Pin Select
0x24A4	GPIO_CMU_CLKOUT1ROUTE_CLR	RW	CLKOUT1 Port/Pin Select
0x24A8	GPIO_CMU_CLKOUT2ROUTE_CLR	RW	CLKOUT2 Port/Pin Select
0x24C4	GPIO_EUSART0_ROU-TEEN_CLR	RW	EUSART0 Pin Enable
0x24C8	GPIO_EU-SART0_CSROUTE_CLR	RW	CS Port/Pin Select
0x24CC	GPIO_EU-SART0_CTSROUTE_CLR	RW	CTS Port/Pin Select
0x24D0	GPIO_EU-SART0_RTSROUTE_CLR	RW	RTS Port/Pin Select
0x24D4	GPIO_EU-SART0_RXROUTE_CLR	RW	RX Port/Pin Select
0x24D8	GPIO_EU-SART0_SCLKROUTE_CLR	RW	SCLK Port/Pin Select
0x24DC	GPIO_EU-SART0_TXROUTE_CLR	RW	TX Port/Pin Select
0x24E4	GPIO_EUSART1_ROU-TEEN_CLR	RW	EUSART1 Pin Enable
0x24E8	GPIO_EU-SART1_CSROUTE_CLR	RW	CS Port/Pin Select
0x24EC	GPIO_EU-SART1_CTSROUTE_CLR	RW	CTS Port/Pin Select

Offset	Name	Type	Description
0x24F0	GPIO_EUSART1_RTSROUTE_CLR	RW	RTS Port/Pin Select
0x24F4	GPIO_EUSART1_RXROUTE_CLR	RW	RX Port/Pin Select
0x24F8	GPIO_EUSART1_SCLKROUTE_CLR	RW	SCLK Port/Pin Select
0x24FC	GPIO_EUSART1_TXROUTE_CLR	RW	TX Port/Pin Select
0x2504	GPIO_EUSART2_ROUTEEN_CLR	RW	EUSART2 Pin Enable
0x2508	GPIO_EUSART2_CSROUTE_CLR	RW	CS Port/Pin Select
0x250C	GPIO_EUSART2_CTSROUTE_CLR	RW	CTS Port/Pin Select
0x2510	GPIO_EUSART2_RTSROUTE_CLR	RW	RTS Port/Pin Select
0x2514	GPIO_EUSART2_RXROUTE_CLR	RW	RX Port/Pin Select
0x2518	GPIO_EUSART2_SCLKROUTE_CLR	RW	SCLK Port/Pin Select
0x251C	GPIO_EUSART2_TXROUTE_CLR	RW	TX Port/Pin Select
0x2538	GPIO_I2C0_ROUTEEN_CLR	RW	I2C0 Pin Enable
0x253C	GPIO_I2C0_SCLROUTE_CLR	RW	SCL Port/Pin Select
0x2540	GPIO_I2C0_SDAROUTE_CLR	RW	SDA Port/Pin Select
0x2548	GPIO_I2C1_ROUTEEN_CLR	RW	I2C1 Pin Enable
0x254C	GPIO_I2C1_SCLROUTE_CLR	RW	SCL Port/Pin Select
0x2550	GPIO_I2C1_SDAROUTE_CLR	RW	SDA Port/Pin Select
0x2558	GPIO_KEYSCAN_ROUTEEN_CLR	RW	KEYSCAN Pin Enable
0x255C	GPIO_KEYSCAN_COLOUT0ROUTE_CLR	RW	COLOUT0 Port/Pin Select
0x2560	GPIO_KEYSCAN_COLOUT1ROUTE_CLR	RW	COLOUT1 Port/Pin Select
0x2564	GPIO_KEYSCAN_COLOUT2ROUTE_CLR	RW	COLOUT2 Port/Pin Select
0x2568	GPIO_KEYSCAN_COLOUT3ROUTE_CLR	RW	COLOUT3 Port/Pin Select
0x256C	GPIO_KEYSCAN_COLOUT4ROUTE_CLR	RW	COLOUT4 Port/Pin Select
0x2570	GPIO_KEYSCAN_COLOUT5ROUTE_CLR	RW	COLOUT5 Port/Pin Select
0x2574	GPIO_KEYSCAN_COLOUT6ROUTE_CLR	RW	COLOUT6 Port/Pin Select

Offset	Name	Type	Description
0x2578	GPIO_KEYSCAN_COL-OUT7ROUTE_CLR	RW	COLOUT7 Port/Pin Select
0x257C	GPIO_KEYSCAN_ROW-SENSE0ROUTE_CLR	RW	ROWSENSE0 Port/Pin Select
0x2580	GPIO_KEYSCAN_ROW-SENSE1ROUTE_CLR	RW	ROWSENSE1 Port/Pin Select
0x2584	GPIO_KEYSCAN_ROW-SENSE2ROUTE_CLR	RW	ROWSENSE2 Port/Pin Select
0x2588	GPIO_KEYSCAN_ROW-SENSE3ROUTE_CLR	RW	ROWSENSE3 Port/Pin Select
0x258C	GPIO_KEYSCAN_ROW-SENSE4ROUTE_CLR	RW	ROWSENSE4 Port/Pin Select
0x2590	GPIO_KEYSCAN_ROW-SENSE5ROUTE_CLR	RW	ROWSENSE5 Port/Pin Select
0x2598	GPIO_LESENSE_ROU-TEEN_CLR	RW	LESENSE Pin Enable
0x259C	GPIO_LESENSE_CH0OUT-ROUTE_CLR	RW	CH0OUT Port/Pin Select
0x25A0	GPIO_LESENSE_CH1OUT-ROUTE_CLR	RW	CH1OUT Port/Pin Select
0x25A4	GPIO_LESENSE_CH2OUT-ROUTE_CLR	RW	CH2OUT Port/Pin Select
0x25A8	GPIO_LESENSE_CH3OUT-ROUTE_CLR	RW	CH3OUT Port/Pin Select
0x25AC	GPIO_LESENSE_CH4OUT-ROUTE_CLR	RW	CH4OUT Port/Pin Select
0x25B0	GPIO_LESENSE_CH5OUT-ROUTE_CLR	RW	CH5OUT Port/Pin Select
0x25B4	GPIO_LESENSE_CH6OUT-ROUTE_CLR	RW	CH6OUT Port/Pin Select
0x25B8	GPIO_LESENSE_CH7OUT-ROUTE_CLR	RW	CH7OUT Port/Pin Select
0x25BC	GPIO_LESENSE_CH8OUT-ROUTE_CLR	RW	CH8OUT Port/Pin Select
0x25C0	GPIO_LESENSE_CH9OUT-ROUTE_CLR	RW	CH9OUT Port/Pin Select
0x25C4	GPIO_LESENSE_CH10OUT-ROUTE_CLR	RW	CH10OUT Port/Pin Select
0x25C8	GPIO_LESENSE_CH11OUT-ROUTE_CLR	RW	CH11OUT Port/Pin Select
0x25CC	GPIO_LESENSE_CH12OUT-ROUTE_CLR	RW	CH12OUT Port/Pin Select
0x25D0	GPIO_LESENSE_CH13OUT-ROUTE_CLR	RW	CH13OUT Port/Pin Select
0x25D4	GPIO_LESENSE_CH14OUT-ROUTE_CLR	RW	CH14OUT Port/Pin Select

Offset	Name	Type	Description
0x25D8	GPIO_LESENSE_CH15OUT-ROUTE_CLR	RW	CH15OUT Port/Pin Select
0x25E0	GPIO_LETIMER_ROU-TEEN_CLR	RW	LETIMER Pin Enable
0x25E4	GPIO_LETIM-ER_OUT0ROUTE_CLR	RW	OUT0 Port/Pin Select
0x25E8	GPIO_LETIM-ER_OUT1ROUTE_CLR	RW	OUT1 Port/Pin Select
0x25F0	GPIO_MODEM_ROUTEEN_CLR	RW	MODEM Pin Enable
0x25F4	GPIO_MO-DEM_ANT0ROUTE_CLR	RW	ANT0 Port/Pin Select
0x25F8	GPIO_MO-DEM_ANT1ROUTE_CLR	RW	ANT1 Port/Pin Select
0x25FC	GPIO_MODEM_ANTROLLO-VERROUTE_CLR	RW	ANTROLLOVER Port/Pin Select
0x2600	GPIO_MO-DEM_ANTRR0ROUTE_CLR	RW	ANTRR0 Port/Pin Select
0x2604	GPIO_MO-DEM_ANTRR1ROUTE_CLR	RW	ANTRR1 Port/Pin Select
0x2608	GPIO_MO-DEM_ANTRR2ROUTE_CLR	RW	ANTRR2 Port/Pin Select
0x260C	GPIO_MO-DEM_ANTRR3ROUTE_CLR	RW	ANTRR3 Port/Pin Select
0x2610	GPIO_MO-DEM_ANTRR4ROUTE_CLR	RW	ANTRR4 Port/Pin Select
0x2614	GPIO_MO-DEM_ANTRR5ROUTE_CLR	RW	ANTRR5 Port/Pin Select
0x2618	GPIO_MODEM_ANTSWEN-ROUTE_CLR	RW	ANTSWEN Port/Pin Select
0x261C	GPIO_MODEM_ANTSWUS-ROUTE_CLR	RW	ANTSWUS Port/Pin Select
0x2620	GPIO_MODEM_ANTTRIG-ROUTE_CLR	RW	ANTTRIG Port/Pin Select
0x2624	GPIO_MODEM_ANTTRIGSTOP-ROUTE_CLR	RW	ANTTRIGSTOP Port/Pin Select
0x2628	GPIO_MO-DEM_DCLKROUTE_CLR	RW	DCLK Port/Pin Select
0x262C	GPIO_MODEM_DIN-ROUTE_CLR	RW	DIN Port/Pin Select
0x2630	GPIO_MODEM_DOUT-ROUTE_CLR	RW	DOUT Port/Pin Select
0x263C	GPIO_PCNT0_S0IN-ROUTE_CLR	RW	S0IN Port/Pin Select
0x2640	GPIO_PCNT0_S1IN-ROUTE_CLR	RW	S1IN Port/Pin Select
0x2648	GPIO_PRS0_ROUTEEN_CLR	RW	PRS0 Pin Enable

Offset	Name	Type	Description
0x264C	GPIO_PRS0_ASYNCH0ROUTE_CLR	RW	ASYNCH0 Port/Pin Select
0x2650	GPIO_PRS0_ASYNCH1ROUTE_CLR	RW	ASYNCH1 Port/Pin Select
0x2654	GPIO_PRS0_ASYNCH2ROUTE_CLR	RW	ASYNCH2 Port/Pin Select
0x2658	GPIO_PRS0_ASYNCH3ROUTE_CLR	RW	ASYNCH3 Port/Pin Select
0x265C	GPIO_PRS0_ASYNCH4ROUTE_CLR	RW	ASYNCH4 Port/Pin Select
0x2660	GPIO_PRS0_ASYNCH5ROUTE_CLR	RW	ASYNCH5 Port/Pin Select
0x2664	GPIO_PRS0_ASYNCH6ROUTE_CLR	RW	ASYNCH6 Port/Pin Select
0x2668	GPIO_PRS0_ASYNCH7ROUTE_CLR	RW	ASYNCH7 Port/Pin Select
0x266C	GPIO_PRS0_ASYNCH8ROUTE_CLR	RW	ASYNCH8 Port/Pin Select
0x2670	GPIO_PRS0_ASYNCH9ROUTE_CLR	RW	ASYNCH9 Port/Pin Select
0x2674	GPIO_PRS0_ASYNCH10ROUTE_CLR	RW	ASYNCH10 Port/Pin Select
0x2678	GPIO_PRS0_ASYNCH11ROUTE_CLR	RW	ASYNCH11 Port/Pin Select
0x267C	GPIO_PRS0_SYNCH0ROUTE_CLR	RW	SYNCH0 Port/Pin Select
0x2680	GPIO_PRS0_SYNCH1ROUTE_CLR	RW	SYNCH1 Port/Pin Select
0x2684	GPIO_PRS0_SYNCH2ROUTE_CLR	RW	SYNCH2 Port/Pin Select
0x2688	GPIO_PRS0_SYNCH3ROUTE_CLR	RW	SYNCH3 Port/Pin Select
0x26F0	GPIO_SYXO0_BUFOUTREQINASYNCROUTE_CLR	RW	BUFOUTREQINASYNC Port/Pin Select
0x26F8	GPIO_TIMER0_ROUTEEN_CLR	RW	TIMER0 Pin Enable
0x26FC	GPIO_TIMER0_CC0ROUTE_CLR	RW	CC0 Port/Pin Select
0x2700	GPIO_TIMER0_CC1ROUTE_CLR	RW	CC1 Port/Pin Select
0x2704	GPIO_TIMER0_CC2ROUTE_CLR	RW	CC2 Port/Pin Select
0x2708	GPIO_TIMER0_CDTI0ROUTE_CLR	RW	CDTI0 Port/Pin Select
0x270C	GPIO_TIMER0_CDTI1ROUTE_CLR	RW	CDTI1 Port/Pin Select
0x2710	GPIO_TIMER0_CDTI2ROUTE_CLR	RW	CDTI2 Port/Pin Select

Offset	Name	Type	Description
0x2718	GPIO_TIMER1_ROUTEEN_CLR	RW	TIMER1 Pin Enable
0x271C	GPIO_TIMER1_CC0ROUTE_CLR	RW	CC0 Port/Pin Select
0x2720	GPIO_TIMER1_CC1ROUTE_CLR	RW	CC1 Port/Pin Select
0x2724	GPIO_TIMER1_CC2ROUTE_CLR	RW	CC2 Port/Pin Select
0x2728	GPIO_TIMER1_CDTI0ROUTE_CLR	RW	CDTI0 Port/Pin Select
0x272C	GPIO_TIMER1_CDTI1ROUTE_CLR	RW	CDTI1 Port/Pin Select
0x2730	GPIO_TIMER1_CDTI2ROUTE_CLR	RW	CDTI2 Port/Pin Select
0x2738	GPIO_TIMER2_ROUTEEN_CLR	RW	TIMER2 Pin Enable
0x273C	GPIO_TIMER2_CC0ROUTE_CLR	RW	CC0 Port/Pin Select
0x2740	GPIO_TIMER2_CC1ROUTE_CLR	RW	CC1 Port/Pin Select
0x2744	GPIO_TIMER2_CC2ROUTE_CLR	RW	CC2 Port/Pin Select
0x2748	GPIO_TIMER2_CDTI0ROUTE_CLR	RW	CDTI0 Port/Pin Select
0x274C	GPIO_TIMER2_CDTI1ROUTE_CLR	RW	CDTI1 Port/Pin Select
0x2750	GPIO_TIMER2_CDTI2ROUTE_CLR	RW	CDTI2 Port/Pin Select
0x2758	GPIO_TIMER3_ROUTEEN_CLR	RW	TIMER3 Pin Enable
0x275C	GPIO_TIMER3_CC0ROUTE_CLR	RW	CC0 Port/Pin Select
0x2760	GPIO_TIMER3_CC1ROUTE_CLR	RW	CC1 Port/Pin Select
0x2764	GPIO_TIMER3_CC2ROUTE_CLR	RW	CC2 Port/Pin Select
0x2768	GPIO_TIMER3_CDTI0ROUTE_CLR	RW	CDTI0 Port/Pin Select
0x276C	GPIO_TIMER3_CDTI1ROUTE_CLR	RW	CDTI1 Port/Pin Select
0x2770	GPIO_TIMER3_CDTI2ROUTE_CLR	RW	CDTI2 Port/Pin Select
0x2778	GPIO_TIMER4_ROUTEEN_CLR	RW	TIMER4 Pin Enable
0x277C	GPIO_TIMER4_CC0ROUTE_CLR	RW	CC0 Port/Pin Select
0x2780	GPIO_TIMER4_CC1ROUTE_CLR	RW	CC1 Port/Pin Select
0x2784	GPIO_TIMER4_CC2ROUTE_CLR	RW	CC2 Port/Pin Select

Offset	Name	Type	Description
0x2788	GPIO_TIMER4_CDTI0ROUTE_CLR	RW	CDTI0 Port/Pin Select
0x278C	GPIO_TIMER4_CDTI1ROUTE_CLR	RW	CDTI1 Port/Pin Select
0x2790	GPIO_TIMER4_CDTI2ROUTE_CLR	RW	CDTI2 Port/Pin Select
0x2798	GPIO_USART0_ROUTEEN_CLR	RW	USART0 Pin Enable
0x279C	GPIO_USART0_CSROUTE_CLR	RW	CS Port/Pin Select
0x27A0	GPIO_USART0_CTSROUTE_CLR	RW	CTS Port/Pin Select
0x27A4	GPIO_USART0_RTSROUTE_CLR	RW	RTS Port/Pin Select
0x27A8	GPIO_USART0_RXROUTE_CLR	RW	RX Port/Pin Select
0x27AC	GPIO_USART0_CLKROUTE_CLR	RW	SCLK Port/Pin Select
0x27B0	GPIO_USART0_TXROUTE_CLR	RW	TX Port/Pin Select
0x3000	GPIO_IPVERSION_TGL	R	Main
0x3030	GPIO_PORTA_CTRL_TGL	RW	Port Control
0x3034	GPIO_PORTA_MODEL_TGL	RW	Mode Low
0x303C	GPIO_PORTA_MODEH_TGL	RW	Mode High
0x3040	GPIO_PORTA_DOUT_TGL	RW	Data Out
0x3044	GPIO_PORTA_DIN_TGL	RH	Data in
0x3060	GPIO_PORTB_CTRL_TGL	RW	Port Control
0x3064	GPIO_PORTB_MODEL_TGL	RW	Mode Low
0x3070	GPIO_PORTB_DOUT_TGL	RW	Data Out
0x3074	GPIO_PORTB_DIN_TGL	RH	Data in
0x3090	GPIO_PORTC_CTRL_TGL	RW	Port Control
0x3094	GPIO_PORTC_MODEL_TGL	RW	Mode Low
0x309C	GPIO_PORTC_MODEH_TGL	RW	Mode High
0x30A0	GPIO_PORTC_DOUT_TGL	RW	Data Out
0x30A4	GPIO_PORTC_DIN_TGL	RH	Data in
0x30C0	GPIO_PORTD_CTRL_TGL	RW	Port Control
0x30C4	GPIO_PORTD_MODEL_TGL	RW	Mode Low
0x30CC	GPIO_PORTD_MODEH_TGL	RW	Mode High
0x30D0	GPIO_PORTD_DOUT_TGL	RW	Data Out
0x30D4	GPIO_PORTD_DIN_TGL	RH	Data in
0x3300	GPIO_LOCK_TGL	W	Lock Register
0x3310	GPIO_GPIOLOCKSTATUS_TGL	RH	Lock Status

Offset	Name	Type	Description
0x3320	GPIO_ABUSALLOC_TGL	RW	A Bus Allocation
0x3324	GPIO_BBUSALLOC_TGL	RW	B Bus Allocation
0x3328	GPIO_CDBUSALLOC_TGL	RW	CD Bus Allocation
0x3400	GPIO_EXTIPSELL_TGL	RW	External Interrupt Port Select Low
0x3404	GPIO_EXTIPSELH_TGL	RW	External Interrupt Port Select High
0x3408	GPIO_EXTIPINSELL_TGL	RW	External Interrupt Pin Select Low
0x340C	GPIO_EXTIPINSELH_TGL	RW	External Interrupt Pin Select High
0x3410	GPIO_EXTIRISE_TGL	RW	External Interrupt Rising Edge Trigger
0x3414	GPIO_EXTIFALL_TGL	RW	External Interrupt Falling Edge Trigger
0x3420	GPIO_IF_TGL	RWH INTFLAG	Interrupt Flag
0x3424	GPIO_IEN_TGL	RW	Interrupt Enable
0x342C	GPIO_EM4WUEN_TGL	RW	EM4 Wakeup Enable
0x3430	GPIO_EM4WUPOL_TGL	RW	EM4 Wakeup Polarity
0x3440	GPIO_DBGROUTEPEN_TGL	RW	Debugger Route Pin Enable
0x3444	GPIO_TRACEROUTEPEN_TGL	RW	Trace Route Pin Enable
0x3460	GPIO_LCDSEG_TGL	RW	LCD Segment Enable
0x3470	GPIO_LCDCOM_TGL	RW	LCD Common Enable
0x3480	GPIO_ACMP0_ROUTEEN_TGL	RW	ACMP0 Pin Enable
0x3484	GPIO_ACMP0_ACMPOUT-ROUTE_TGL	RW	ACMPOUT Port/Pin Select
0x348C	GPIO_ACMP1_ROUTEEN_TGL	RW	ACMP1 Pin Enable
0x3490	GPIO_ACMP1_ACMPOUT-ROUTE_TGL	RW	ACMPOUT Port/Pin Select
0x3498	GPIO_CMU_ROUTEEN_TGL	RW	CMU Pin Enable
0x349C	GPIO_CMU_CLKIN0ROUTE_TGL	RW	CLKIN0 Port/Pin Select
0x34A0	GPIO_CMU_CLKOUT0ROUTE_TGL	RW	CLKOUT0 Port/Pin Select
0x34A4	GPIO_CMU_CLKOUT1ROUTE_TGL	RW	CLKOUT1 Port/Pin Select
0x34A8	GPIO_CMU_CLKOUT2ROUTE_TGL	RW	CLKOUT2 Port/Pin Select
0x34C4	GPIO_EUSART0_ROU-TEEN_TGL	RW	EUSART0 Pin Enable
0x34C8	GPIO_EU-SART0_CSROUTE_TGL	RW	CS Port/Pin Select
0x34CC	GPIO_EU-SART0_CTSROUTE_TGL	RW	CTS Port/Pin Select
0x34D0	GPIO_EU-SART0_RTSROUTE_TGL	RW	RTS Port/Pin Select
0x34D4	GPIO_EU-SART0_RXROUTE_TGL	RW	RX Port/Pin Select

Offset	Name	Type	Description
0x34D8	GPIO_EUSART0_SCLKROUTE_TGL	RW	SCLK Port/Pin Select
0x34DC	GPIO_EUSART0_TXROUTE_TGL	RW	TX Port/Pin Select
0x34E4	GPIO_EUSART1_ROUTEEN_TGL	RW	EUSART1 Pin Enable
0x34E8	GPIO_EUSART1_CSROUTE_TGL	RW	CS Port/Pin Select
0x34EC	GPIO_EUSART1_CTSROUTE_TGL	RW	CTS Port/Pin Select
0x34F0	GPIO_EUSART1_RTSROUTE_TGL	RW	RTS Port/Pin Select
0x34F4	GPIO_EUSART1_RXROUTE_TGL	RW	RX Port/Pin Select
0x34F8	GPIO_EUSART1_SCLKROUTE_TGL	RW	SCLK Port/Pin Select
0x34FC	GPIO_EUSART1_TXROUTE_TGL	RW	TX Port/Pin Select
0x3504	GPIO_EUSART2_ROUTEEN_TGL	RW	EUSART2 Pin Enable
0x3508	GPIO_EUSART2_CSROUTE_TGL	RW	CS Port/Pin Select
0x350C	GPIO_EUSART2_CTSROUTE_TGL	RW	CTS Port/Pin Select
0x3510	GPIO_EUSART2_RTSROUTE_TGL	RW	RTS Port/Pin Select
0x3514	GPIO_EUSART2_RXROUTE_TGL	RW	RX Port/Pin Select
0x3518	GPIO_EUSART2_SCLKROUTE_TGL	RW	SCLK Port/Pin Select
0x351C	GPIO_EUSART2_TXROUTE_TGL	RW	TX Port/Pin Select
0x3538	GPIO_I2C0_ROUTEEN_TGL	RW	I2C0 Pin Enable
0x353C	GPIO_I2C0_SCLROUTE_TGL	RW	SCL Port/Pin Select
0x3540	GPIO_I2C0_SDAROUTE_TGL	RW	SDA Port/Pin Select
0x3548	GPIO_I2C1_ROUTEEN_TGL	RW	I2C1 Pin Enable
0x354C	GPIO_I2C1_SCLROUTE_TGL	RW	SCL Port/Pin Select
0x3550	GPIO_I2C1_SDAROUTE_TGL	RW	SDA Port/Pin Select
0x3558	GPIO_KEYSCAN_ROUTEEN_TGL	RW	KEYSCAN Pin Enable
0x355C	GPIO_KEYSCAN_COUT0ROUTE_TGL	RW	COLOUT0 Port/Pin Select
0x3560	GPIO_KEYSCAN_COUT1ROUTE_TGL	RW	COLOUT1 Port/Pin Select

Offset	Name	Type	Description
0x3564	GPIO_KEYSCAN_COL-OUT2ROUTE_TGL	RW	COLOUT2 Port/Pin Select
0x3568	GPIO_KEYSCAN_COL-OUT3ROUTE_TGL	RW	COLOUT3 Port/Pin Select
0x356C	GPIO_KEYSCAN_COL-OUT4ROUTE_TGL	RW	COLOUT4 Port/Pin Select
0x3570	GPIO_KEYSCAN_COL-OUT5ROUTE_TGL	RW	COLOUT5 Port/Pin Select
0x3574	GPIO_KEYSCAN_COL-OUT6ROUTE_TGL	RW	COLOUT6 Port/Pin Select
0x3578	GPIO_KEYSCAN_COL-OUT7ROUTE_TGL	RW	COLOUT7 Port/Pin Select
0x357C	GPIO_KEYSCAN_ROW-SENSE0ROUTE_TGL	RW	ROWSENSE0 Port/Pin Select
0x3580	GPIO_KEYSCAN_ROW-SENSE1ROUTE_TGL	RW	ROWSENSE1 Port/Pin Select
0x3584	GPIO_KEYSCAN_ROW-SENSE2ROUTE_TGL	RW	ROWSENSE2 Port/Pin Select
0x3588	GPIO_KEYSCAN_ROW-SENSE3ROUTE_TGL	RW	ROWSENSE3 Port/Pin Select
0x358C	GPIO_KEYSCAN_ROW-SENSE4ROUTE_TGL	RW	ROWSENSE4 Port/Pin Select
0x3590	GPIO_KEYSCAN_ROW-SENSE5ROUTE_TGL	RW	ROWSENSE5 Port/Pin Select
0x3598	GPIO_LESENSE_ROU-TEEN_TGL	RW	LESENSE Pin Enable
0x359C	GPIO_LESENSE_CH0OUT-ROUTE_TGL	RW	CH0OUT Port/Pin Select
0x35A0	GPIO_LESENSE_CH1OUT-ROUTE_TGL	RW	CH1OUT Port/Pin Select
0x35A4	GPIO_LESENSE_CH2OUT-ROUTE_TGL	RW	CH2OUT Port/Pin Select
0x35A8	GPIO_LESENSE_CH3OUT-ROUTE_TGL	RW	CH3OUT Port/Pin Select
0x35AC	GPIO_LESENSE_CH4OUT-ROUTE_TGL	RW	CH4OUT Port/Pin Select
0x35B0	GPIO_LESENSE_CH5OUT-ROUTE_TGL	RW	CH5OUT Port/Pin Select
0x35B4	GPIO_LESENSE_CH6OUT-ROUTE_TGL	RW	CH6OUT Port/Pin Select
0x35B8	GPIO_LESENSE_CH7OUT-ROUTE_TGL	RW	CH7OUT Port/Pin Select
0x35BC	GPIO_LESENSE_CH8OUT-ROUTE_TGL	RW	CH8OUT Port/Pin Select
0x35C0	GPIO_LESENSE_CH9OUT-ROUTE_TGL	RW	CH9OUT Port/Pin Select

Offset	Name	Type	Description
0x35C4	GPIO_LESENSE_CH10OUT-ROUTE_TGL	RW	CH10OUT Port/Pin Select
0x35C8	GPIO_LESENSE_CH11OUT-ROUTE_TGL	RW	CH11OUT Port/Pin Select
0x35CC	GPIO_LESENSE_CH12OUT-ROUTE_TGL	RW	CH12OUT Port/Pin Select
0x35D0	GPIO_LESENSE_CH13OUT-ROUTE_TGL	RW	CH13OUT Port/Pin Select
0x35D4	GPIO_LESENSE_CH14OUT-ROUTE_TGL	RW	CH14OUT Port/Pin Select
0x35D8	GPIO_LESENSE_CH15OUT-ROUTE_TGL	RW	CH15OUT Port/Pin Select
0x35E0	GPIO_LETIMER_ROU-TEEN_TGL	RW	LETIMER Pin Enable
0x35E4	GPIO_LETIM-ER_OUT0ROUTE_TGL	RW	OUT0 Port/Pin Select
0x35E8	GPIO_LETIM-ER_OUT1ROUTE_TGL	RW	OUT1 Port/Pin Select
0x35F0	GPIO_MODEM_ROUTEEN_TGL	RW	MODEM Pin Enable
0x35F4	GPIO_MO-DEM_ANT0ROUTE_TGL	RW	ANT0 Port/Pin Select
0x35F8	GPIO_MO-DEM_ANT1ROUTE_TGL	RW	ANT1 Port/Pin Select
0x35FC	GPIO_MODEM_ANTROLLO-VERROUTE_TGL	RW	ANTROLLOVER Port/Pin Select
0x3600	GPIO_MO-DEM_ANTRR0ROUTE_TGL	RW	ANTRR0 Port/Pin Select
0x3604	GPIO_MO-DEM_ANTRR1ROUTE_TGL	RW	ANTRR1 Port/Pin Select
0x3608	GPIO_MO-DEM_ANTRR2ROUTE_TGL	RW	ANTRR2 Port/Pin Select
0x360C	GPIO_MO-DEM_ANTRR3ROUTE_TGL	RW	ANTRR3 Port/Pin Select
0x3610	GPIO_MO-DEM_ANTRR4ROUTE_TGL	RW	ANTRR4 Port/Pin Select
0x3614	GPIO_MO-DEM_ANTRR5ROUTE_TGL	RW	ANTRR5 Port/Pin Select
0x3618	GPIO_MODEM_ANTSWEN-ROUTE_TGL	RW	ANTSWEN Port/Pin Select
0x361C	GPIO_MODEM_ANTSWUS-ROUTE_TGL	RW	ANTSWUS Port/Pin Select
0x3620	GPIO_MODEM_ANTTRIG-ROUTE_TGL	RW	ANTTRIG Port/Pin Select
0x3624	GPIO_MODEM_ANTTRIGSTOP-ROUTE_TGL	RW	ANTTRIGSTOP Port/Pin Select
0x3628	GPIO_MO-DEM_DCLKROUTE_TGL	RW	DCLK Port/Pin Select

Offset	Name	Type	Description
0x362C	GPIO_MODEM_DIN-ROUTE_TGL	RW	DIN Port/Pin Select
0x3630	GPIO_MODEM_DOUT-ROUTE_TGL	RW	DOUT Port/Pin Select
0x363C	GPIO_PCNT0_S0IN-ROUTE_TGL	RW	S0IN Port/Pin Select
0x3640	GPIO_PCNT0_S1IN-ROUTE_TGL	RW	S1IN Port/Pin Select
0x3648	GPIO_PRS0_ROUTEEN_TGL	RW	PRS0 Pin Enable
0x364C	GPIO_PRS0_ASYNCH0ROUTE_TGL	RW	ASYNCH0 Port/Pin Select
0x3650	GPIO_PRS0_ASYNCH1ROUTE_TGL	RW	ASYNCH1 Port/Pin Select
0x3654	GPIO_PRS0_ASYNCH2ROUTE_TGL	RW	ASYNCH2 Port/Pin Select
0x3658	GPIO_PRS0_ASYNCH3ROUTE_TGL	RW	ASYNCH3 Port/Pin Select
0x365C	GPIO_PRS0_ASYNCH4ROUTE_TGL	RW	ASYNCH4 Port/Pin Select
0x3660	GPIO_PRS0_ASYNCH5ROUTE_TGL	RW	ASYNCH5 Port/Pin Select
0x3664	GPIO_PRS0_ASYNCH6ROUTE_TGL	RW	ASYNCH6 Port/Pin Select
0x3668	GPIO_PRS0_ASYNCH7ROUTE_TGL	RW	ASYNCH7 Port/Pin Select
0x366C	GPIO_PRS0_ASYNCH8ROUTE_TGL	RW	ASYNCH8 Port/Pin Select
0x3670	GPIO_PRS0_ASYNCH9ROUTE_TGL	RW	ASYNCH9 Port/Pin Select
0x3674	GPIO_PRS0_ASYNCH10ROUTE_TGL	RW	ASYNCH10 Port/Pin Select
0x3678	GPIO_PRS0_ASYNCH11ROUTE_TGL	RW	ASYNCH11 Port/Pin Select
0x367C	GPIO_PRS0_SYNCH0ROUTE_TGL	RW	SYNCH0 Port/Pin Select
0x3680	GPIO_PRS0_SYNCH1ROUTE_TGL	RW	SYNCH1 Port/Pin Select
0x3684	GPIO_PRS0_SYNCH2ROUTE_TGL	RW	SYNCH2 Port/Pin Select
0x3688	GPIO_PRS0_SYNCH3ROUTE_TGL	RW	SYNCH3 Port/Pin Select
0x36F0	GPIO_SYXO0_BUFOUTREQI-NASYNCROUTE_TGL	RW	BUFOUTREQINASYNC Port/Pin Select
0x36F8	GPIO_TIMER0_ROUTEEN_TGL	RW	TIMER0 Pin Enable
0x36FC	GPIO_TIM-ER0_CC0ROUTE_TGL	RW	CC0 Port/Pin Select

Offset	Name	Type	Description
0x3700	GPIO_TIMER0_CC1ROUTE_TGL	RW	CC1 Port/Pin Select
0x3704	GPIO_TIMER0_CC2ROUTE_TGL	RW	CC2 Port/Pin Select
0x3708	GPIO_TIMER0_CDTI0ROUTE_TGL	RW	CDTI0 Port/Pin Select
0x370C	GPIO_TIMER0_CDTI1ROUTE_TGL	RW	CDTI1 Port/Pin Select
0x3710	GPIO_TIMER0_CDTI2ROUTE_TGL	RW	CDTI2 Port/Pin Select
0x3718	GPIO_TIMER1_ROUTEEN_TGL	RW	TIMER1 Pin Enable
0x371C	GPIO_TIMER1_CC0ROUTE_TGL	RW	CC0 Port/Pin Select
0x3720	GPIO_TIMER1_CC1ROUTE_TGL	RW	CC1 Port/Pin Select
0x3724	GPIO_TIMER1_CC2ROUTE_TGL	RW	CC2 Port/Pin Select
0x3728	GPIO_TIMER1_CDTI0ROUTE_TGL	RW	CDTI0 Port/Pin Select
0x372C	GPIO_TIMER1_CDTI1ROUTE_TGL	RW	CDTI1 Port/Pin Select
0x3730	GPIO_TIMER1_CDTI2ROUTE_TGL	RW	CDTI2 Port/Pin Select
0x3738	GPIO_TIMER2_ROUTEEN_TGL	RW	TIMER2 Pin Enable
0x373C	GPIO_TIMER2_CC0ROUTE_TGL	RW	CC0 Port/Pin Select
0x3740	GPIO_TIMER2_CC1ROUTE_TGL	RW	CC1 Port/Pin Select
0x3744	GPIO_TIMER2_CC2ROUTE_TGL	RW	CC2 Port/Pin Select
0x3748	GPIO_TIMER2_CDTI0ROUTE_TGL	RW	CDTI0 Port/Pin Select
0x374C	GPIO_TIMER2_CDTI1ROUTE_TGL	RW	CDTI1 Port/Pin Select
0x3750	GPIO_TIMER2_CDTI2ROUTE_TGL	RW	CDTI2 Port/Pin Select
0x3758	GPIO_TIMER3_ROUTEEN_TGL	RW	TIMER3 Pin Enable
0x375C	GPIO_TIMER3_CC0ROUTE_TGL	RW	CC0 Port/Pin Select
0x3760	GPIO_TIMER3_CC1ROUTE_TGL	RW	CC1 Port/Pin Select
0x3764	GPIO_TIMER3_CC2ROUTE_TGL	RW	CC2 Port/Pin Select
0x3768	GPIO_TIMER3_CDTI0ROUTE_TGL	RW	CDTI0 Port/Pin Select

Offset	Name	Type	Description
0x376C	GPIO_TIMER3_CDTI1ROUTE_TGL	RW	CDTI1 Port/Pin Select
0x3770	GPIO_TIMER3_CDTI2ROUTE_TGL	RW	CDTI2 Port/Pin Select
0x3778	GPIO_TIMER4_ROUTEEN_TGL	RW	TIMER4 Pin Enable
0x377C	GPIO_TIMER4_CC0ROUTE_TGL	RW	CC0 Port/Pin Select
0x3780	GPIO_TIMER4_CC1ROUTE_TGL	RW	CC1 Port/Pin Select
0x3784	GPIO_TIMER4_CC2ROUTE_TGL	RW	CC2 Port/Pin Select
0x3788	GPIO_TIMER4_CDTI0ROUTE_TGL	RW	CDTI0 Port/Pin Select
0x378C	GPIO_TIMER4_CDTI1ROUTE_TGL	RW	CDTI1 Port/Pin Select
0x3790	GPIO_TIMER4_CDTI2ROUTE_TGL	RW	CDTI2 Port/Pin Select
0x3798	GPIO_USART0_ROUTEEN_TGL	RW	USART0 Pin Enable
0x379C	GPIO_USART0_CSROUTE_TGL	RW	CS Port/Pin Select
0x37A0	GPIO_USART0_CTSROUTE_TGL	RW	CTS Port/Pin Select
0x37A4	GPIO_USART0_RTSROUTE_TGL	RW	RTS Port/Pin Select
0x37A8	GPIO_USART0_RXROUTE_TGL	RW	RX Port/Pin Select
0x37AC	GPIO_USART0_CLKROUTE_TGL	RW	SCLK Port/Pin Select
0x37B0	GPIO_USART0_TXROUTE_TGL	RW	TX Port/Pin Select

23.6 GPIO Register Description

23.6.1 GPIO_IPVERSION - Main

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x6																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x6	R	**ip version id**
	IPVERSION ID			

23.6.2 GPIO_PORTA_CTRL - Port Control

Offset	Bit Position																															
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x0							0x4									0x0							0x4						
Access				RW							RW									RW							RW						
Name				DINDISALT							SLEWRATEALT									DINDIS							SLEWRATE						

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28	DINDISALT	0x0	RW	**Data In Disable Alt**
	Data input disable for port pins using alternate modes.			
27:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:20	SLEWRATEALT	0x4	RW	**Slew Rate Alt**
	Slewrate limit for port pins using alternate modes. Higher values provide faster slewrates.			
19:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	DINDIS	0x0	RW	**Data In Disable**
	Data input disable for port pins not using alternate modes.			
11:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	SLEWRATE	0x4	RW	**Slew Rate**
	Slewrate limit for port pins using not alternate modes. Higher values provide faster slewrates.			
3:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

23.6.3 GPIO_PORTA_MODEL - Mode Low

Offset	Bit Position																													
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0				0x0				0x0				0x0				0x0				0x0				0x0				0x0			
Access	RW				RW				RW				RW				RW				RW				RW				RW			
Name	MODE7				MODE6				MODE5				MODE4				MODE3				MODE2				MODE1				MODE0			

Bit	Name	Reset	Access	Description
31:28	MODE7	0x0	RW	**MODE n**
	MODE n			

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

Bit	Name	Reset	Access	Description
27:24	MODE6	0x0	RW	**MODE n**
	MODE n			

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.

Bit	Name	Reset	Access	Description
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
23:20	MODE5 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
19:16	MODE4 MODE n	0x0	RW	**MODE n**

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
15:12	MODE3 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.

Bit	Name	Reset	Access	Description
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
11:8	MODE2 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

7:4	MODE1 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.

Bit	Name	Reset	Access	Description
	11	WIREDANDPULLUPFILTER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUPFILTER		Open-drain output using alternate control with filter and pullup.
3:0	MODE0 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFILTER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUPFILTER	Open-drain output using alternate control with filter and pullup.

23.6.4 GPIO_PORTA_MODEH - Mode High

Offset	Bit Position																													
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0				0x0				0x0				0x0				0x0				0x0				0x0			
Access					RW				RW				RW				RW				RW				RW				RW			
Name					MODE6				MODE5				MODE4				MODE3				MODE2				MODE1				MODE0			

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:24	MODE6 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFILTER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUPFILTER	Open-drain output using alternate control with filter and pullup.

23:20	MODE5 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.

Bit	Name	Reset	Access	Description
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
19:16	MODE4 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
15:12	MODE3 MODE n	0x0	RW	**MODE n**

Bit	Name	Reset	Access	Description

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

11:8	MODE2 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.

Bit	Name	Reset	Access	Description
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
7:4	MODE1 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
3:0	MODE0 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.

Bit	Name	Reset	Access	Description
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFILTER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUPFILTER		Open-drain output using alternate control with filter and pullup.

23.6.5 GPIO_PORTA_DOUT - Data Out

Offset	Bit Position																													
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									DOUT							

Bit	Name	Reset	Access	Description
31:15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14:0	DOUT Data output	0x0	RW	**Data output**

23.6.6 GPIO_PORTA_DIN - Data in

Offset	Bit Position																													
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									DIN							

Bit	Name	Reset	Access	Description
31:15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14:0	DIN Data input	0x0	R	**Data input**

23.6.7 GPIO_PORTB_CTRL - Port Control

Offset	Bit Position																													
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x0							0x4									0x0							0x4					
Access				RW							RW									RW							RW					
Name				DINDISALT							SLEWRATEALT									DINDIS							SLEWRATE					

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28	DINDISALT	0x0	RW	**Data In Disable Alt**
	Data input disable for port pins using alternate modes.			
27:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:20	SLEWRATEALT	0x4	RW	**Slew Rate Alt**
	Slewrate limit for port pins using alternate modes. Higher values provide faster slewrates.			
19:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	DINDIS	0x0	RW	**Data In Disable**
	Data input disable for port pins not using alternate modes.			
11:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	SLEWRATE	0x4	RW	**Slew Rate**
	Slewrate limit for port pins using not alternate modes. Higher values provide faster slewrates.			
3:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

23.6.8 GPIO_PORTB_MODEL - Mode Low

Offset	Bit Position																													
0x064	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0				0x0				0x0				0x0				0x0				0x0				0x0				0x0			
Access	RW				RW				RW				RW				RW				RW				RW				RW			
Name	MODE7				MODE6				MODE5				MODE4				MODE3				MODE2				MODE1				MODE0			

Bit	Name	Reset	Access	Description
31:28	MODE7	0x0	RW	**MODE n**
	MODE n			

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

Bit	Name	Reset	Access	Description
27:24	MODE6	0x0	RW	**MODE n**
	MODE n			

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.

Bit	Name	Reset	Access	Description
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
23:20	MODE5 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
19:16	MODE4 MODE n	0x0	RW	**MODE n**

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
15:12	MODE3 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.

Bit	Name	Reset	Access	Description
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
11:8	MODE2 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

7:4	MODE1 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.

Bit	Name	Reset	Access	Description
	11	WIREDANDPULLUPFILTER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUPFILTER		Open-drain output using alternate control with filter and pullup.
3:0	MODE0 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFILTER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUPFILTER	Open-drain output using alternate control with filter and pullup.

23.6.9 GPIO_PORTB_DOUT - Data Out

Offset	Bit Position																													
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									DOUT							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	DOUT	0x0	RW	**Data output**
	Data output			

23.6.10 GPIO_PORTB_DIN - Data in

Offset	Bit Position																													
0x074	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									DIN							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	DIN	0x0	R	**Data input**
	Data input			

23.6.11 GPIO_PORTC_CTRL - Port Control

Offset	Bit Position																													
0x090	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x0							0x4									0x0							0x4					
Access				RW							RW									RW							RW					
Name				DINDISALT							SLEWRATEALT									DINDIS							SLEWRATE					

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28	DINDISALT	0x0	RW	**Data In Disable Alt**
	Data input disable for port pins using alternate modes.			
27:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:20	SLEWRATEALT	0x4	RW	**Slew Rate Alt**
	Slewrate limit for port pins using alternate modes. Higher values provide faster slewrates.			
19:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	DINDIS	0x0	RW	**Data In Disable**
	Data input disable for port pins not using alternate modes.			
11:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	SLEWRATE	0x4	RW	**Slew Rate**
	Slewrate limit for port pins using not alternate modes. Higher values provide faster slewrates.			
3:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

23.6.12 GPIO_PORTC_MODEL - Mode Low

Offset	Bit Position																													
0x094	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0				0x0				0x0				0x0				0x0				0x0				0x0				0x0			
Access	RW				RW				RW				RW				RW				RW				RW				RW			
Name	MODE7				MODE6				MODE5				MODE4				MODE3				MODE2				MODE1				MODE0			

Bit	Name	Reset	Access	Description
31:28	MODE7	0x0	RW	**MODE n**
	MODE n			

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

Bit	Name	Reset	Access	Description
27:24	MODE6	0x0	RW	**MODE n**
	MODE n			

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.

Bit	Name	Reset	Access	Description
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
23:20	MODE5 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
19:16	MODE4 MODE n	0x0	RW	**MODE n**

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
15:12	MODE3 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.

Bit	Name	Reset	Access	Description
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
11:8	MODE2 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

7:4	MODE1 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.

Bit	Name	Reset	Access	Description
	11	WIREDANDPULLUPFILTER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUPFILTER		Open-drain output using alternate control with filter and pullup.
3:0	MODE0 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFILTER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUPFILTER	Open-drain output using alternate control with filter and pullup.

23.6.13 GPIO_PORTC_MODEH - Mode High

Offset	Bit Position																													
0x09C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0				0x0				0x0				0x0			
Access																	RW				RW				RW				RW			
Name																	MODE3				MODE2				MODE1				MODE0			

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:12	MODE3 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFILTER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUPFILTER	Open-drain output using alternate control with filter and pullup.

Bit	Name	Reset	Access	Description
11:8	MODE2 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.

Bit	Name	Reset	Access	Description
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
7:4	MODE1 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
3:0	MODE0 MODE n	0x0	RW	**MODE n**

Bit	Name	Reset	Access	Description

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

23.6.14 GPIO_PORTC_DOUT - Data Out

Offset	Bit Position																															
0x0A0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0					
Access																											RW					
Name																											DOUT					

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:0	DOUT Data output	0x0	RW	**Data output**

23.6.15 GPIO_PORTC_DIN - Data in

Offset	Bit Position																															
0x0A4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0											
Access																					R											
Name																					DIN											

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:0	DIN Data input	0x0	R	**Data input**

23.6.16 GPIO_PORTD_CTRL - Port Control

Offset	Bit Position																													
0x0C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x0							0x4									0x0							0x4					
Access				RW							RW									RW							RW					
Name				DINDISALT							SLEWRATEALT									DINDIS							SLEWRATE					

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28	DINDISALT	0x0	RW	**Data In Disable Alt**
	Data input disable for port pins using alternate modes.			
27:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:20	SLEWRATEALT	0x4	RW	**Slew Rate Alt**
	Slewrate limit for port pins using alternate modes. Higher values provide faster slewrates.			
19:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	DINDIS	0x0	RW	**Data In Disable**
	Data input disable for port pins not using alternate modes.			
11:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	SLEWRATE	0x4	RW	**Slew Rate**
	Slewrate limit for port pins using not alternate modes. Higher values provide faster slewrates.			
3:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

23.6.17 GPIO_PORTD_MODEL - Mode Low

Offset	Bit Position																													
0x0C4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0				0x0				0x0				0x0				0x0				0x0				0x0				0x0			
Access	RW				RW				RW				RW				RW				RW				RW				RW			
Name	MODE7				MODE6				MODE5				MODE4				MODE3				MODE2				MODE1				MODE0			

Bit	Name	Reset	Access	Description
31:28	MODE7	0x0	RW	**MODE n**
	MODE n			

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

Bit	Name	Reset	Access	Description
27:24	MODE6	0x0	RW	**MODE n**
	MODE n			

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.

Bit	Name	Reset	Access	Description
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
23:20	MODE5 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
19:16	MODE4 MODE n	0x0	RW	**MODE n**

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
15:12	MODE3 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.

Bit	Name	Reset	Access	Description
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
11:8	MODE2 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

7:4	MODE1 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.

Bit	Name	Reset	Access	Description
	11	WIREDANDPULLUPFILTER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUPFILTER		Open-drain output using alternate control with filter and pullup.
3:0	MODE0 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFILTER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUPFILTER	Open-drain output using alternate control with filter and pullup.

23.6.18 GPIO_PORTD_MODEH - Mode High

Offset	Bit Position																													
0x0CC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0				0x0				0x0				0x0				0x0				0x0				0x0				0x0			
Access	RW				RW				RW				RW				RW				RW				RW				RW			
Name	MODE7				MODE6				MODE5				MODE4				MODE3				MODE2				MODE1				MODE0			

Bit	Name	Reset	Access	Description
31:28	MODE7	0x0	RW	**MODE n**
	MODE n			

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

Bit	Name	Reset	Access	Description
27:24	MODE6	0x0	RW	**MODE n**
	MODE n			

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.

Bit	Name	Reset	Access	Description
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
23:20	MODE5 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
19:16	MODE4 MODE n	0x0	RW	**MODE n**

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
15:12	MODE3 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.

Bit	Name	Reset	Access	Description
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
11:8	MODE2 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

7:4	MODE1 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.

Bit	Name	Reset	Access	Description
	11	WIREDANDPULLUPFILTER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUPFILTER		Open-drain output using alternate control with filter and pullup.
3:0	MODE0 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFILTER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUPFILTER	Open-drain output using alternate control with filter and pullup.

23.6.19 GPIO_PORTD_DOUT - Data Out

Offset	Bit Position																													
0x0D0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	RW															
Name																	DOUT															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	DOUT	0x0	RW	**Data output**
	Data output			

23.6.20 GPIO_PORTD_DIN - Data in

Offset	Bit Position																													
0x0D4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	R															
Name																	DIN															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	DIN	0x0	R	**Data input**
	Data input			

23.6.21 GPIO_LOCK - Lock Register

Offset	Bit Position																													
0x300	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0xA534								
Access																								W								
Name																								LOCKKEY								

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0xA534	W	**Configuration Lock Key**

Write any other value than the unlock code to lock configuration registers. Write the unlock code to unlock (See text for detailed list of configuration registers.)

Value	Mode	Description
42292	UNLOCK	Unlock code

23.6.22 GPIO_GPIOLOCKSTATUS - Lock Status

Offset	Bit Position																													
0x310	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																R
Name																																LOCK

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	LOCK	0x0	R	**GPIO LOCK status**

Indicates current lock status of GPIO registers

Value	Mode	Description
0	UNLOCKED	Registers are unlocked
1	LOCKED	Registers are locked

23.6.23 GPIO_ABUSALLOC - A Bus Allocation

Offset	Bit Position																													
0x320	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0								0x0								0x0								0x0			
Access					RW								RW								RW								RW			
Name					AODD1								AODD0								AEVEN1								AEVEN0			

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:24	AODD1	0x0	RW	**A Bus Odd 1**

peripheral allocation to A Bus Odd 1

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1
4	VDAC0CH1	The bus is allocated to VDAC0 CH1

23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	AODD0	0x0	RW	**A Bus Odd 0**

peripheral allocation to A Bus Odd 0

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1
4	VDAC0CH0	The bus is allocated to VDAC0 CH0

15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:8	AEVEN1	0x0	RW	**A Bus Even 1**

peripheral allocation to A Bus Even 1

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0

Bit	Name	Reset	Access	Description
	3	ACMP1		The bus is allocated to ACMP1
	4	VDAC0CH1		The bus is allocated to VDAC0 CH1
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	AEVEN0	0x0	RW	**A Bus Even 0**

peripheral allocation to A Bus Even 0

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1
4	VDAC0CH0	The bus is allocated to VDAC0 CH0

23.6.24 GPIO_BBUSALLOC - B Bus Allocation

Offset	Bit Position																													
0x324	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0								0x0								0x0								0x0			
Access					RW								RW								RW								RW			
Name					BODD1								BODD0								BEVEN1								BEVEN0			

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:24	BODD1	0x0	RW	**B Bus Odd 1**

peripheral allocation to B Bus Odd 1

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1
4	VDAC0CH1	The bus is allocated to VDAC0 CH1

23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	BODD0	0x0	RW	**B Bus Odd 0**

peripheral allocation to B Bus Odd 0

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1
4	VDAC0CH0	The bus is allocated to VDAC0 CH0

15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:8	BEVEN1	0x0	RW	**B Bus Even 1**

peripheral allocation to B Bus Even 1

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0

Bit	Name	Reset	Access	Description
	3	ACMP1		The bus is allocated to ACMP1
	4	VDAC0CH1		The bus is allocated to VDAC0 CH1
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	BEVEN0	0x0	RW	**B Bus Even 0**

peripheral allocation to B Bus Even 0

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1
4	VDAC0CH0	The bus is allocated to VDAC0 CH0

23.6.25 GPIO_CDBUSALLOC - CD Bus Allocation

Offset	Bit Position																													
0x328	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0								0x0								0x0								0x0			
Access					RW								RW								RW								RW			
Name					CDODD1								CDODD0								CDEVEN1								CDEVEN0			

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:24	CDODD1	0x0	RW	**CD Bus Odd 1**
	peripheral allocation to CD Bus Odd 1			

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1
4	VDAC0CH1	The bus is allocated to VDAC0 CH1

23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	CDODD0	0x0	RW	**CD Bus Odd 0**
	peripheral allocation to CD Bus Odd 0			

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1
4	VDAC0CH0	The bus is allocated to VDAC0 CH0

15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:8	CDEVEN1	0x0	RW	**CD Bus Even 1**
	peripheral allocation to CD Bus Even 1			

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0

Bit	Name	Reset	Access	Description
	3	ACMP1		The bus is allocated to ACMP1
	4	VDAC0CH1		The bus is allocated to VDAC0 CH1
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	CDEVEN0	0x0	RW	**CD Bus Even 0**

peripheral allocation to CD Bus Even 0

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1
4	VDAC0CH0	The bus is allocated to VDAC0 CH0

23.6.26 GPIO_EXTIPSELL - External Interrupt Port Select Low

Offset	Bit Position																											
0x400	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset			0x0				0x0				0x0				0x0				0x0				0x0				0x0				0x0	
Access			RW				RW				RW				RW				RW				RW				RW				RW	
Name			EXTIPSEL7				EXTIPSEL6				EXTIPSEL5				EXTIPSEL4				EXTIPSEL3				EXTIPSEL2				EXTIPSEL1				EXTIPSEL0	

Bit	Name	Reset	Access	Description
31:30	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
29:28	EXTIPSEL7	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 7 (EXTI7)			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
27:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25:24	EXTIPSEL6	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 6 (EXTI6)			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
23:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:20	EXTIPSEL5	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 5 (EXTI5)			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected

Bit	Name	Reset	Access	Description
19:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17:16	EXTIPSEL4	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 4 (EXTI4)			

Value	Mode	Description
0	PORTA	Port A group selected
1	PORTB	Port B group selected
2	PORTC	Port C group selected
3	PORTD	Port D group selected

Bit	Name	Reset	Access	Description
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:12	EXTIPSEL3	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 3 (EXTI3)			

Value	Mode	Description
0	PORTA	Port A group selected
1	PORTB	Port B group selected
2	PORTC	Port C group selected
3	PORTD	Port D group selected

Bit	Name	Reset	Access	Description
11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	EXTIPSEL2	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 2 (EXTI2)			

Value	Mode	Description
0	PORTA	Port A group selected
1	PORTB	Port B group selected
2	PORTC	Port C group selected
3	PORTD	Port D group selected

Bit	Name	Reset	Access	Description
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	EXTIPSEL1	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 1 (EXTI1)			

Value	Mode	Description
0	PORTA	Port A group selected
1	PORTB	Port B group selected
2	PORTC	Port C group selected
3	PORTD	Port D group selected

Bit	Name	Reset	Access	Description
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	EXTIPSEL0	0x0	RW	**External Interrupt Port Select** Port select for external interrupt 0 (EXTI0)

Value	Mode	Description
0	PORTA	Port A group selected
1	PORTB	Port B group selected
2	PORTC	Port C group selected
3	PORTD	Port D group selected

23.6.27 GPIO_EXTIPSELH - External Interrupt Port Select High

Offset	Bit Position																											
0x404	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset			0x0				0x0				0x0				0x0				0x0				0x0				0x0				0x0	
Access			RW				RW				RW				RW				RW				RW				RW				RW	
Name			EXTIPSEL7				EXTIPSEL6				EXTIPSEL5				EXTIPSEL4				EXTIPSEL3				EXTIPSEL2				EXTIPSEL1				EXTIPSEL0	

Bit	Name	Reset	Access	Description
31:30	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
29:28	EXTIPSEL7	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 7+8			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
27:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25:24	EXTIPSEL6	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 6+8			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
23:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:20	EXTIPSEL5	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 5+8			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected

Bit	Name	Reset	Access	Description
19:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17:16	EXTIPSEL4	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 4+8			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:12	EXTIPSEL3	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 3+8			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	EXTIPSEL2	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 2+8			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	EXTIPSEL1	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 1+8			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected

Bit	Name	Reset	Access	Description
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	EXTIPSEL0	0x0	RW	**External Interrupt Port Select** Port select for external interrupt 0+8

Value	Mode	Description
0	PORTA	Port A group selected
1	PORTB	Port B group selected
2	PORTC	Port C group selected
3	PORTD	Port D group selected

23.6.28 GPIO_EXTIPINSELL - External Interrupt Pin Select Low

Offset	Bit Position																											
0x408	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset			0x0				0x0				0x0				0x0				0x0				0x0				0x0				0x0	
Access			RW				RW				RW				RW				RW				RW				RW				RW	
Name			EXTIPINSEL7				EXTIPINSEL6				EXTIPINSEL5				EXTIPINSEL4				EXTIPINSEL3				EXTIPINSEL2				EXTIPINSEL1				EXTIPINSEL0	

Bit	Name	Reset	Access	Description
31:30	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
29:28	EXTIPINSEL7	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt 7 (EXTI7)			
	Value	Mode		Description
	0	PIN0		OFFSET=0
	1	PIN1		OFFSET=1
	2	PIN2		OFFSET=2
	3	PIN3		OFFSET=3
27:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25:24	EXTIPINSEL6	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt 6 (EXTI6)			
	Value	Mode		Description
	0	PIN0		OFFSET=0
	1	PIN1		OFFSET=1
	2	PIN2		OFFSET=2
	3	PIN3		OFFSET=3
23:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:20	EXTIPINSEL5	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt 5 (EXTI5)			
	Value	Mode		Description
	0	PIN0		OFFSET=0
	1	PIN1		OFFSET=1
	2	PIN2		OFFSET=2
	3	PIN3		OFFSET=3

Bit	Name	Reset	Access	Description
19:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17:16	EXTIPINSEL4	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt 4 (EXTI4)			

Value	Mode	Description
0	PIN0	OFFSET=0
1	PIN1	OFFSET=1
2	PIN2	OFFSET=2
3	PIN3	OFFSET=3

Bit	Name	Reset	Access	Description
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:12	EXTIPINSEL3	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt 3 (EXTI3)			

Value	Mode	Description
0	PIN0	OFFSET=0
1	PIN1	OFFSET=1
2	PIN2	OFFSET=2
3	PIN3	OFFSET=3

Bit	Name	Reset	Access	Description
11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	EXTIPINSEL2	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt 2 (EXTI2)			

Value	Mode	Description
0	PIN0	OFFSET=0
1	PIN1	OFFSET=1
2	PIN2	OFFSET=2
3	PIN3	OFFSET=3

Bit	Name	Reset	Access	Description
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	EXTIPINSEL1	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt 1 (EXTI1)			

Value	Mode	Description
0	PIN0	OFFSET=0
1	PIN1	OFFSET=1
2	PIN2	OFFSET=2
3	PIN3	OFFSET=3

Bit	Name	Reset	Access	Description
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	EXTIPINSEL0	0x0	RW	**External Interrupt Pin select**

OFFSET select for External Interrupt 0 (EXTI0)

Value	Mode	Description
0	PIN0	OFFSET=0
1	PIN1	OFFSET=1
2	PIN2	OFFSET=2
3	PIN3	OFFSET=3

23.6.29 GPIO_EXTIPINSELH - External Interrupt Pin Select High

Offset	Bit Position																											
0x40C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset			0x0				0x0				0x0				0x0				0x0				0x0				0x0				0x0	
Access			RW				RW				RW				RW				RW				RW				RW				RW	
Name			EXTIPINSEL7				EXTIPINSEL6				EXTIPINSEL5				EXTIPINSEL4				EXTIPINSEL3				EXTIPINSEL2				EXTIPINSEL1				EXTIPINSEL0	

Bit	Name	Reset	Access	Description
31:30	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
29:28	EXTIPINSEL7	0x0	RW	**External Interrupt Pin select**

OFFSET select for External Interrupt {b+8} (EXTI{b+8})

Value	Mode	Description
0	PIN8	OFFSET=8
1	PIN9	OFFSET=9
2	PIN10	OFFSET=10
3	PIN11	OFFSET=11

Bit	Name	Reset	Access	Description
27:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25:24	EXTIPINSEL6	0x0	RW	**External Interrupt Pin select**

OFFSET select for External Interrupt {b+8} (EXTI{b+8})

Value	Mode	Description
0	PIN8	OFFSET=8
1	PIN9	OFFSET=9
2	PIN10	OFFSET=10
3	PIN11	OFFSET=11

Bit	Name	Reset	Access	Description
23:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:20	EXTIPINSEL5	0x0	RW	**External Interrupt Pin select**

OFFSET select for External Interrupt {b+8} (EXTI{b+8})

Value	Mode	Description
0	PIN8	OFFSET=8
1	PIN9	OFFSET=9
2	PIN10	OFFSET=10
3	PIN11	OFFSET=11

Bit	Name	Reset	Access	Description
19:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17:16	EXTIPINSEL4	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt {b+8} (EXTI{b+8})			

Value	Mode	Description
0	PIN8	OFFSET=8
1	PIN9	OFFSET=9
2	PIN10	OFFSET=10
3	PIN11	OFFSET=11

Bit	Name	Reset	Access	Description
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:12	EXTIPINSEL3	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt {b+8} (EXTI{b+8})			

Value	Mode	Description
0	PIN8	OFFSET=8
1	PIN9	OFFSET=9
2	PIN10	OFFSET=10
3	PIN11	OFFSET=11

Bit	Name	Reset	Access	Description
11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	EXTIPINSEL2	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt {b+8} (EXTI{b+8})			

Value	Mode	Description
0	PIN8	OFFSET=8
1	PIN9	OFFSET=9
2	PIN10	OFFSET=10
3	PIN11	OFFSET=11

Bit	Name	Reset	Access	Description
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	EXTIPINSEL1	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt {b+8} (EXTI{b+8})			

Value	Mode	Description
0	PIN8	OFFSET=8
1	PIN9	OFFSET=9
2	PIN10	OFFSET=10
3	PIN11	OFFSET=11

Bit	Name	Reset	Access	Description
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	EXTIPINSEL0	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt {b+8} (EXTI{b+8})			

Value	Mode	Description
0	PIN8	OFFSET=8
1	PIN9	OFFSET=9
2	PIN10	OFFSET=10
3	PIN11	OFFSET=11

23.6.30 GPIO_EXTIRISE - External Interrupt Rising Edge Trigger

Offset	Bit Position																													
0x410	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	RW															
Name																	EXTIRISE															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	EXTIRISE	0x0	RW	**EXT Int Rise**
	External Interrupt n Rising Edge Trigger Enable			

23.6.31 GPIO_EXTIFALL - External Interrupt Falling Edge Trigger

Offset	Bit Position																													
0x414	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									EXTIFALL							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	EXTIFALL	0x0	RW	**EXT Int FALL**
	External Interrupt n Falling Edge Trigger Enable			

23.6.32 GPIO_IF - Interrupt Flag

Offset	Bit Position																													
0x420	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0												0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access					RW												RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name					EM4WU												EXTIF15	EXTIF14	EXTIF13	EXTIF12	EXTIF11	EXTIF10	EXTIF9	EXTIF8	EXTIF7	EXTIF6	EXTIF5	EXTIF4	EXTIF3	EXTIF2	EXTIF1	EXTIF0

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:16	EM4WU	0x0	RW	**EM4 wake up**
15	EXTIF15 External Pin interrupt flag	0x0	RW	**External Pin Flag**
14	EXTIF14 External Pin interrupt flag	0x0	RW	**External Pin Flag**
13	EXTIF13 External Pin interrupt flag	0x0	RW	**External Pin Flag**
12	EXTIF12 External Pin interrupt flag	0x0	RW	**External Pin Flag**
11	EXTIF11 External Pin interrupt flag	0x0	RW	**External Pin Flag**
10	EXTIF10 External Pin interrupt flag	0x0	RW	**External Pin Flag**
9	EXTIF9 External Pin interrupt flag	0x0	RW	**External Pin Flag**
8	EXTIF8 External Pin interrupt flag	0x0	RW	**External Pin Flag**
7	EXTIF7 External Pin interrupt flag	0x0	RW	**External Pin Flag**
6	EXTIF6 External Pin interrupt flag	0x0	RW	**External Pin Flag**
5	EXTIF5 External Pin interrupt flag	0x0	RW	**External Pin Flag**
4	EXTIF4 External Pin interrupt flag	0x0	RW	**External Pin Flag**
3	EXTIF3	0x0	RW	**External Pin Flag**

Bit	Name	Reset	Access	Description
	External Pin interrupt flag			
2	EXTIF2	0x0	RW	**External Pin Flag**
	External Pin interrupt flag			
1	EXTIF1	0x0	RW	**External Pin Flag**
	External Pin interrupt flag			
0	EXTIF0	0x0	RW	**External Pin Flag**
	External Pin interrupt flag			

23.6.33 GPIO_IEN - Interrupt Enable

Offset	Bit Position																															
0x424	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name					EM4WUIEN11	EM4WUIEN10	EM4WUIEN9	EM4WUIEN8	EM4WUIEN7	EM4WUIEN6	EM4WUIEN5	EM4WUIEN4	EM4WUIEN3	EM4WUIEN2	EM4WUIEN1	EM4WUIEN0	EXTIEN15	EXTIEN14	EXTIEN13	EXTIEN12	EXTIEN11	EXTIEN10	EXTIEN9	EXTIEN8	EXTIEN7	EXTIEN6	EXTIEN5	EXTIEN4	EXTIEN3	EXTIEN2	EXTIEN1	EXTIEN0

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27	EM4WUIEN11	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
26	EM4WUIEN10	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
25	EM4WUIEN9	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
24	EM4WUIEN8	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
23	EM4WUIEN7	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
22	EM4WUIEN6	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
21	EM4WUIEN5	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
20	EM4WUIEN4	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
19	EM4WUIEN3	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
18	EM4WUIEN2	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
17	EM4WUIEN1	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
16	EM4WUIEN0	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
15	EXTIEN15	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			

Bit	Name	Reset	Access	Description
14	EXTIEN14	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
13	EXTIEN13	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
12	EXTIEN12	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
11	EXTIEN11	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
10	EXTIEN10	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
9	EXTIEN9	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
8	EXTIEN8	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
7	EXTIEN7	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
6	EXTIEN6	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
5	EXTIEN5	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
4	EXTIEN4	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
3	EXTIEN3	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
2	EXTIEN2	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
1	EXTIEN1	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
0	EXTIEN0	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			

23.6.34 GPIO_EM4WUEN - EM4 Wakeup Enable

Offset	Bit Position																													
0x42C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0																											
Access					RW																											
Name					EM4WUEN																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:16	EM4WUEN	0x0	RW	**EM4 wake up enable**
	Write 1 to enable EM4 wake up request, write 0 to disable EM4 wake up request			
15:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

23.6.35 GPIO_EM4WUPOL - EM4 Wakeup Polarity

Offset	Bit Position																													
0x430	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0																											
Access					RW																											
Name					EM4WUPOL																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:16	EM4WUPOL	0x0	RW	**EM4 Wake-Up Polarity**
	EM4 Wakeup Polarity			
15:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

23.6.36 GPIO_DBGROUTEPEN - Debugger Route Pin Enable

Offset	Bit Position																															
0x440	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x1	0x1	0x1	0x1
Access																													RW	RW	RW	RW
Name																													TDIPEN	TDOPEN	SWDIOTMSPEN	SWCLKTCKPEN

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	TDIPEN	0x1	RW	**JTAG Test Debug Input Pin Enable**
	Enable JTAG TDI connection to pin.			
2	TDOPEN	0x1	RW	**JTAG Test Debug Output Pin Enable**
	Enable JTAG TDO connection to pin.			
1	SWDIOTMSPEN	0x1	RW	**Route Location 0**
	Enable Serial Wire Data and JTAG Test Mode Select connection to pin. WARNING: When the pin is disabled, the device can no longer be accessed by a debugger. A reset will set the pin back to a default state as enabled. If you disable this pin, make sure you have at least a 3 second timeout at the start of your program code before you disable the pin. This way, the debugger will have time to halt the device after a reset before the pin is disabled.			
0	SWCLKTCKPEN	0x1	RW	**Route Pin Enable**
	Enable Serial Wire and JTAG CLock connection to pin. WARNING: When the pin is disabled, the device can no longer be accessed by a debugger. A reset will set the pin back to a default state as enabled. If you disable this pin, make sure you have at least a 3 second timeout at the start of your program code before you disable the pin. This way, the debugger will have time to halt the device after a reset before the pin is disabled.			

23.6.37 GPIO_TRACEROUTEPEN - Trace Route Pin Enable

Offset	Bit Position																													
0x444	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											RW	RW	RW	RW	RW	RW
Name																											TRACEDATA3PEN	TRACEDATA2PEN	TRACEDATA1PEN	TRACEDATA0PEN	TRACECLKPEN	SWVPEN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	TRACEDATA3PEN	0x0	RW	**Trace Data3 Pin Enable**
	Trace Data3 Pin Enable			
4	TRACEDATA2PEN	0x0	RW	**Trace Data2 Pin Enable**
	Trace Data2 Pin Enable			
3	TRACEDATA1PEN	0x0	RW	**Trace Data1 Pin Enable**
	Trace Data1 Pin Enable			
2	TRACEDATA0PEN	0x0	RW	**Trace Data0 Pin Enable**
	Trace Data0 Pin Enable			
1	TRACECLKPEN	0x0	RW	**Trace Clk Pin Enable**
	Trace Clk Pin Enable			
0	SWVPEN	0x0	RW	**Serial Wire Viewer Output Pin Enable**
	Serial Wire Viewer Output Pin Enable			

23.6.38 GPIO_LCDSEG - LCD Segment Enable

Offset	Bit Position																													
0x460	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0																											
Access					RW																											
Name					LCDSEGALLOC																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:0	LCDSEGALLOC	0x0	RW	**LCD Segment Allocation**
	Enables individual LCD_SEGx pins. Bit 0 enables SEG0, bit 1 enables SEG1, etc.			

23.6.39 GPIO_LCDCOM - LCD Common Enable

Offset	Bit Position																													
0x470	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													RW		
Name																													LCDCOMALLOC		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	LCDCOMALLOC	0x0	RW	**LCD Common Allocation**
	Enables individual LCD_COMx pins. Bit 0 enables COM0, bit 1 enables COM1, etc.			

23.6.40 GPIO_ACMP0_ROUTEEN - ACMP0 Pin Enable

Offset	Bit Position																														
0x480	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																ACMPOUTPEN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	ACMPOUTPEN	0x0	RW	**ACMPOUT pin enable control bit**
	ACMPOUT pin enable control bit			

23.6.41 GPIO_ACMP0_ACMPOUTROUTE - ACMPOUT Port/Pin Select

Offset	Bit Position																														
0x484	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset														0x0																	0x0	
Access														RW																	RW	
Name														PIN																	PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ACMPOUT pin select register**
	ACMPOUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ACMPOUT port select register**
	ACMPOUT port select register			

23.6.42 GPIO_ACMP1_ROUTEEN - ACMP1 Pin Enable

Offset	Bit Position																														
0x48C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																ACMPOUTPEN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	ACMPOUTPEN	0x0	RW	**ACMPOUT pin enable control bit**
	ACMPOUT pin enable control bit			

23.6.43 GPIO_ACMP1_ACMPOUTROUTE - ACMPOUT Port/Pin Select

Offset	Bit Position																														
0x490	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset														0x0																		0x0
Access														RW																		RW
Name														PIN																		PORT

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ACMPOUT pin select register**
	ACMPOUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ACMPOUT port select register**
	ACMPOUT port select register			

23.6.44 GPIO_CMU_ROUTEEN - CMU Pin Enable

Offset	Bit Position																														
0x498	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														CLKOUT2PEN	CLKOUT1PEN	CLKOUT0PEN

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	CLKOUT2PEN	0x0	RW	**CLKOUT2 pin enable control bit**
	CLKOUT2 pin enable control bit			
1	CLKOUT1PEN	0x0	RW	**CLKOUT1 pin enable control bit**
	CLKOUT1 pin enable control bit			
0	CLKOUT0PEN	0x0	RW	**CLKOUT0 pin enable control bit**
	CLKOUT0 pin enable control bit			

23.6.45 GPIO_CMU_CLKIN0ROUTE - CLKIN0 Port/Pin Select

Offset	Bit Position																														
0x49C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset														0x0																	0x0	
Access														RW																	RW	
Name														PIN																	PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CLKIN0 pin select register**
	CLKIN0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CLKIN0 port select register**
	CLKIN0 port select register			

23.6.46 GPIO_CMU_CLKOUT0ROUTE - CLKOUT0 Port/Pin Select

Offset	Bit Position																													
0x4A0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CLKOUT0 pin select register**
	CLKOUT0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CLKOUT0 port select register**
	CLKOUT0 port select register			

23.6.47 GPIO_CMU_CLKOUT1ROUTE - CLKOUT1 Port/Pin Select

Offset	Bit Position																													
0x4A4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CLKOUT1 pin select register**
	CLKOUT1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CLKOUT1 port select register**
	CLKOUT1 port select register			

23.6.48 GPIO_CMU_CLKOUT2ROUTE - CLKOUT2 Port/Pin Select

Offset	Bit Position																													
0x4A8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CLKOUT2 pin select register**
	CLKOUT2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CLKOUT2 port select register**
	CLKOUT2 port select register			

23.6.49 GPIO_EUSART0_ROUTEEN - EUSART0 Pin Enable

Offset	Bit Position																													
0x4C4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												TXPEN	SCLKPEN	RXPEN	RTSPEN	CSPEN

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	TXPEN	0x0	RW	**TX pin enable control bit**
	TX pin enable control bit			
3	SCLKPEN	0x0	RW	**SCLK pin enable control bit**
	SCLK pin enable control bit			
2	RXPEN	0x0	RW	**RX pin enable control bit**
	RX pin enable control bit			
1	RTSPEN	0x0	RW	**RTS pin enable control bit**
	RTS pin enable control bit			
0	CSPEN	0x0	RW	**CS pin enable control bit**
	CS pin enable control bit			

23.6.50 GPIO_EUSART0_CSROUTE - CS Port/Pin Select

Offset	Bit Position																													
0x4C8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CS pin select register**
	CS pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CS port select register**
	CS port select register			

23.6.51 GPIO_EUSART0_CTSROUTE - CTS Port/Pin Select

Offset	Bit Position																													
0x4CC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CTS pin select register**
	CTS pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CTS port select register**
	CTS port select register			

23.6.52 GPIO_EUSART0_RTSROUTE - RTS Port/Pin Select

Offset	Bit Position																													
0x4D0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**RTS pin select register**
	RTS pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**RTS port select register**
	RTS port select register			

23.6.53 GPIO_EUSART0_RXROUTE - RX Port/Pin Select

Offset	Bit Position																													
0x4D4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**RX pin select register**
	RX pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**RX port select register**
	RX port select register			

23.6.54 GPIO_EUSART0_SCLKROUTE - SCLK Port/Pin Select

Offset	Bit Position																													
0x4D8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**SCLK pin select register**
	SCLK pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**SCLK port select register**
	SCLK port select register			

23.6.55 GPIO_EUSART0_TXROUTE - TX Port/Pin Select

Offset	Bit Position																													
0x4DC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**TX pin select register**
	TX pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**TX port select register**
	TX port select register			

23.6.56 GPIO_EUSART1_ROUTEEN - EUSART1 Pin Enable

Offset	Bit Position																													
0x4E4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												TXPEN	SCLKPEN	RXPEN	RTSPEN	CSPEN

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	TXPEN	0x0	RW	**TX pin enable control bit**
	TX pin enable control bit			
3	SCLKPEN	0x0	RW	**SCLK pin enable control bit**
	SCLK pin enable control bit			
2	RXPEN	0x0	RW	**RX pin enable control bit**
	RX pin enable control bit			
1	RTSPEN	0x0	RW	**RTS pin enable control bit**
	RTS pin enable control bit			
0	CSPEN	0x0	RW	**CS pin enable control bit**
	CS pin enable control bit			

23.6.57 GPIO_EUSART1_CSROUTE - CS Port/Pin Select

Offset	Bit Position																													
0x4E8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CS pin select register**
	CS pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CS port select register**
	CS port select register			

23.6.58 GPIO_EUSART1_CTSROUTE - CTS Port/Pin Select

Offset	Bit Position																													
0x4EC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CTS pin select register**
	CTS pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CTS port select register**
	CTS port select register			

23.6.59 GPIO_EUSART1_RTSROUTE - RTS Port/Pin Select

Offset	Bit Position																													
0x4F0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**RTS pin select register**
	RTS pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**RTS port select register**
	RTS port select register			

23.6.60 GPIO_EUSART1_RXROUTE - RX Port/Pin Select

Offset	Bit Position																													
0x4F4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**RX pin select register**
	RX pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**RX port select register**
	RX port select register			

23.6.61 GPIO_EUSART1_SCLKROUTE - SCLK Port/Pin Select

Offset	Bit Position																													
0x4F8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**SCLK pin select register**
	SCLK pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**SCLK port select register**
	SCLK port select register			

23.6.62 GPIO_EUSART1_TXROUTE - TX Port/Pin Select

Offset	Bit Position																													
0x4FC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**TX pin select register**
	TX pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**TX port select register**
	TX port select register			

23.6.63 GPIO_EUSART2_ROUTEEN - EUSART2 Pin Enable

Offset	Bit Position																													
0x504	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												TXPEN	SCLKPEN	RXPEN	RTSPEN	CSPEN

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	TXPEN	0x0	RW	**TX pin enable control bit**
	TX pin enable control bit			
3	SCLKPEN	0x0	RW	**SCLK pin enable control bit**
	SCLK pin enable control bit			
2	RXPEN	0x0	RW	**RX pin enable control bit**
	RX pin enable control bit			
1	RTSPEN	0x0	RW	**RTS pin enable control bit**
	RTS pin enable control bit			
0	CSPEN	0x0	RW	**CS pin enable control bit**
	CS pin enable control bit			

23.6.64 GPIO_EUSART2_CSROUTE - CS Port/Pin Select

Offset	Bit Position																													
0x508	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CS pin select register**
	CS pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CS port select register**
	CS port select register			

23.6.65 GPIO_EUSART2_CTSROUTE - CTS Port/Pin Select

Offset	Bit Position																													
0x50C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CTS pin select register**
	CTS pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CTS port select register**
	CTS port select register			

23.6.66 GPIO_EUSART2_RTSROUTE - RTS Port/Pin Select

Offset	Bit Position																													
0x510	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**RTS pin select register**
	RTS pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**RTS port select register**
	RTS port select register			

23.6.67 GPIO_EUSART2_RXROUTE - RX Port/Pin Select

Offset	Bit Position																													
0x514	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**RX pin select register**
	RX pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**RX port select register**
	RX port select register			

23.6.68 GPIO_EUSART2_SCLKROUTE - SCLK Port/Pin Select

Offset	Bit Position																													
0x518	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**SCLK pin select register**
	SCLK pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**SCLK port select register**
	SCLK port select register			

23.6.69 GPIO_EUSART2_TXROUTE - TX Port/Pin Select

Offset	Bit Position																													
0x51C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**TX pin select register**
	TX pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**TX port select register**
	TX port select register			

23.6.70 GPIO_I2C0_ROUTEEN - I2C0 Pin Enable

Offset	Bit Position																													
0x538	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															SDAPEN	SCLPEN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	SDAPEN	0x0	RW	**SDA pin enable control bit**
	SDA pin enable control bit			
0	SCLPEN	0x0	RW	**SCL pin enable control bit**
	SCL pin enable control bit			

23.6.71 GPIO_I2C0_SCLROUTE - SCL Port/Pin Select

Offset	Bit Position																													
0x53C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset														0x0																	0x0	
Access														RW																	RW	
Name														PIN																	PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**SCL pin select register**
	SCL pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**SCL port select register**
	SCL port select register			

23.6.72 GPIO_I2C0_SDAROUTE - SDA Port/Pin Select

Offset	Bit Position																													
0x540	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**SDA pin select register**
	SDA pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**SDA port select register**
	SDA port select register			

23.6.73 GPIO_I2C1_ROUTEEN - I2C1 Pin Enable

Offset	Bit Position																													
0x548	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															SDAPEN	SCLPEN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	SDAPEN	0x0	RW	**SDA pin enable control bit**
	SDA pin enable control bit			
0	SCLPEN	0x0	RW	**SCL pin enable control bit**
	SCL pin enable control bit			

23.6.74 GPIO_I2C1_SCLROUTE - SCL Port/Pin Select

Offset	Bit Position																													
0x54C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**SCL pin select register**
	SCL pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**SCL port select register**
	SCL port select register			

23.6.75 GPIO_I2C1_SDAROUTE - SDA Port/Pin Select

Offset	Bit Position																													
0x550	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**SDA pin select register**
	SDA pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**SDA port select register**
	SDA port select register			

23.6.76 GPIO_KEYSCAN_ROUTEEN - KEYSCAN Pin Enable

Offset	Bit Position																													
0x558	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																									RW	RW	RW	RW	RW	RW	RW	RW
Name																									COLOUT7PEN	COLOUT6PEN	COLOUT5PEN	COLOUT4PEN	COLOUT3PEN	COLOUT2PEN	COLOUT1PEN	COLOUT0PEN

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	COLOUT7PEN	0x0	RW	**COLOUT7 pin enable control bit**
	COLOUT7 pin enable control bit			
6	COLOUT6PEN	0x0	RW	**COLOUT6 pin enable control bit**
	COLOUT6 pin enable control bit			
5	COLOUT5PEN	0x0	RW	**COLOUT5 pin enable control bit**
	COLOUT5 pin enable control bit			
4	COLOUT4PEN	0x0	RW	**COLOUT4 pin enable control bit**
	COLOUT4 pin enable control bit			
3	COLOUT3PEN	0x0	RW	**COLOUT3 pin enable control bit**
	COLOUT3 pin enable control bit			
2	COLOUT2PEN	0x0	RW	**COLOUT2 pin enable control bit**
	COLOUT2 pin enable control bit			
1	COLOUT1PEN	0x0	RW	**COLOUT1 pin enable control bit**
	COLOUT1 pin enable control bit			
0	COLOUT0PEN	0x0	RW	**COLOUT0 pin enable control bit**
	COLOUT0 pin enable control bit			

23.6.77 GPIO_KEYSCAN_COLOUT0ROUTE - COLOUT0 Port/Pin Select

Offset	Bit Position																													
0x55C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**COLOUT0 pin select register**
	COLOUT0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**COLOUT0 port select register**
	COLOUT0 port select register			

23.6.78 GPIO_KEYSCAN_COLOUT1ROUTE - COLOUT1 Port/Pin Select

Offset	Bit Position																													
0x560	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**COLOUT1 pin select register**
	COLOUT1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**COLOUT1 port select register**
	COLOUT1 port select register			

23.6.79 GPIO_KEYSCAN_COLOUT2ROUTE - COLOUT2 Port/Pin Select

Offset	Bit Position																													
0x564	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**COLOUT2 pin select register**
	COLOUT2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**COLOUT2 port select register**
	COLOUT2 port select register			

23.6.80 GPIO_KEYSCAN_COLOUT3ROUTE - COLOUT3 Port/Pin Select

Offset	Bit Position																													
0x568	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**COLOUT3 pin select register**
	COLOUT3 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**COLOUT3 port select register**
	COLOUT3 port select register			

23.6.81 GPIO_KEYSCAN_COLOUT4ROUTE - COLOUT4 Port/Pin Select

Offset	Bit Position																													
0x56C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**COLOUT4 pin select register**
	COLOUT4 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**COLOUT4 port select register**
	COLOUT4 port select register			

23.6.82 GPIO_KEYSCAN_COLOUT5ROUTE - COLOUT5 Port/Pin Select

Offset	Bit Position																													
0x570	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**COLOUT5 pin select register**
	COLOUT5 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**COLOUT5 port select register**
	COLOUT5 port select register			

23.6.83 GPIO_KEYSCAN_COLOUT6ROUTE - COLOUT6 Port/Pin Select

Offset	Bit Position																													
0x574	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**COLOUT6 pin select register**
	COLOUT6 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**COLOUT6 port select register**
	COLOUT6 port select register			

23.6.84 GPIO_KEYSCAN_COLOUT7ROUTE - COLOUT7 Port/Pin Select

Offset	Bit Position																													
0x578	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**COLOUT7 pin select register**
	COLOUT7 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**COLOUT7 port select register**
	COLOUT7 port select register			

23.6.85 GPIO_KEYSCAN_ROWSENSE0ROUTE - ROWSENSE0 Port/Pin Select

Offset	Bit Position																															
0x57C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ROWSENSE0 pin select register**
	ROWSENSE0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ROWSENSE0 port select register**
	ROWSENSE0 port select register			

23.6.86 GPIO_KEYSCAN_ROWSENSE1ROUTE - ROWSENSE1 Port/Pin Select

Offset	Bit Position																															
0x580	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ROWSENSE1 pin select register**
	ROWSENSE1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ROWSENSE1 port select register**
	ROWSENSE1 port select register			

23.6.87 GPIO_KEYSCAN_ROWSENSE2ROUTE - ROWSENSE2 Port/Pin Select

Offset	Bit Position																													
0x584	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ROWSENSE2 pin select register**
	ROWSENSE2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ROWSENSE2 port select register**
	ROWSENSE2 port select register			

23.6.88 GPIO_KEYSCAN_ROWSENSE3ROUTE - ROWSENSE3 Port/Pin Select

Offset	Bit Position																													
0x588	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ROWSENSE3 pin select register**
	ROWSENSE3 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ROWSENSE3 port select register**
	ROWSENSE3 port select register			

23.6.89 GPIO_KEYSCAN_ROWSENSE4ROUTE - ROWSENSE4 Port/Pin Select

Offset	Bit Position																													
0x58C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ROWSENSE4 pin select register**
	ROWSENSE4 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ROWSENSE4 port select register**
	ROWSENSE4 port select register			

23.6.90 GPIO_KEYSCAN_ROWSENSE5ROUTE - ROWSENSE5 Port/Pin Select

Offset	Bit Position																													
0x590	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ROWSENSE5 pin select register**
	ROWSENSE5 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ROWSENSE5 port select register**
	ROWSENSE5 port select register			

23.6.91 GPIO_LESENSE_ROUTEEN - LESENSE Pin Enable

Offset	Bit Position																													
0x598	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name																	CH15OUTPEN	CH14OUTPEN	CH13OUTPEN	CH12OUTPEN	CH11OUTPEN	CH10OUTPEN	CH9OUTPEN	CH8OUTPEN	CH7OUTPEN	CH6OUTPEN	CH5OUTPEN	CH4OUTPEN	CH3OUTPEN	CH2OUTPEN	CH1OUTPEN	CH0OUTPEN

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15	CH15OUTPEN	0x0	RW	**CH15OUT pin enable control bit**
	CH15OUT pin enable control bit			
14	CH14OUTPEN	0x0	RW	**CH14OUT pin enable control bit**
	CH14OUT pin enable control bit			
13	CH13OUTPEN	0x0	RW	**CH13OUT pin enable control bit**
	CH13OUT pin enable control bit			
12	CH12OUTPEN	0x0	RW	**CH12OUT pin enable control bit**
	CH12OUT pin enable control bit			
11	CH11OUTPEN	0x0	RW	**CH11OUT pin enable control bit**
	CH11OUT pin enable control bit			
10	CH10OUTPEN	0x0	RW	**CH10OUT pin enable control bit**
	CH10OUT pin enable control bit			
9	CH9OUTPEN	0x0	RW	**CH9OUT pin enable control bit**
	CH9OUT pin enable control bit			
8	CH8OUTPEN	0x0	RW	**CH8OUT pin enable control bit**
	CH8OUT pin enable control bit			
7	CH7OUTPEN	0x0	RW	**CH7OUT pin enable control bit**
	CH7OUT pin enable control bit			
6	CH6OUTPEN	0x0	RW	**CH6OUT pin enable control bit**
	CH6OUT pin enable control bit			
5	CH5OUTPEN	0x0	RW	**CH5OUT pin enable control bit**
	CH5OUT pin enable control bit			
4	CH4OUTPEN	0x0	RW	**CH4OUT pin enable control bit**
	CH4OUT pin enable control bit			
3	CH3OUTPEN	0x0	RW	**CH3OUT pin enable control bit**

Bit	Name	Reset	Access	Description
	CH3OUT pin enable control bit			
2	CH2OUTPEN	0x0	RW	**CH2OUT pin enable control bit**
	CH2OUT pin enable control bit			
1	CH1OUTPEN	0x0	RW	**CH1OUT pin enable control bit**
	CH1OUT pin enable control bit			
0	CH0OUTPEN	0x0	RW	**CH0OUT pin enable control bit**
	CH0OUT pin enable control bit			

23.6.92 GPIO_LESENSE_CH0OUTROUTE - CH0OUT Port/Pin Select

Offset	Bit Position																													
0x59C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CH0OUT pin select register**
	CH0OUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CH0OUT port select register**
	CH0OUT port select register			

23.6.93 GPIO_LESENSE_CH1OUTROUTE - CH1OUT Port/Pin Select

Offset	Bit Position																													
0x5A0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CH1OUT pin select register**
	CH1OUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CH1OUT port select register**
	CH1OUT port select register			

23.6.94 GPIO_LESENSE_CH2OUTROUTE - CH2OUT Port/Pin Select

Offset	Bit Position																													
0x5A4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CH2OUT pin select register**
	CH2OUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CH2OUT port select register**
	CH2OUT port select register			

23.6.95 GPIO_LESENSE_CH3OUTROUTE - CH3OUT Port/Pin Select

Offset	Bit Position																													
0x5A8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CH3OUT pin select register**
	CH3OUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CH3OUT port select register**
	CH3OUT port select register			

23.6.96 GPIO_LESENSE_CH4OUTROUTE - CH4OUT Port/Pin Select

Offset	Bit Position																													
0x5AC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CH4OUT pin select register**
	CH4OUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CH4OUT port select register**
	CH4OUT port select register			

23.6.97 GPIO_LESENSE_CH5OUTROUTE - CH5OUT Port/Pin Select

Offset	Bit Position																													
0x5B0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CH5OUT pin select register**
	CH5OUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CH5OUT port select register**
	CH5OUT port select register			

23.6.98 GPIO_LESENSE_CH6OUTROUTE - CH6OUT Port/Pin Select

Offset	Bit Position																													
0x5B4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CH6OUT pin select register**
	CH6OUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CH6OUT port select register**
	CH6OUT port select register			

23.6.99 GPIO_LESENSE_CH7OUTROUTE - CH7OUT Port/Pin Select

Offset	Bit Position																													
0x5B8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CH7OUT pin select register**
	CH7OUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CH7OUT port select register**
	CH7OUT port select register			

23.6.100 GPIO_LESENSE_CH8OUTROUTE - CH8OUT Port/Pin Select

Offset	Bit Position																													
0x5BC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CH8OUT pin select register**
	CH8OUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CH8OUT port select register**
	CH8OUT port select register			

23.6.101 GPIO_LESENSE_CH9OUTROUTE - CH9OUT Port/Pin Select

Offset	Bit Position																													
0x5C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CH9OUT pin select register**
	CH9OUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CH9OUT port select register**
	CH9OUT port select register			

23.6.102 GPIO_LESENSE_CH10OUTROUTE - CH10OUT Port/Pin Select

Offset	Bit Position																													
0x5C4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CH10OUT pin select register**
	CH10OUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CH10OUT port select register**
	CH10OUT port select register			

23.6.103 GPIO_LESENSE_CH11OUTROUTE - CH11OUT Port/Pin Select

Offset	Bit Position																													
0x5C8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CH11OUT pin select register**
	CH11OUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CH11OUT port select register**
	CH11OUT port select register			

23.6.104 GPIO_LESENSE_CH12OUTROUTE - CH12OUT Port/Pin Select

Offset	Bit Position																													
0x5CC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CH12OUT pin select register**
	CH12OUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CH12OUT port select register**
	CH12OUT port select register			

23.6.105 GPIO_LESENSE_CH13OUTROUTE - CH13OUT Port/Pin Select

Offset	Bit Position																													
0x5D0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CH13OUT pin select register**
	CH13OUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CH13OUT port select register**
	CH13OUT port select register			

23.6.106 GPIO_LESENSE_CH14OUTROUTE - CH14OUT Port/Pin Select

Offset	Bit Position																													
0x5D4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CH14OUT pin select register**
	CH14OUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CH14OUT port select register**
	CH14OUT port select register			

23.6.107 GPIO_LESENSE_CH15OUTROUTE - CH15OUT Port/Pin Select

Offset	Bit Position																													
0x5D8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CH15OUT pin select register**
	CH15OUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CH15OUT port select register**
	CH15OUT port select register			

23.6.108 GPIO_LETIMER_ROUTEEN - LETIMER Pin Enable

Offset	Bit Position																													
0x5E0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															OUT1PEN	OUT0PEN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	OUT1PEN	0x0	RW	**OUT1 pin enable control bit**
	OUT1 pin enable control bit			
0	OUT0PEN	0x0	RW	**OUT0 pin enable control bit**
	OUT0 pin enable control bit			

23.6.109 GPIO_LETIMER_OUT0ROUTE - OUT0 Port/Pin Select

Offset	Bit Position																													
0x5E4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**OUT0 pin select register**
	OUT0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**OUT0 port select register**
	OUT0 port select register			

23.6.110 GPIO_LETIMER_OUT1ROUTE - OUT1 Port/Pin Select

Offset	Bit Position																													
0x5E8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**OUT1 pin select register**
	OUT1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**OUT1 port select register**
	OUT1 port select register			

23.6.111 GPIO_MODEM_ROUTEEN - MODEM Pin Enable

Offset	Bit Position																													
0x5F0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name																		DOUTPEN	DCLKPEN	ANTTRIGSTOPPEN	ANTTRIGPEN	ANTSWUSPEN	ANTSWENPEN	ANTRR5PEN	ANTRR4PEN	ANTRR3PEN	ANTRR2PEN	ANTRR1PEN	ANTRR0PEN	ANTROLLOVERPEN	ANT1PEN	ANT0PEN

Bit	Name	Reset	Access	Description
31:15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14	DOUTPEN	0x0	RW	**DOUT pin enable control bit**
	DOUT pin enable control bit			
13	DCLKPEN	0x0	RW	**DCLK pin enable control bit**
	DCLK pin enable control bit			
12	ANTTRIGSTOPPEN	0x0	RW	**ANTTRIGSTOP pin enable control bit**
	ANTTRIGSTOP pin enable control bit			
11	ANTTRIGPEN	0x0	RW	**ANTTRIG pin enable control bit**
	ANTTRIG pin enable control bit			
10	ANTSWUSPEN	0x0	RW	**ANTSWUS pin enable control bit**
	ANTSWUS pin enable control bit			
9	ANTSWENPEN	0x0	RW	**ANTSWEN pin enable control bit**
	ANTSWEN pin enable control bit			
8	ANTRR5PEN	0x0	RW	**ANTRR5 pin enable control bit**
	ANTRR5 pin enable control bit			
7	ANTRR4PEN	0x0	RW	**ANTRR4 pin enable control bit**
	ANTRR4 pin enable control bit			
6	ANTRR3PEN	0x0	RW	**ANTRR3 pin enable control bit**
	ANTRR3 pin enable control bit			
5	ANTRR2PEN	0x0	RW	**ANTRR2 pin enable control bit**
	ANTRR2 pin enable control bit			
4	ANTRR1PEN	0x0	RW	**ANTRR1 pin enable control bit**
	ANTRR1 pin enable control bit			
3	ANTRR0PEN	0x0	RW	**ANTRR0 pin enable control bit**
	ANTRR0 pin enable control bit			

Bit	Name	Reset	Access	Description
2	ANTROLLOVERPEN	0x0	RW	**ANTROLLOVER pin enable control bit**
	ANTROLLOVER pin enable control bit			
1	ANT1PEN	0x0	RW	**ANT1 pin enable control bit**
	ANT1 pin enable control bit			
0	ANT0PEN	0x0	RW	**ANT0 pin enable control bit**
	ANT0 pin enable control bit			

23.6.112 GPIO_MODEM_ANT0ROUTE - ANT0 Port/Pin Select

Offset	Bit Position																													
0x5F4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANT0 pin select register**
	ANT0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANT0 port select register**
	ANT0 port select register			

23.6.113 GPIO_MODEM_ANT1ROUTE - ANT1 Port/Pin Select

Offset	Bit Position																													
0x5F8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANT1 pin select register**
	ANT1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANT1 port select register**
	ANT1 port select register			

23.6.114 GPIO_MODEM_ANTROLLOVERROUTE - ANTROLLOVER Port/Pin Select

Offset	Bit Position																													
0x5FC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTROLLOVER pin select register**
	ANTROLLOVER pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTROLLOVER port select register**
	ANTROLLOVER port select register			

23.6.115 GPIO_MODEM_ANTRR0ROUTE - ANTRR0 Port/Pin Select

Offset	Bit Position																													
0x600	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTRR0 pin select register**
	ANTRR0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTRR0 port select register**
	ANTRR0 port select register			

23.6.116 GPIO_MODEM_ANTRR1ROUTE - ANTRR1 Port/Pin Select

Offset	Bit Position																													
0x604	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTRR1 pin select register**
	ANTRR1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTRR1 port select register**
	ANTRR1 port select register			

23.6.117 GPIO_MODEM_ANTRR2ROUTE - ANTRR2 Port/Pin Select

Offset	Bit Position																													
0x608	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTRR2 pin select register**
	ANTRR2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTRR2 port select register**
	ANTRR2 port select register			

23.6.118 GPIO_MODEM_ANTRR3ROUTE - ANTRR3 Port/Pin Select

Offset	Bit Position																													
0x60C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTRR3 pin select register**
	ANTRR3 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTRR3 port select register**
	ANTRR3 port select register			

23.6.119 GPIO_MODEM_ANTRR4ROUTE - ANTRR4 Port/Pin Select

Offset	Bit Position																													
0x610	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTRR4 pin select register**
	ANTRR4 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTRR4 port select register**
	ANTRR4 port select register			

23.6.120 GPIO_MODEM_ANTRR5ROUTE - ANTRR5 Port/Pin Select

Offset	Bit Position																													
0x614	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTRR5 pin select register**
	ANTRR5 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTRR5 port select register**
	ANTRR5 port select register			

23.6.121 GPIO_MODEM_ANTSWENROUTE - ANTSWEN Port/Pin Select

Offset	Bit Position																													
0x618	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTSWEN pin select register**
	ANTSWEN pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTSWEN port select register**
	ANTSWEN port select register			

23.6.122 GPIO_MODEM_ANTSWUSROUTE - ANTSWUS Port/Pin Select

Offset	Bit Position																													
0x61C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTSWUS pin select register**
	ANTSWUS pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTSWUS port select register**
	ANTSWUS port select register			

23.6.123 GPIO_MODEM_ANTTRIGROUTE - ANTTRIG Port/Pin Select

Offset	Bit Position																													
0x620	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTTRIG pin select register**
	ANTTRIG pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTTRIG port select register**
	ANTTRIG port select register			

23.6.124 GPIO_MODEM_ANTTRIGSTOPROUTE - ANTTRIGSTOP Port/Pin Select

Offset	Bit Position																													
0x624	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTTRIGSTOP pin select register**
	ANTTRIGSTOP pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTTRIGSTOP port select register**
	ANTTRIGSTOP port select register			

23.6.125 GPIO_MODEM_DCLKROUTE - DCLK Port/Pin Select

Offset	Bit Position																													
0x628	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**DCLK pin select register**
	DCLK pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**DCLK port select register**
	DCLK port select register			

23.6.126 GPIO_MODEM_DINROUTE - DIN Port/Pin Select

Offset	Bit Position																													
0x62C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**DIN pin select register**
	DIN pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**DIN port select register**
	DIN port select register			

23.6.127 GPIO_MODEM_DOUTROUTE - DOUT Port/Pin Select

Offset	Bit Position																													
0x630	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**DOUT pin select register**
	DOUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**DOUT port select register**
	DOUT port select register			

23.6.128 GPIO_PCNT0_S0INROUTE - S0IN Port/Pin Select

Offset	Bit Position																													
0x63C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**S0IN pin select register**
	S0IN pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**S0IN port select register**
	S0IN port select register			

23.6.129 GPIO_PCNT0_S1INROUTE - S1IN Port/Pin Select

Offset	Bit Position																													
0x640	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**S1IN pin select register**
	S1IN pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**S1IN port select register**
	S1IN port select register			

23.6.130 GPIO_PRS0_ROUTEEN - PRS0 Pin Enable

Offset	Bit Position																													
0x648	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name																	SYNCH3PEN	SYNCH2PEN	SYNCH1PEN	SYNCH0PEN	ASYNCH11PEN	ASYNCH10PEN	ASYNCH9PEN	ASYNCH8PEN	ASYNCH7PEN	ASYNCH6PEN	ASYNCH5PEN	ASYNCH4PEN	ASYNCH3PEN	ASYNCH2PEN	ASYNCH1PEN	ASYNCH0PEN

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15	SYNCH3PEN	0x0	RW	**SYNCH3 pin enable control bit**
	SYNCH3 pin enable control bit			
14	SYNCH2PEN	0x0	RW	**SYNCH2 pin enable control bit**
	SYNCH2 pin enable control bit			
13	SYNCH1PEN	0x0	RW	**SYNCH1 pin enable control bit**
	SYNCH1 pin enable control bit			
12	SYNCH0PEN	0x0	RW	**SYNCH0 pin enable control bit**
	SYNCH0 pin enable control bit			
11	ASYNCH11PEN	0x0	RW	**ASYNCH11 pin enable control bit**
	ASYNCH11 pin enable control bit			
10	ASYNCH10PEN	0x0	RW	**ASYNCH10 pin enable control bit**
	ASYNCH10 pin enable control bit			
9	ASYNCH9PEN	0x0	RW	**ASYNCH9 pin enable control bit**
	ASYNCH9 pin enable control bit			
8	ASYNCH8PEN	0x0	RW	**ASYNCH8 pin enable control bit**
	ASYNCH8 pin enable control bit			
7	ASYNCH7PEN	0x0	RW	**ASYNCH7 pin enable control bit**
	ASYNCH7 pin enable control bit			
6	ASYNCH6PEN	0x0	RW	**ASYNCH6 pin enable control bit**
	ASYNCH6 pin enable control bit			
5	ASYNCH5PEN	0x0	RW	**ASYNCH5 pin enable control bit**
	ASYNCH5 pin enable control bit			
4	ASYNCH4PEN	0x0	RW	**ASYNCH4 pin enable control bit**
	ASYNCH4 pin enable control bit			
3	ASYNCH3PEN	0x0	RW	**ASYNCH3 pin enable control bit**

Bit	Name	Reset	Access	Description
	ASYNCH3 pin enable control bit			
2	ASYNCH2PEN	0x0	RW	**ASYNCH2 pin enable control bit**
	ASYNCH2 pin enable control bit			
1	ASYNCH1PEN	0x0	RW	**ASYNCH1 pin enable control bit**
	ASYNCH1 pin enable control bit			
0	ASYNCH0PEN	0x0	RW	**ASYNCH0 pin enable control bit**
	ASYNCH0 pin enable control bit			

23.6.131 GPIO_PRS0_ASYNCH0ROUTE - ASYNCH0 Port/Pin Select

Offset	Bit Position																													
0x64C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH0 pin select register**
	ASYNCH0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH0 port select register**
	ASYNCH0 port select register			

23.6.132 GPIO_PRS0_ASYNCH1ROUTE - ASYNCH1 Port/Pin Select

Offset	Bit Position																													
0x650	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH1 pin select register**
	ASYNCH1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH1 port select register**
	ASYNCH1 port select register			

23.6.133 GPIO_PRS0_ASYNCH2ROUTE - ASYNCH2 Port/Pin Select

Offset	Bit Position																													
0x654	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH2 pin select register**
	ASYNCH2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH2 port select register**
	ASYNCH2 port select register			

23.6.134 GPIO_PRS0_ASYNCH3ROUTE - ASYNCH3 Port/Pin Select

Offset	Bit Position																													
0x658	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH3 pin select register**
	ASYNCH3 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH3 port select register**
	ASYNCH3 port select register			

23.6.135 GPIO_PRS0_ASYNCH4ROUTE - ASYNCH4 Port/Pin Select

Offset	Bit Position																													
0x65C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH4 pin select register**
	ASYNCH4 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH4 port select register**
	ASYNCH4 port select register			

23.6.136 GPIO_PRS0_ASYNCH5ROUTE - ASYNCH5 Port/Pin Select

Offset	Bit Position																													
0x660	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH5 pin select register**
	ASYNCH5 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH5 port select register**
	ASYNCH5 port select register			

23.6.137 GPIO_PRS0_ASYNCH6ROUTE - ASYNCH6 Port/Pin Select

Offset	Bit Position																													
0x664	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH6 pin select register**
	ASYNCH6 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH6 port select register**
	ASYNCH6 port select register			

23.6.138 GPIO_PRS0_ASYNCH7ROUTE - ASYNCH7 Port/Pin Select

Offset	Bit Position																													
0x668	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH7 pin select register**
	ASYNCH7 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH7 port select register**
	ASYNCH7 port select register			

23.6.139 GPIO_PRS0_ASYNCH8ROUTE - ASYNCH8 Port/Pin Select

Offset	Bit Position																													
0x66C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH8 pin select register**
	ASYNCH8 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH8 port select register**
	ASYNCH8 port select register			

23.6.140 GPIO_PRS0_ASYNCH9ROUTE - ASYNCH9 Port/Pin Select

Offset	Bit Position																													
0x670	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH9 pin select register**
	ASYNCH9 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH9 port select register**
	ASYNCH9 port select register			

23.6.141 GPIO_PRS0_ASYNCH10ROUTE - ASYNCH10 Port/Pin Select

Offset	Bit Position																													
0x674	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH10 pin select register**
	ASYNCH10 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH10 port select register**
	ASYNCH10 port select register			

23.6.142 GPIO_PRS0_ASYNCH11ROUTE - ASYNCH11 Port/Pin Select

Offset	Bit Position																													
0x678	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH11 pin select register**
	ASYNCH11 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH11 port select register**
	ASYNCH11 port select register			

23.6.143 GPIO_PRS0_SYNCH0ROUTE - SYNCH0 Port/Pin Select

Offset	Bit Position																													
0x67C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**SYNCH0 pin select register**
	SYNCH0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**SYNCH0 port select register**
	SYNCH0 port select register			

23.6.144 GPIO_PRS0_SYNCH1ROUTE - SYNCH1 Port/Pin Select

Offset	Bit Position																													
0x680	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**SYNCH1 pin select register**
	SYNCH1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**SYNCH1 port select register**
	SYNCH1 port select register			

23.6.145 GPIO_PRS0_SYNCH2ROUTE - SYNCH2 Port/Pin Select

Offset	Bit Position																													
0x684	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**SYNCH2 pin select register**
	SYNCH2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**SYNCH2 port select register**
	SYNCH2 port select register			

23.6.146 GPIO_PRS0_SYNCH3ROUTE - SYNCH3 Port/Pin Select

Offset	Bit Position																													
0x688	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**SYNCH3 pin select register**
	SYNCH3 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**SYNCH3 port select register**
	SYNCH3 port select register			

23.6.147 GPIO_SYXO0_BUFOUTREQINASYNCROUTE - BUFOUTREQINASYNC Port/Pin Select

Offset	Bit Position																													
0x6F0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**BUFOUTREQINASYNC pin select register**
	BUFOUTREQINASYNC pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**BUFOUTREQINASYNC port select register**
	BUFOUTREQINASYNC port select register			

23.6.148 GPIO_TIMER0_ROUTEEN - TIMER0 Pin Enable

Offset	Bit Position																													
0x6F8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											RW	RW	RW	RW	RW	RW
Name																											CCC2PEN	CCC1PEN	CCC0PEN	CC2PEN	CC1PEN	CC0PEN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	CCC2PEN	0x0	RW	**CDTI2 pin enable control bit**
	CDTI2 pin enable control bit			
4	CCC1PEN	0x0	RW	**CDTI1 pin enable control bit**
	CDTI1 pin enable control bit			
3	CCC0PEN	0x0	RW	**CDTI0 pin enable control bit**
	CDTI0 pin enable control bit			
2	CC2PEN	0x0	RW	**CC2 pin enable control bit**
	CC2 pin enable control bit			
1	CC1PEN	0x0	RW	**CC1 pin enable control bit**
	CC1 pin enable control bit			
0	CC0PEN	0x0	RW	**CC0 pin enable control bit**
	CC0 pin enable control bit			

23.6.149 GPIO_TIMER0_CC0ROUTE - CC0 Port/Pin Select

Offset	Bit Position																													
0x6FC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CC0 pin select register**
	CC0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CC0 port select register**
	CC0 port select register			

23.6.150 GPIO_TIMER0_CC1ROUTE - CC1 Port/Pin Select

Offset	Bit Position																													
0x700	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CC1 pin select register**
	CC1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CC1 port select register**
	CC1 port select register			

23.6.151 GPIO_TIMER0_CC2ROUTE - CC2 Port/Pin Select

Offset	Bit Position																													
0x704	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CC2 pin select register**
	CC2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CC2 port select register**
	CC2 port select register			

23.6.152 GPIO_TIMER0_CDTI0ROUTE - CDTI0 Port/Pin Select

Offset	Bit Position																													
0x708	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CDTI0 pin select register**
	CDTI0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CDTI0 port select register**
	CDTI0 port select register			

23.6.153 GPIO_TIMER0_CDTI1ROUTE - CDTI1 Port/Pin Select

Offset	Bit Position																													
0x70C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CDTI1 pin select register**
	CDTI1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CDTI1 port select register**
	CDTI1 port select register			

23.6.154 GPIO_TIMER0_CDTI2ROUTE - CDTI2 Port/Pin Select

Offset	Bit Position																													
0x710	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CDTI2 pin select register**
	CDTI2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CDTI2 port select register**
	CDTI2 port select register			

23.6.155 GPIO_TIMER1_ROUTEEN - TIMER1 Pin Enable

Offset	Bit Position																													
0x718	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											RW	RW	RW	RW	RW	RW
Name																											CCC2PEN	CCC1PEN	CCC0PEN	CC2PEN	CC1PEN	CC0PEN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	CCC2PEN	0x0	RW	**CDTI2 pin enable control bit**
	CDTI2 pin enable control bit			
4	CCC1PEN	0x0	RW	**CDTI1 pin enable control bit**
	CDTI1 pin enable control bit			
3	CCC0PEN	0x0	RW	**CDTI0 pin enable control bit**
	CDTI0 pin enable control bit			
2	CC2PEN	0x0	RW	**CC2 pin enable control bit**
	CC2 pin enable control bit			
1	CC1PEN	0x0	RW	**CC1 pin enable control bit**
	CC1 pin enable control bit			
0	CC0PEN	0x0	RW	**CC0 pin enable control bit**
	CC0 pin enable control bit			

23.6.156 GPIO_TIMER1_CC0ROUTE - CC0 Port/Pin Select

Offset	Bit Position																													
0x71C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CC0 pin select register**
	CC0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CC0 port select register**
	CC0 port select register			

23.6.157 GPIO_TIMER1_CC1ROUTE - CC1 Port/Pin Select

Offset	Bit Position																													
0x720	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CC1 pin select register**
	CC1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CC1 port select register**
	CC1 port select register			

23.6.158 GPIO_TIMER1_CC2ROUTE - CC2 Port/Pin Select

Offset	Bit Position																													
0x724	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CC2 pin select register**
	CC2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CC2 port select register**
	CC2 port select register			

23.6.159 GPIO_TIMER1_CDTI0ROUTE - CDTI0 Port/Pin Select

Offset	Bit Position																													
0x728	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CDTI0 pin select register**
	CDTI0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CDTI0 port select register**
	CDTI0 port select register			

23.6.160 GPIO_TIMER1_CDTI1ROUTE - CDTI1 Port/Pin Select

Offset	Bit Position																													
0x72C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CDTI1 pin select register**
	CDTI1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CDTI1 port select register**
	CDTI1 port select register			

23.6.161 GPIO_TIMER1_CDTI2ROUTE - CDTI2 Port/Pin Select

Offset	Bit Position																													
0x730	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CDTI2 pin select register**
	CDTI2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CDTI2 port select register**
	CDTI2 port select register			

23.6.162 GPIO_TIMER2_ROUTEEN - TIMER2 Pin Enable

Offset	Bit Position																													
0x738	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											RW	RW	RW	RW	RW	RW
Name																											CCC2PEN	CCC1PEN	CCC0PEN	CC2PEN	CC1PEN	CC0PEN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	CCC2PEN	0x0	RW	**CDTI2 pin enable control bit**
	CDTI2 pin enable control bit			
4	CCC1PEN	0x0	RW	**CDTI1 pin enable control bit**
	CDTI1 pin enable control bit			
3	CCC0PEN	0x0	RW	**CDTI0 pin enable control bit**
	CDTI0 pin enable control bit			
2	CC2PEN	0x0	RW	**CC2 pin enable control bit**
	CC2 pin enable control bit			
1	CC1PEN	0x0	RW	**CC1 pin enable control bit**
	CC1 pin enable control bit			
0	CC0PEN	0x0	RW	**CC0 pin enable control bit**
	CC0 pin enable control bit			

23.6.163 GPIO_TIMER2_CC0ROUTE - CC0 Port/Pin Select

Offset	Bit Position																													
0x73C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CC0 pin select register**
	CC0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CC0 port select register**
	CC0 port select register			

23.6.164 GPIO_TIMER2_CC1ROUTE - CC1 Port/Pin Select

Offset	Bit Position																													
0x740	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CC1 pin select register**
	CC1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CC1 port select register**
	CC1 port select register			

23.6.165 GPIO_TIMER2_CC2ROUTE - CC2 Port/Pin Select

Offset	Bit Position																													
0x744	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CC2 pin select register**
	CC2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CC2 port select register**
	CC2 port select register			

23.6.166 GPIO_TIMER2_CDTI0ROUTE - CDTI0 Port/Pin Select

Offset	Bit Position																													
0x748	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CDTI0 pin select register**
	CDTI0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CDTI0 port select register**
	CDTI0 port select register			

23.6.167 GPIO_TIMER2_CDTI1ROUTE - CDTI1 Port/Pin Select

Offset	Bit Position																													
0x74C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CDTI1 pin select register**
	CDTI1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CDTI1 port select register**
	CDTI1 port select register			

23.6.168 GPIO_TIMER2_CDTI2ROUTE - CDTI2 Port/Pin Select

Offset	Bit Position																													
0x750	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CDTI2 pin select register**
	CDTI2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CDTI2 port select register**
	CDTI2 port select register			

23.6.169 GPIO_TIMER3_ROUTEEN - TIMER3 Pin Enable

Offset	Bit Position																													
0x758	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											RW	RW	RW	RW	RW	RW
Name																											CCC2PEN	CCC1PEN	CCC0PEN	CC2PEN	CC1PEN	CC0PEN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	CCC2PEN	0x0	RW	**CDTI2 pin enable control bit**
	CDTI2 pin enable control bit			
4	CCC1PEN	0x0	RW	**CDTI1 pin enable control bit**
	CDTI1 pin enable control bit			
3	CCC0PEN	0x0	RW	**CDTI0 pin enable control bit**
	CDTI0 pin enable control bit			
2	CC2PEN	0x0	RW	**CC2 pin enable control bit**
	CC2 pin enable control bit			
1	CC1PEN	0x0	RW	**CC1 pin enable control bit**
	CC1 pin enable control bit			
0	CC0PEN	0x0	RW	**CC0 pin enable control bit**
	CC0 pin enable control bit			

23.6.170 GPIO_TIMER3_CC0ROUTE - CC0 Port/Pin Select

Offset	Bit Position																													
0x75C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CC0 pin select register**
	CC0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CC0 port select register**
	CC0 port select register			

23.6.171 GPIO_TIMER3_CC1ROUTE - CC1 Port/Pin Select

Offset	Bit Position																													
0x760	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CC1 pin select register**
	CC1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CC1 port select register**
	CC1 port select register			

23.6.172 GPIO_TIMER3_CC2ROUTE - CC2 Port/Pin Select

Offset	Bit Position																													
0x764	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CC2 pin select register**
	CC2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CC2 port select register**
	CC2 port select register			

23.6.173 GPIO_TIMER3_CDTI0ROUTE - CDTI0 Port/Pin Select

Offset	Bit Position																													
0x768	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CDTI0 pin select register**
	CDTI0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CDTI0 port select register**
	CDTI0 port select register			

23.6.174 GPIO_TIMER3_CDTI1ROUTE - CDTI1 Port/Pin Select

Offset	Bit Position																													
0x76C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CDTI1 pin select register**
	CDTI1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CDTI1 port select register**
	CDTI1 port select register			

23.6.175 GPIO_TIMER3_CDTI2ROUTE - CDTI2 Port/Pin Select

Offset	Bit Position																													
0x770	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CDTI2 pin select register**
	CDTI2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CDTI2 port select register**
	CDTI2 port select register			

23.6.176 GPIO_TIMER4_ROUTEEN - TIMER4 Pin Enable

Offset	Bit Position																													
0x778	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											RW	RW	RW	RW	RW	RW
Name																											CCC2PEN	CCC1PEN	CCC0PEN	CC2PEN	CC1PEN	CC0PEN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	CCC2PEN	0x0	RW	**CDTI2 pin enable control bit**
	CDTI2 pin enable control bit			
4	CCC1PEN	0x0	RW	**CDTI1 pin enable control bit**
	CDTI1 pin enable control bit			
3	CCC0PEN	0x0	RW	**CDTI0 pin enable control bit**
	CDTI0 pin enable control bit			
2	CC2PEN	0x0	RW	**CC2 pin enable control bit**
	CC2 pin enable control bit			
1	CC1PEN	0x0	RW	**CC1 pin enable control bit**
	CC1 pin enable control bit			
0	CC0PEN	0x0	RW	**CC0 pin enable control bit**
	CC0 pin enable control bit			

23.6.177 GPIO_TIMER4_CC0ROUTE - CC0 Port/Pin Select

Offset	Bit Position																													
0x77C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CC0 pin select register**
	CC0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CC0 port select register**
	CC0 port select register			

23.6.178 GPIO_TIMER4_CC1ROUTE - CC1 Port/Pin Select

Offset	Bit Position																													
0x780	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CC1 pin select register**
	CC1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CC1 port select register**
	CC1 port select register			

23.6.179 GPIO_TIMER4_CC2ROUTE - CC2 Port/Pin Select

Offset	Bit Position																													
0x784	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CC2 pin select register**
	CC2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CC2 port select register**
	CC2 port select register			

23.6.180 GPIO_TIMER4_CDTI0ROUTE - CDTI0 Port/Pin Select

Offset	Bit Position																													
0x788	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CDTI0 pin select register**
	CDTI0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CDTI0 port select register**
	CDTI0 port select register			

23.6.181 GPIO_TIMER4_CDTI1ROUTE - CDTI1 Port/Pin Select

Offset	Bit Position																													
0x78C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CDTI1 pin select register**
	CDTI1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CDTI1 port select register**
	CDTI1 port select register			

23.6.182 GPIO_TIMER4_CDTI2ROUTE - CDTI2 Port/Pin Select

Offset	Bit Position																													
0x790	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CDTI2 pin select register**
	CDTI2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CDTI2 port select register**
	CDTI2 port select register			

23.6.183 GPIO_USART0_ROUTEEN - USART0 Pin Enable

Offset	Bit Position																													
0x798	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												TXPEN	CLKPEN	RXPEN	RTSPEN	CSPEN

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	TXPEN	0x0	RW	**TX pin enable control bit**
	TX pin enable control bit			
3	CLKPEN	0x0	RW	**SCLK pin enable control bit**
	SCLK pin enable control bit			
2	RXPEN	0x0	RW	**RX pin enable control bit**
	RX pin enable control bit			
1	RTSPEN	0x0	RW	**RTS pin enable control bit**
	RTS pin enable control bit			
0	CSPEN	0x0	RW	**CS pin enable control bit**
	CS pin enable control bit			

23.6.184 GPIO_USART0_CSROUTE - CS Port/Pin Select

Offset	Bit Position																													
0x79C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CS pin select register**
	CS pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CS port select register**
	CS port select register			

23.6.185 GPIO_USART0_CTSROUTE - CTS Port/Pin Select

Offset	Bit Position																													
0x7A0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CTS pin select register**
	CTS pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CTS port select register**
	CTS port select register			

23.6.186 GPIO_USART0_RTSROUTE - RTS Port/Pin Select

Offset	Bit Position																													
0x7A4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**RTS pin select register**
	RTS pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**RTS port select register**
	RTS port select register			

23.6.187 GPIO_USART0_RXROUTE - RX Port/Pin Select

Offset	Bit Position																													
0x7A8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**RX pin select register**
	RX pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**RX port select register**
	RX port select register			

23.6.188 GPIO_USART0_CLKROUTE - SCLK Port/Pin Select

Offset	Bit Position																													
0x7AC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

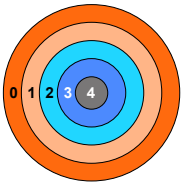
Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**SCLK pin select register**
	SCLK pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**SCLK port select register**
	SCLK port select register			

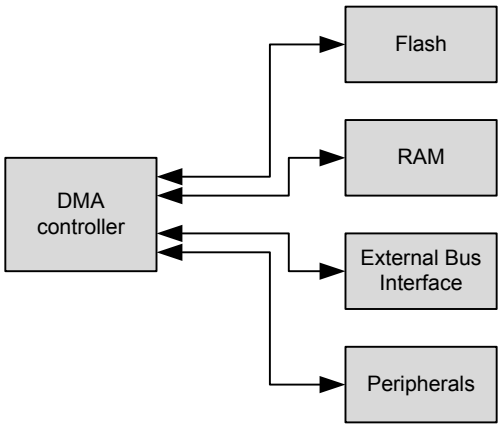
23.6.189 GPIO_USART0_TXROUTE - TX Port/Pin Select

Offset	Bit Position																													
0x7B0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**TX pin select register**
	TX pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**TX port select register**
	TX port select register			

24. LDMA - Linked DMA





Quick Facts

What?

The LDMA controller can move data without CPU intervention, effectively reducing the energy consumption for a data transfer.

Why?

The LDMA can perform data transfers more energy efficiently than the CPU and allows autonomous operation in low energy modes.

How?

The LDMA controller has multiple highly configurable, prioritized DMA channels. A linked list of flexible descriptors makes it possible to tailor the controller to the specific needs of an application.

24.1 Introduction

The Linked Direct Memory Access (LDMA) controller performs memory transfer operations independently of the CPU. This has the benefit of reducing the energy consumption and the workload of the CPU, and enables the system to stay in low energy modes while still routing data to memory and peripherals. For example, moving data from the ADC to memory.

24.1.1 Features

- Flexible Source and Destination transfers
  - Memory-to-memory
  - Memory-to-peripheral
  - Peripheral-to-memory
  - Peripheral-to-peripheral
- DMA transfers triggered by peripherals, software, or linked list
- Single or multiple data transfers for each peripheral or software request
- Inter-channel and hardware event synchronization via trigger and wait functions
- Supports single or multiple descriptors
  - Single descriptor
  - Linked list of descriptors
  - Circular and ping-pong buffers
  - Scatter-Gather
  - Looping
  - Pause and restart triggered by other channels
  - Sophisticated flow control which can function without CPU interaction
- Channel arbitration includes:
  - Fixed priority
  - Simple round robin
  - Round robin with programmable multiple interleaved entries for higher priority requesters
- Programmable data size and source and destination address strides
- Programmable interrupt generation at the end of each DMA descriptor execution
- Little-endian/big-endian conversion
- DMA write-immediate function

24.2 Block Diagram

An overview of the LDMA and the modules it interacts with is shown in Figure 24.1 LDMA Block Diagram on page 975.

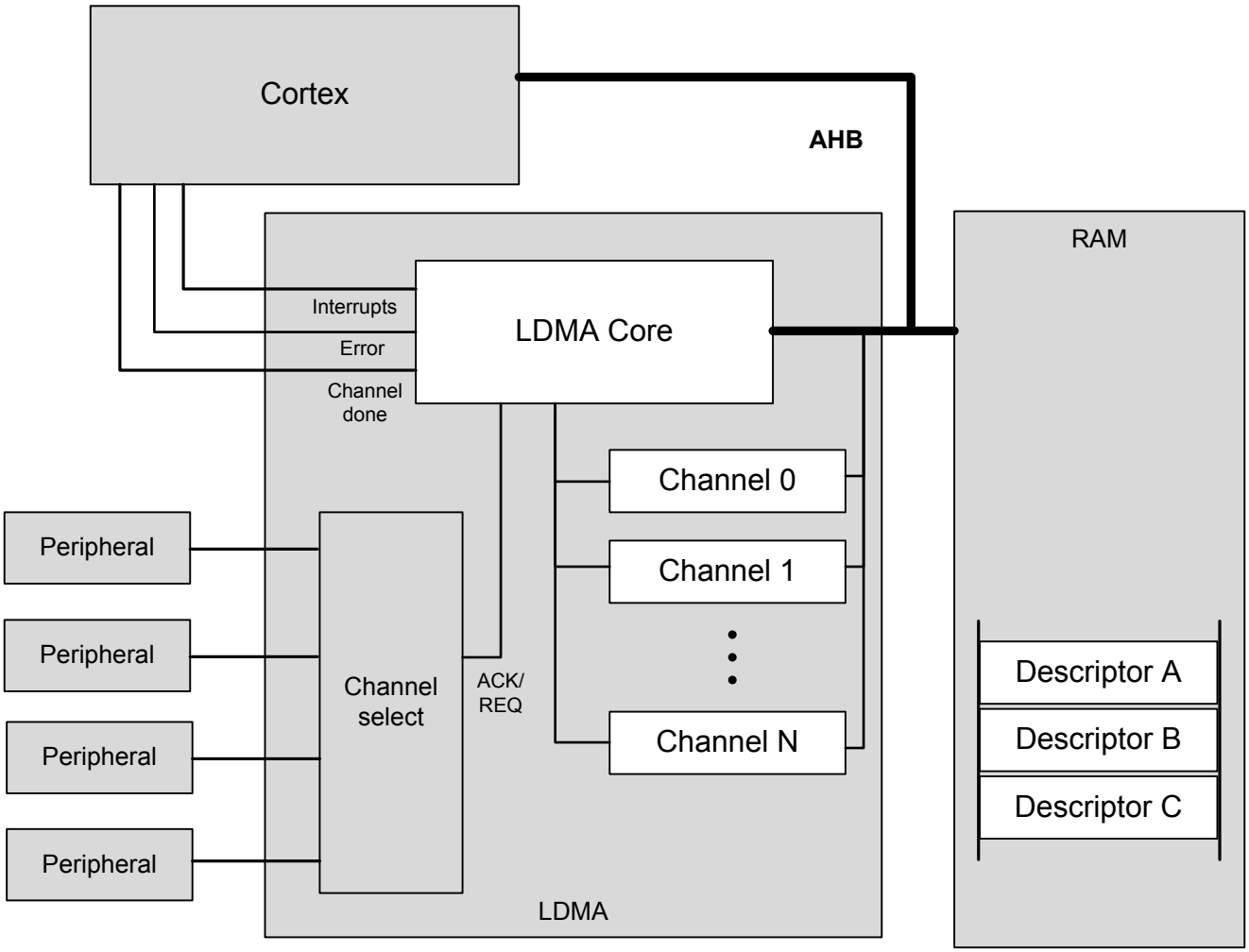


Figure 24.1. LDMA Block Diagram

The Linked DMA Controller consists of three main parts

- A DMA core that executes transfers and communicates status to the core
- A channel select block that routes peripheral DMA requests and acknowledge signals to the DMA
- A set of internal channel configuration registers for tracking the progress of each DMA channel

The DMA has access to all system memory through the AHB bus and the AHB->APB bridge. It can load channel descriptors from memory with no CPU intervention.

24.3 Functional Description

The Linked DMA Controller is highly flexible. It is capable of transferring data between peripherals and memory without involvement from the processor core. This can be used to increase system performance by off-loading the processor from copying large amounts of data or avoiding frequent interrupts to service peripherals needing more data or having available data. It can also be used to reduce the system energy consumption by making the LDMA work autonomously with some EM2/3 peripherals for data transfer without having to wake up the processor core from sleep.

The Linked DMA Controller has 8 independent channels. Each of these channels can be connected to any of the available peripheral DMA transfer request input sources by writing to the channel configuration registers, see 24.3.2 Channel Configuration. In addition, each channel can also be triggered directly by software, which is useful for memory-to-memory transfers.

The channel descriptors determine what the Linked DMA Controller will do when it receives DMA transfer request. The initial descriptor is written directly to the LDMA's channel registers. If desired, the initial descriptor can link to additional linked descriptors stored in memory (RAM or Flash). Alternatively, software may also load the initial descriptor by writing the descriptor address to the LDMA_CHx_LINK register and then setting the corresponding bit the LDMA_LINKLOAD register.

Before enabling a channel, the software must take care to properly configure the channel registers including the link address and any linked descriptors. When a channel is triggered, the Linked DMA Controller will perform the memory transfers as specified by the descriptors. A descriptor contains the memory address to read from, the memory address to write to, link address of the next descriptor, the number of bytes to be transferred, etc. The channel descriptor is described in detail in 24.3.7 Channel Descriptor Data Structure.

The Linked DMA Controller supports both fixed priority and round robin arbitration. The number of fixed and round robin channels is programmable. For round robin channels, the number of arbitration slots requested for each channel is programmable. Using this scheme, it is possible to ensure that timing-critical transfers are serviced on time.

DMA transfers take place by reading a block of data at a time from the source, storing it in the LDMA’s local FIFO, then writing the block out to the destination from the FIFO. Interrupts may optionally be signaled to the CPU’s interrupt controller at the end of any DMA transfer or at the completion of a descriptor if the DONEIFSEN bit is set. An AHB error will always generate an interrupt.

24.3.1 Channel Descriptor

Each DMA channel has descriptor registers. A transfer can be initialized by software writing to the registers or by the DMA itself copying a descriptor from RAM to memory. When using a linked list of descriptors the first descriptor should be initialized by the CPU. The DMA itself will then copy linked descriptors to its descriptor registers as required. In addition to manually initializing the first transfer, software may also cause the LDMA to load the initial descriptor by writing the descriptor address to the LDMA_CHx_LINK register and then setting the corresponding bit the LDMA_LINKLOAD register.

The contents of the descriptor registers are dynamically updated during the DMA transfer. The contents of descriptors in memory are not edited by the controller.

Some descriptor field values are only used for linked descriptors. For example, the SRCMODE and DSTMODE bits of the LDMA_CHx_CTRL registers determine if a linked descriptor is using relative or absolute addressing. Software writes to the address registers will always use absolute addressing and never set these bits. Therefore, these bits are read only.

24.3.1.1 DMA Transfer Size

A DMA transfer is the smallest unit of data that can be transfered by the LDMA. The LDMA supports byte, half-word and word sized transfers. The SIZE field in the LDMA_CHx_CTRL register specifies the data width of one DMA transfer.

24.3.1.2 Source/Destination Increments

The SRCINC and DSTINC in the LDMA_CHx_CTRL register determines the increment between DMA transfers. The increment is in units of DMA transfers and using an increment size of 1 will transfer contiguous bytes, half-words, or words depending on the value of the SIZE field. Multiple unit increments are useful for transferring or packing/unpacking alligned data. For example using an increment of 4 with a size of BYTE will transfer word aligned bytes. An increment of 2 units with a size of HALFWORD is suitable for the transfer of word aligned half-word data. The LDMA can also pack or unpack data by using a different increment size for source and destination. For example - to convert from word aligned byte data (unpacked) to contiguous byte data (packed), set the SIZE to BYTE, SRCINC to 4, and DSTINC to 1.

SIZE may also be set to NONE which will cause the LDMA to read or write the same location for every DMA transfer. This is usefull for accessing peripheral FIFO or data registers.

24.3.1.3 Block Size

The block size defines the amount of data transferred in one arbitration. It consists of one or more DMA transfers. See 24.3.6.1 Arbitration Priority for more details.

24.3.1.4 Transfer Count

The descriptor transfer count defines how many DMA transfers to perform. The number of bytes transferred by the descripter will depend on both the transfer count XFERCNT and the SIZE field settings. TOTAL_BYTES = XFERCNT * SIZE

24.3.1.5 Descriptor List

A descriptor list consists of one or more descriptors which are executed serially. This list may be a simple sequence of descriptors, a loop of descriptors, or a combination of the two.

Each descriptor in the list can be one of several types.

- Single Transfer descriptor: Transfers TOTAL_BYTES of data and then stops.
- Linked Transfer descriptor: Transfers TOTAL_BYTES of data and then loads the next linked descriptor.
- Loop Transfer descriptor: Transfers TOTAL_BYTES of data and performs loop control (see 24.3.2.2 Loop Counter).
- Sync descriptor: Handle synchronization of the list with other entities (see 24.3.7.2 SYNC Descriptor Structure).
- WRI descriptor: Writes a value to a location in memory (see 24.3.7.3 WRI Descriptor Structure).

24.3.1.6 Addresses

Before initiating a transfer, software should write the source address, destination address, and if applicable the link address to the descriptor registers. Alternatively, software may load a descriptor from memory by writing the descriptor address to the LDMA_CHx_LINK register and setting the corresponding bit in the LDMA_LINKLOAD register.

During a DMA transfer, the DMA source and destination address registers are pointers to the next transfer address. The LDMA will update the SRC and DST addresses after each transfer. If software halts a DMA transfer by clearing the enable bit, the SRC and DST addresses will indicate the next transfer address.

When a descriptor is finished the DMA will either halt or load the next (linked) descriptor depending on the value of the LINK field in the LDMA_Chx_LINK register. After loading a linked descriptor, the descriptor registers will reflect the content of the loaded descriptor. Note that the linked descriptor must be word aligned in memory. The two least significant bits of the LDMA_CHx_LINK register are used by the LINK and LINKMODE bits. The two least significant bits of the link address are always zero.

24.3.1.7 Addressing Modes

The DMA descriptors support absolute addressing or relative addressing. When using relative addressing, the offset is relative to the current contents of the respective address registers. Regardless of the descriptor addressing modes, the address registers always indicate the absolute address. For example, when loading a descriptor using relative SRC addressing, the LDMA will add the descriptor source address (offset) to the contents of the SRCADDR register (base address). After loading, the SRCADDR register will indicate the absolute address of the loaded descriptor.

The initial descriptor must use absolute addressing. The LDMA will ignore the DSTMODE, SRCMODE, and LINKMODE bits for the initial descriptor and interpret the addresses as an absolute addresses.

Relative addressing is most useful for the link address. The initial descriptor will indicate the absolute address of the linked descriptors in memory. The linked descriptors might be an array of structures. In this case the offset between descriptors is constant and is always 4 words or 16 bytes (each descriptor has 4 words). The LINK address is not incremented or decremented after each transfer. Thus, a relative offset of 0x10 may be used for all linked descriptors.

The source and destination addresses also support relative addressing. When using relative addressing with the source or destination address registers, the LDMA adds the relative offset to the current contents of the respective address register. Since the source and destination addresses are normally incremented after each transfer, the final address will point to one unit past the last transfer. Thus, an offset of zero will give the next sequential data address.

See the example 24.4.6 2D Copy for an common use of relative addressing.

24.3.1.8 Byte Swap

Enabling byte swap reverses the endianness of the incoming source data read into the LDMA's FIFO. Byte swap is only valid for transfer sizes of word and half-word. Note that linked structure reads are not byte swapped.

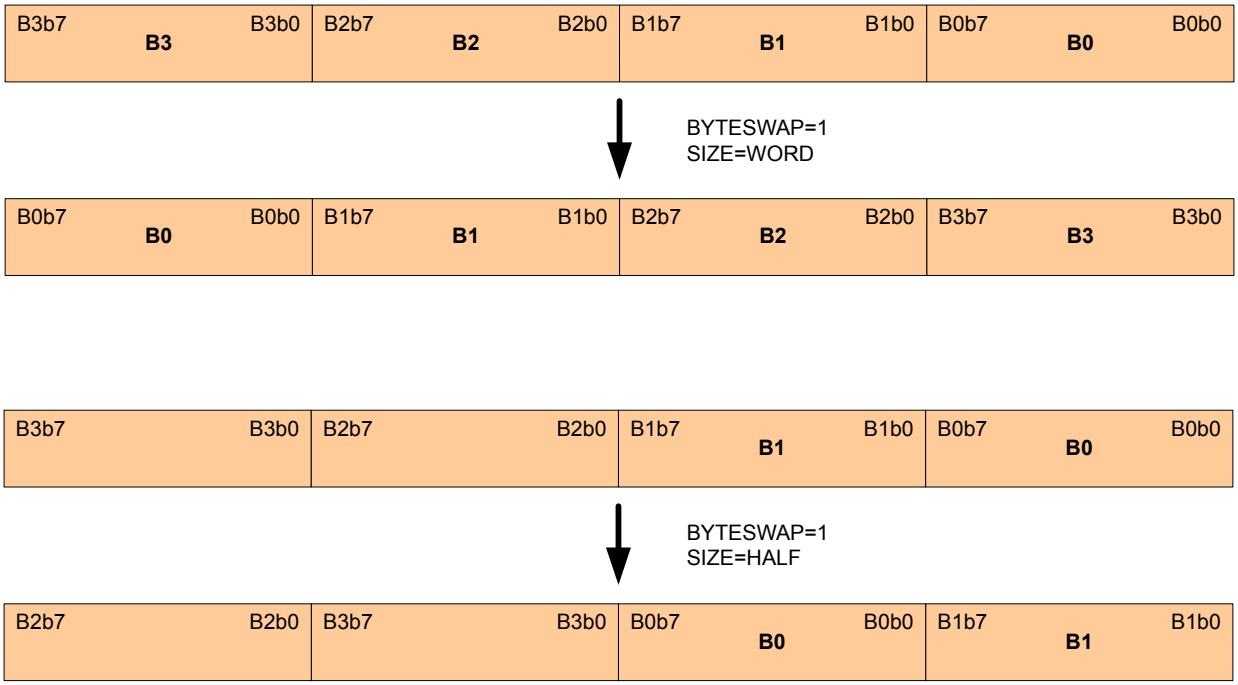


Figure 24.2. Word and Half-Word Endian Byte Swap Examples

24.3.1.9 DMA Size and Source/Destination Increment Programming

The DMA channels' SIZE, SRCINC, and DSTINC bit-fields are programmed to best utilize memory resources. They provide a means for memory packing and unpacking, as well as for matching the size of data being transmitted to or received from an IO peripheral. The following figure shows how 32-bit words of data are read from a memory source into the DMA's internal transfer FIFO, and then written out to the memory destination. The memory organization in bytes is shown as well as the first read to and write from the DMA's FIFO.

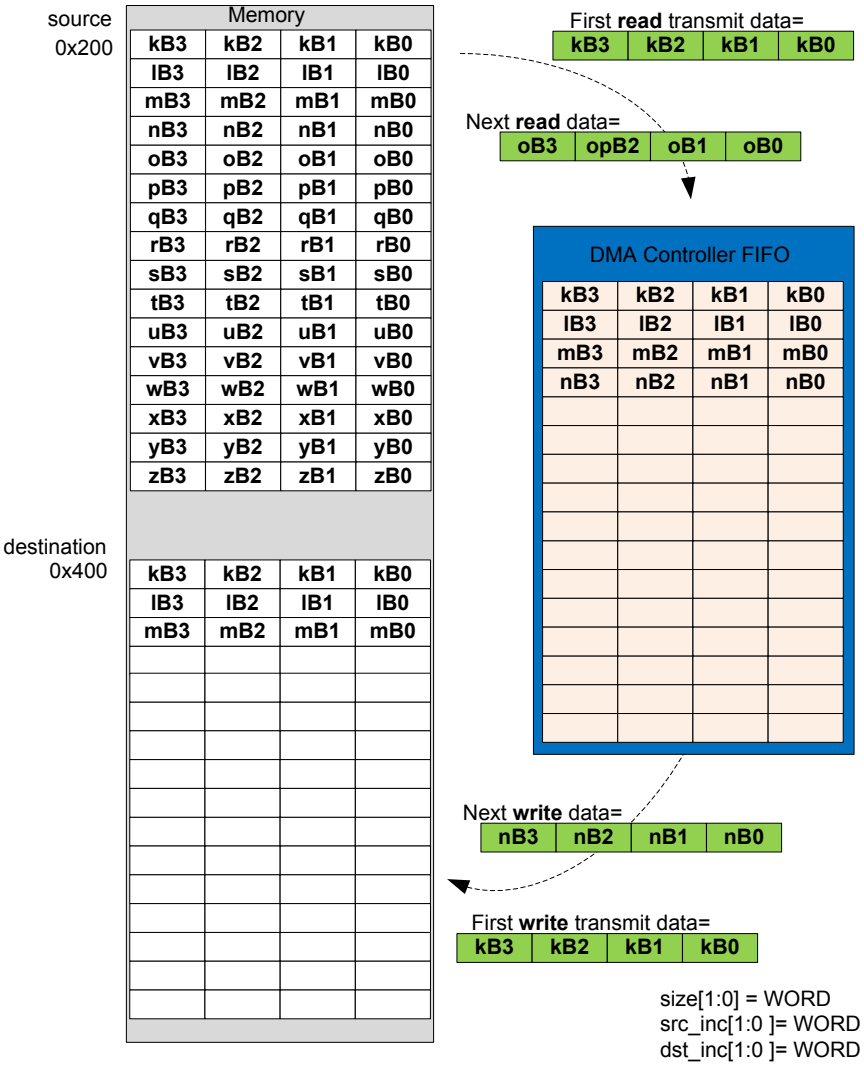


Figure 24.3. Memory-to-Memory Transfer WORD Size Example

The next example shows four variations of half-word sized transfers, with all possible combinations of half- and full-word source and destination increments. Note that when the size and source/destination increments are all configured for half-word, the resulting DMA transfer organization is equivalent to the full-word sized transfer in the previous example. The difference is that the half-word configuration requires twice as many DMA transfers.

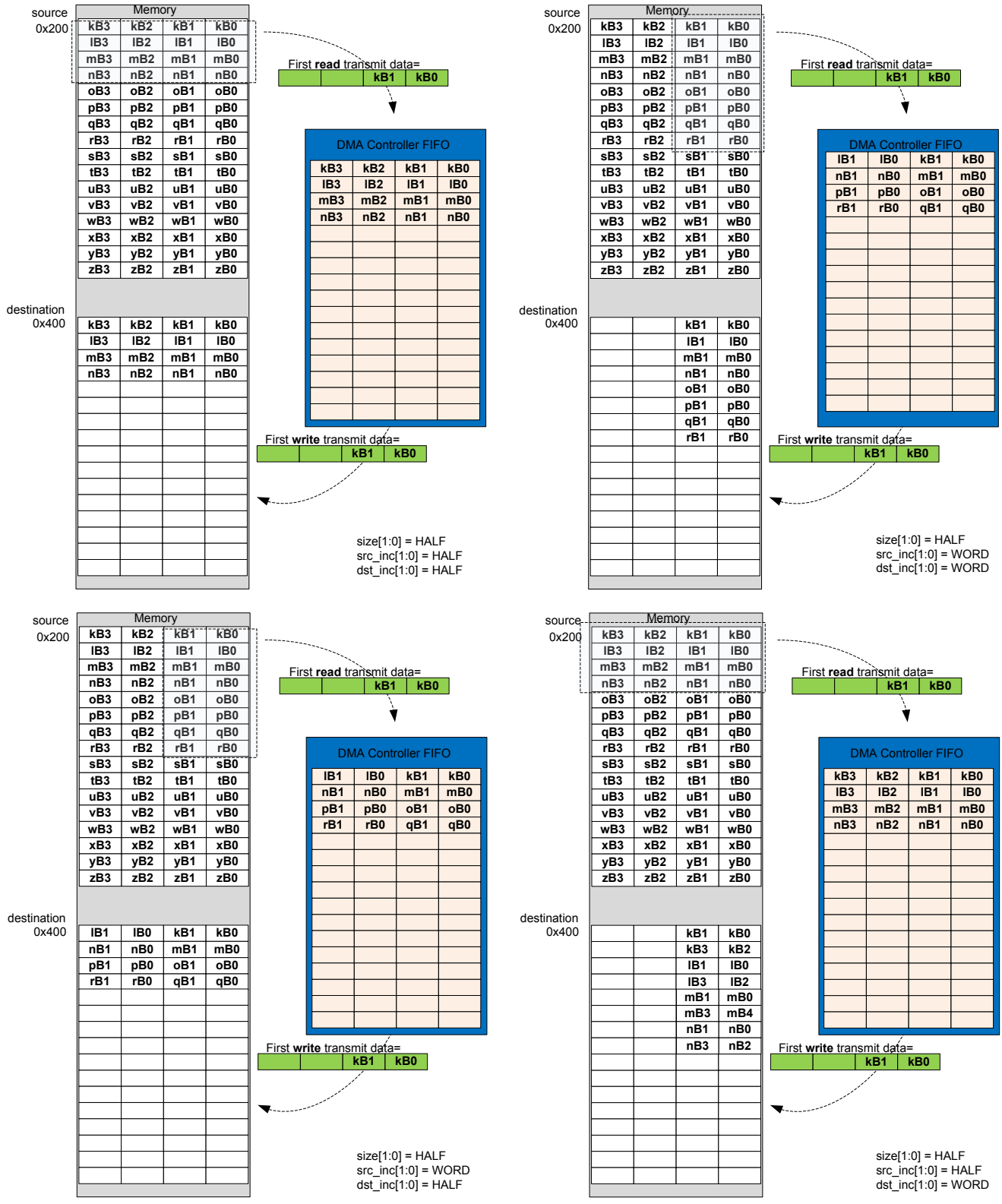


Figure 24.4. Memory-to-Memory Transfer HALF Size Examples

Fields SRCINCSIGN and DSTINCSIGN allow for address decrement. These can be used to mirror an image, for example, in the pixel copy application.

24.3.2 Channel Configuration

Each DMA channel has associated configuration and loop counter registers for controlling direction of address increment , arbitration slots, and descriptor looping.

24.3.2.1 Address Increment/Decrement

Normally DMA transfers increment the source and destination addresses after each DMA transfer. Each channel is also capable of decrementing the source and/or destination addresses after each DMA transfer. This may be useful for flipping an array or copying data from tail to head. For example, a data packet might be prepared as an array of data with increasing addresses and then transmitted from the highest address to the lowest address, from tail to head.

After reset the SRCINCSIGN and DSTINCSIGN bits in the LDMA_CHx_CFG register are cleared causing the source and destination addresses to increment after each transfer. If the SRCINCSIGN bit is set , the DMA will decrement the source address after each transfer. If the DSTINCSIGN bit in the LDMA_CHx_CFG register is set , the DMA will decrement the destination address after each transfer. Setting only one of these bits will flip the data. Setting both bits will copy from tail to head, but will not flip the data.

The SRCINCSIGN and DSTINCSIGN bits apply to all descriptors used by that channel. Software should take care to set the starting source and/or destination address to the highest data address when decrementing.

24.3.2.2 Loop Counter

Each channel has a LDMA_CHx_LOOP register that includes a loop counter field. To use looping, software should initialize the loop counter with the desired number of repetitions before enabling the transfer. A descriptor with the DECLOOPCNT bit set to TRUE will repeat the loop and decrement the loop counter until LOOPCNT = 0.

For a looping descriptor, with DECLOOPCNT=1, the LINK address in the LDMA_CHx_LINK register is used as the loop address. While LOOPCNT is greater than zero, the descriptor will execute and then the LDMA will load the next descriptor using the address specified in the LDMA_CHx_LINK register. This feature enables looping of multiple descriptors. To repeat a single descriptor, the LINK address of the descriptor should point to itself.

After LOOPCNT reaches zero, if the LINK bit in the descriptor LINK word is clear the transfer stops. If the LINK bit is set, the LDMA will load the next sequential descriptor located immediately following the looping descriptor. The behavior of the LINK bit is different for a looping descriptor. This is necessary because the LINK address is re-purposed as the loop address for a looping descriptor.

Note that LOOPCNT sets the number of repeats, not the number of iterations. The total number of loop iterations will be LOOPCNT plus 1. Normally, the LOOPCNT should be set to one or more repeats.

Also note that because there is only one LOOPCNT per channel, software intervention is required to update the LOOPCNT if a sequence of transfers contains multiple loops. It is also possible to use a write immediate DMA data transfer to update the LDMA_CHx_LOOP register.

24.3.3 Channel Select Configuration

The channel select block determines which peripheral request signal connects to each DMA channel.

This configuration is done by software through the SOURCESEL and SIGSEL fields of the LDMA_CHn_REQSEL register. SOURCESEL selects the peripheral and SIGSEL picks which DMA request signals to use from the selected peripheral. Please refer to 24.5 LDMA Source Selection Details for more information.

24.3.4 Starting a Transfer

A transfer may be started by software, a peripheral request, or a descriptor load.

Software may initiate a transfer by setting the bit for the desired channel in the LDMA_SWREQ register. In this case the channel should set SOURCESEL to NONE to prevent unintentional triggering of the channel by a peripheral.

A peripheral may trigger the channel by configuring the peripheral source and signal as described in 24.3.3 Channel Select Configuration

The LDMA may also be configured to begin a transfer immediately after a new descriptor is loaded by setting the STRUCTREQ field of the LDMA_CHx_CTRL register or descriptor word.

This configuration is done by software through the SOURCESEL and SIGSEL fields of the LDMA_CHn_REQSEL register. SOURCESEL selects the peripheral and SIGSEL picks which DMA request signals to use from the selected peripheral.

24.3.4.1 Peripheral Transfer Requests

By default peripherals issue a Single Request (SREQ) when any data is present. For peripherals with a data buffer or FIFO this occurs any time the FIFO is not empty. Upon receiving an SREQ the LDMA will perform one DMA transfer and stop till another request is made.

It is generally more efficient to wait for a peripheral to accumulate data and transfer in a burst. This both reduces overhead of the DMA engine and allows EM2 peripherals to save power by using the LDMA less often. To enable this, set the IGNORESREQ bit in the LDMA_CHx_CTRL register (or descriptor) which will cause the LDMA to ignore SREQs and wait for a full Request (REQ) signal before any data is transferred. For most peripherals with a FIFO the REQ signal is set when the FIFO is full, or a predetermined threshold has been reached. See the individual peripheral chapters for more information.

24.3.5 Managing Transfer Errors

LDMA transfer errors are normally managed using interrupts. Software should clear the ERROR flag in the bit in the LDMA_IF register and enable error interrupts by setting the ERROR bit in the LDMA_IEN register before initiating a DMA transfer.

The LDMA interrupt handler should check the ERROR flag bit in the LDMA_IF register. If the ERROR flag bit is set, it should then read the CHERROR field in the LDMA_STATUS register to determine the errant channel. The interrupt handler should reset the channel and clear the ERROR flag bit in the LDMA_IF register before returning.

24.3.6 Arbitration

While multiple channels are configured simultaneously the LDMA engine can only be actively copying data for one channel at a time. Arbitration determines which channel is being serviced at any point in time. The LDMA will choose a channel through arbitration, transfer BLOCK_SIZE elements of that channel and then arbitrate again choosing another channel to service. This allows high priority channels to be serviced while lower priority channels are in the middle of a transfer.

24.3.6.1 Arbitration Priority

There are two modes in determining priority when the controller arbitrates: fixed priority and round robin priority.

In fixed priority mode, channel 0 has the highest priority. As the channel number increases, the priority decreases. When the LDMA controller is idle or when a transfer completes, the highest priority channel with an active request is granted the transfer. This mode guarantees smallest latency for the highest priority requesters. It is best suited for systems where peak bandwidth is well below LDMA controller's maximum ability to serve. The drawback of this mode is the possibility of starvation for lowest priority requesters.

In the round robin priority mode, each active requesting channel is serviced in the order of priority. A late arriving request on a higher priority channel will not get serviced until the next round. This mode minimizes the risk of starving low-priority latency-tolerant requesters. The drawback of this mode is higher risk of starving low-latency requesters.

The NUMFIXED field in the LDMA_CTRL register determines which channels are fixed priority and which are round robin. Channels lower than NUMFIXED are fixed priority while those above it are round robin. A value of 0x0 implies all channels are round robin. A value of 0x4 implies channels 0 through 3 are fixed priority and 4 through 7 are round robin. A value of 7 implies that channels 0 through 6 are fixed and channel 7 is round robin. This is functionally equivalent to having 8 fixed priority channels.

Fixed priority channels always take priority over round robin. As long as NUMFIXED is greater than 0, there is a possibility that a higher priority channel can starve the remaining channels.

To address the drawbacks of using fixed priority or round robin priority the LDMA implements the concept of arbitration slots. This allows for channels to have high bandwidth and low latency while preventing starvation of latency tolerant low priority channels.

Each channel has a two bit ARBSLOT field in its LDM_CHx_CFG register. This field only applies to channels marked as round robin (determined by NUMFIXED). The channels in the same arbitration slot are treated equally with round robin scheduling. Channels marked with a higher arbitration slot will get serviced more frequently. By default all channels are placed in arbitration slot 1.

Every time the channels in slot 1 get serviced the channels in slot 2 get serviced twice, those in slot 4 get serviced 4 times, and those in slot 8 get serviced 7 times. The specific arbitration allocation can be seen by the following table. The highest arbitration slot is serviced every other arbitration cycle, allowing for low latency response. If there are no requests from channels in arbitration slot then that slot is immediately skipped.

Table 24.1. Arbitration Slot Order

Arbslot order	8	4	8	2	8	4	8	1	8	4	8	2	8	4
Arbslot1								1						
Arbslot2				1								1		
Arbslot4		1				1				1				1
Arbslot8	1		1		1		1		1		1		1	

The top row shows the order at which the arbitration slots are executed. The remaining part of the table shows a more visual interpretation of the arbitration order.

For example, if we have one low latency channel (CHNL0) and two latency tolerant channels (CHNL1 and CHNL2). We could use the following settings.

LDMA_CTRL.NUMFIXED = 0; set round robin for all channels.

CHNL0_CFG.ARBSLOTS = TWO;

CHNL1_CFG.ARBSLOTS = ONE;

CHNL2_CFG.ARBSLOTS = ONE;

If all channels are constantly requesting transfers, then the arbitration order is: CHNL0, CHNL1, CHNL0, CHNL2, CHNL0, CHNL1, CHNL0, CHNL2, CHNL0, etc

Note, there are no channels assigned to arbitration slot four or eight in this example, so those slots are skipped and the final sequence is ARBSLOT2, ARBSLOT1, ARBSLOT2, ARBSLOT1, etc...

Channel 1 and Channel 2 are selected in round robin order when arbitration slot 1 is executed.

If we replace the ARBSLOTS value for channel 0 with EIGHT, then the sequence would look like the following:

CHNL0, CHNL0, CHNL0, CHNL0, CHNL1, CHNL0, CHNL0, CHNL0, CHNL2, CHNL0, CHNL0, CHNL0, CHNL0, CHNL1, etc.

24.3.6.2 DMA Transfer Arbitration

In addition to the inter channel arbitration, software can configure when the controller arbitrates during a DMA transfer. This provides reduced latency to higher priority channels when configuring low priority transfers with more arbitration cycles.

The LDMA provides four bits that configure how many DMA transfers occur before it re-arbitrates. These bits are known as the BLOCKSIZE bits and they map to the arbitration rate as shown below. For example, if BLOCKSIZE = 4 then the arbitration rate is 6, that is, the controller arbitrates every 6 DMA transfers.

Table 24.2 AHB Bus Transfer Arbitration Interval on page 984 lists the arbitration rates.

Table 24.2. AHB Bus Transfer Arbitration Interval

BLOCKSIZE	Arbitrate After x DMA transfers
0	x = 1
1	x = 2
2	x = 3
3	x = 4
4	x = 6
5	x = 8
6	x = 12
7	x = 16
8	x = 24
9	x = 32
10	x = 64
11	x = 128
12	x = 256
13	x = 512
14	x = 1024
15	x = lock

Note: Software must take care not to assign a low-priority channel with a large BLOCKSIZE because this prevents the controller from servicing high-priority requests, until it re-arbitrates.

The number of DMA transfers that need to be done is specified by the user in XFERCNT. When XFERCNT > BLOCKSIZE and is not an integer multiple of BLOCKSIZE then the controller always performs sequences of BLOCKSIZE transfers until XFERCNT < BLOCKSIZE remain to be transferred. The controller performs the remaining XFERCNT transfers at the end of the DMA cycle.

Software must store the value of the BLOCKSIZE bits in the channel control data structure. See 24.3.7.1 XFER Descriptor Structure for more information about the location of the BLOCKSIZE bits in the data structure.

24.3.7 Channel Descriptor Data Structure

Each channel descriptor consists of four 32-bit words:

- CTRL - control word contains information like transfer count and block size.
- SRC - source address points to where to copy data from
- DST - destination address points to where to copy data to
- LINK - link address points to where to load the next linked descriptor

These words map directly to the LDMA_CHx_CTRL, LDMA_CHx_SRC, LDMA_CHx_DST, and LDMA_CHx_LINK registers. The usage of the SRC and DST fields may differ depending on the structure type

There are three different types of descriptor data structures: **XFER, SYNC, and WRI**

24.3.7.1 XFER Descriptor Structure

This descriptor defines a typical data transfer which may be a Normal, Link, or Loop transfer.

Only this structure type can be written directly into LDMA's registers by the CPU. All descriptors may be linked to. Please refer to the register descriptions for additional information.

For specifying XFER structure type, set STRUCTTYPE to 0. Please see the peripheral register descriptions for information on the fields in this structure.

Name	Bit Position																													
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRL	DSTMODE	SRCMODE	DSTINC		SIZE		SRCINC		IGNORESREQ	DECLOOPCNT	REQMODE	DONEIFSEN	BLOCKSIZE				BYTESWAP	XFERCNT											STRUCTREQ		STRUCTTYPE	
SRC	SRCADDR																													
DST	DSTADDR																													
LINK	LINKADDR																												LINK	LINKMODE

24.3.7.2 SYNC Descriptor Structure

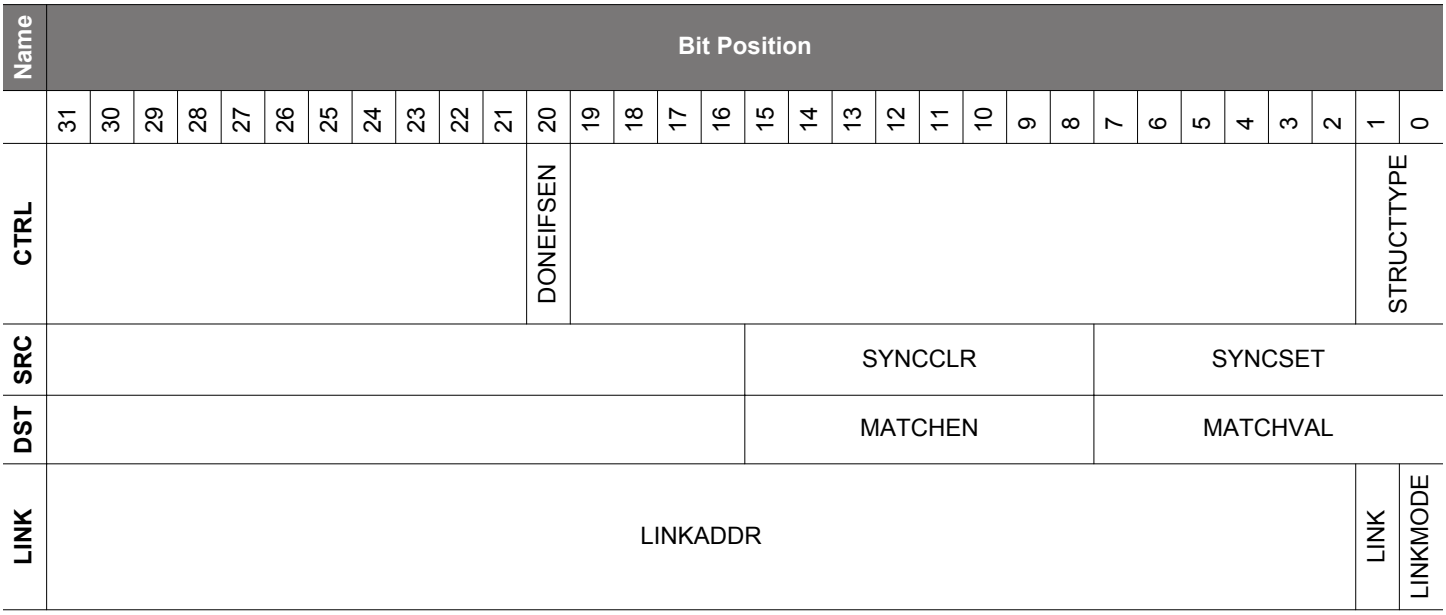
This descriptor defines an intra-channel synchronizing structure. It allows the channel to wait for some external stimulus before continuing on to the next descriptor. This structure is also used to provide stimulus to another channel to indicate that it may continue.

For example channel 1 may be configured to transfer a header into a buffer while channel 2 is simultaneously transferring data into the same structure. When channel 1 has completed it can wait for a sync signal from channel 2 before transferring the now complete buffer to a peripheral.

Synch descriptors do nothing until a condition is met. The condition is formed by the SYNCTRIG field in the LDMA_SYNC register and the MATCHEN and MATCHVAL fields of the descriptor. When (SYNCTRIG & MATCHEN) == (MATCHVAL & MATCHEN) the next descriptor is loaded. In addition to waiting for the condition a Link descriptor can set or clear bits in SYNCTRIG to meet the conditions of another channel and cause it to continue. The CPU also has the ability to set and clear the SYNCTRIG bits from software.

This structure type can only be linked in from memory.

For specifying SYNC structure type, set STRUCTTYPE to 1.



Bit	Name	Description
1:0	STRUCTTYPE	**Descriptor Type**
	This field indicates which type of descriptor this is. It must be 1 for a SYNC descriptor.	
20	DONEIFSEN	**Done if Set indicator**
	If set the interrupt flag will be set when descriptor completes.	
15:8	SYNCCLR	**Sync Trigger Clear**
	This bit-field is used to clear corresponding bits within the SYNCTRIG field of the SYNC LDMA_SYNC register. To clear a given bit, a one should be loaded to the corresponding bit. Set is given priority over clear if both corresponding bits are loaded with a one. The sync trigger clear function can only be used when loaded from a linked structure. Alternately, the user can directly write the SYNCTRIG bit-field if required.	
7:0	SYNCSET	**Sync Trigger Set**
	This bit-field is used to set corresponding bits within the SYNCTRIG bit-field. To set a given bit, a one should be loaded to the corresponding bit. Set is given priority over clear if both corresponding bits are loaded with a one. The sync trigger set function can only be used when loaded from a linked structure. Alternately, the user can directly write the SYNCTRIG bit-field if required.	
15:8	MATCHEN	**Sync Trigger Match Enable**
	This bit-field serves as the SYNCTRIG match enable. A sync match triggers the load of the next linked DMA structure as specified by link_mode, when: (SYNCTRIG & MATCHEN) == (MATCHVAL & MATCHEN).	
7:0	MATCHVAL	**Sync Trigger Match Value**

Bit	Name	Description
		This bit-field serves as the SYNCTRIG match value. A sync match triggers the load of the next linked DMA structure as specified by link_mode, when: (SYNCTRIG & MATCHEN) == (MATCHVAL & MATCHEN).

24.3.7.3 WRI Descriptor Structure

This descriptor defines a write-immediate structure. This allows a list of descriptors to write a value to a register or memory location. For example, if a channel wishes to perform two loops in a descriptor sequence a WRI may be used to program the loop count for the second loop.

This structure type can only be linked in from memory.

For specifying WRI structure type, set STRUCTTYPE to 2.

Name	Bit Position																													
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRL												DONEIFSEN																			STRUCTTYPE	
SRC	IMMVAL																															
DST	DSTADDR																															
LINK	LINKADDR																														LINK	LINKMODE

Bit	Name	Description
1:0	STRUCTTYPE	**Descriptor Type**
		This field indicates which type of descriptor this is. It must be 2 for a WRI descriptor.
20	DONEIFSEN	**Done if Set indicator**
		If set the interrupt flag will be set when descriptor completes.
31:0	IMMVAL	**Immediate Value for Write**
		This bit-field specifies the immediate data value that is to be written to the address pointed to by DSTADDR. Only one write occurs for WRI structures.
31:0	DSTADDR	**Address to write**
		This bit-field specifies the address the immediate data should be written to.

24.3.8 Interaction with the EMU

The LDMA interacts with the Energy Management Unit (EMU) to allow transfers from a low energy peripheral while in EM2.

When using the ADC in EM2 or EM3 the EMU can wake up the LDMA as needed to allow data transfers to occur.

24.3.9 Interrupts

The LDMA_IF Interrupt flag register contains one DONE bit for each channel and one combined ERROR bit. When enabled, these interrupts are available as interrupts to the M33 core. They are combined into one interrupt vector, DMA_INT. If the interrupt for the DMA is enabled in the ARM M33 core, an interrupt will be made if one or more of the interrupt flags in LDMA_IF and their corresponding bits in LDMA_IEN are set.

When a descriptor finishes execution the interrupt flag for that channel will be set if the DONEIFSEN field of the LDMA_CHx_LOOP register is set. If LINK and DONEIFSEN are both set when the descriptor completes the interrupt and the linked descriptor will be immediately loaded. When the final descriptor in a linked list (LINK = 0) is finished the interrupt flag is always set regardless of the state of DONEIFSEN.

24.3.10 Debugging

For a peripheral request DMA transfer, if software sets a bit for a channel in the LDMA_DBGHALT register then the DMA will halt during a debug halt and the SRC and DST registers in the debug window will show the transfer in progress. Otherwise, during debug halt the DMA will continue to run and complete the entire transfer causing the descriptor registers to indicate the transfer has completed.

24.4 Examples

This section provides examples of common LDMA usage. All examples assume the LDMA is in the reset state with the channel being configured disabled and LDAM_CHx_CFG, LDMA_CHx_LOOP, and LDMA_CHx_LINK cleared.

24.4.1 Single Direct Register DMA Transfer

This simple example uses only the Channel Descriptor registers directly and does not use linking. Software writes directly to the LDMA channel registers. This example does not use a memory based descriptor list.

This example is suitable for most simple transfers that are limited to transferring one block of data. It supports anything that can be done using a single descriptor. This includes endian conversion and packing/unpacking data. Channel 0 is used for this example.

The LDMA will be used to copy 127 contiguous half words (254 bytes) from 0x0 to 0x1000. It will allow arbitration every 4 transfers and is triggered by a CPU write to the LDMA_SWREQ register. The CH0 interrupt flag will be set when the transfer completes since the descriptor does not link to another descriptor.

- Configure LDMA_CH0_CTRL
  - DSTMODE = 0 (absolute)
  - SRCMODE = 0 (absolute)
  - SIZE = HALFWORD (16 bits)
  - DSTINC = 0 (1 half-word)
  - SRCINC = 0 (1 half-word)
  - DECLOOPCNT=0 (unused)
  - REQMODE = 1 (one request transfers all data)
  - BLOCKSIZE = 3 (4 transfers)
  - BYTESWAP=0 (no byte swap)
  - XFERCNT=127 (transfer 127 half words)
  - STRUCTTPYE=0 (TRANSFER)
- Write source address to LDMA_CH0_SRC register
- Write destination address to LDMA_CH0_DST register
- Configure the LDMA_CH0REQSEL register for the desired peripheral or select none for a memory-to-memory transfer
- Clear and enable interrupts.
  - Write a 1 to bit 0 of the LDMA_IFC register to clear the CH0 DONE flag
  - Write a 1 to bit 0 of the LDMA_IEN register to enable the CH0 interrupt
- Write a 1 to bit 0 of the LDMA_CHEN register to enable CH0

The REQMODE field is normally cleared to zero for a peripheral request transfer and will transfer the specified block size for each peripheral request. The REQMODE may be set to 1 for a memory-to-memory transfer or any time it is desired for a single DMA request to initiate complete transfer.

24.4.2 Descriptor Linked List

This example shows how to use a Linked List of descriptors. Each descriptor has a link address which points to the next descriptor in the list. A descriptor may be removed from the Linked list by altering the Link address of the one before it to point to the one after it. Descriptor Linked lists are useful when handling an array of buffers for communication data. For example, a bad packet can be removed from a receiver queue by simply removing the descriptor from the linked list.

Software loads the first descriptor into the DMA by writing the descriptor address to LDMA_CHx_LINK and setting the bit for that channel in the LDMA_LINKLOAD register. This method is preferred when using a linked list in memory since it treats the first descriptor just like all the others. However, it is also allowed for software to write the first descriptor directly to the LDMA registers.

In this example 4 descriptors are executed in series. the interrupt flag is set after the 2nd and 4th (last) descriptors have completed.

- Prepare a list of descriptors using the XFER structure type in RAM
- Initialize the CTRL, SRC, and DST members as desired
  - Setting STRUCTREQ in the CTRL word for descriptors 2-4 will cause them to begin transferring data as soon as they are loaded.
- Write 0x00000013 to the LINK member of all but the last descriptor
  - LINKMODE = 1 (relative addressing)
  - LINK = 1 (Link to the next descriptor)
  - LINKADDR = 0x00000010 (size of descriptor)
- Set the DONEIFSEN bit in the CTRL member of the 2nd structure so that the interrupt flag will be set when it completes
- Write 0x00000000 to the LINK member of the last descriptor
  - LINK = 0 (Do not link to the next descriptor)
  - LINKMODE = 0 (don't care)
  - LINKADDR = 0x00000000 (don't care)

Each descriptor now points to the start of the next descriptor as shown on the left in Figure 24.5 Descriptor Linked List on page 989. To remove a descriptor from the linked list modify the LINK address of the descriptor of the one before to point to the one after. For example to remove the third descriptor, add 0x00000010 to the LINK register of the second descriptor. The second descriptor will now point to the forth descriptor and skip over the third descriptor as shown on the right in Figure 24.5 Descriptor Linked List on page 989.

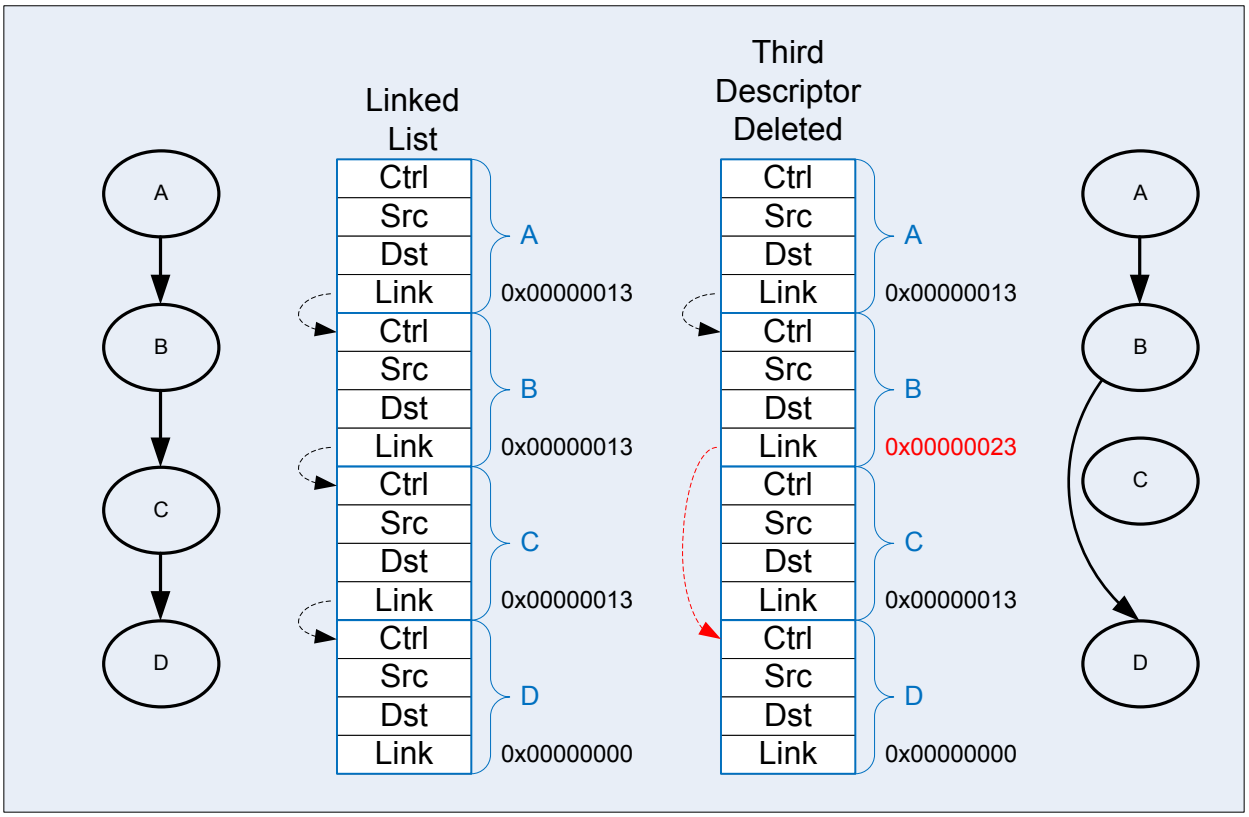


Figure 24.5. Descriptor Linked List

To start execution of the linked list of descriptors:

- Write the absolute address of the first descriptor to the LINKADR field of the LDMA_CH0_LINK register
- Set the LINK bit of LDMA_CH0_LINK register.
- Configure the LDMA_CH0REQSEL register for the desired peripheral or select none for memory-to-memory
- Clear and enable interrupts as desired
- Set bit 0 in the LDMA_LINKLOAD register to initiate loading and execution of the first descriptor

Alternatively, software can manually copy the first descriptor contents to the LDMA_CH0_CTRL, LDMA_CH0_SRC, LDMA_CH0_DST, and LDMA_CH0_LINK registers and then enable the channel in the LDMA_CHEN register.

24.4.3 Single Descriptor Looped Transfer

This example demonstrates how to use looping using a single descriptor. This method allows a single DMA transfer to be repeated a specified number of times. The looping descriptor is stored in memory and reloaded by hardware. After a specified number of iterations, the transfer stops.

CH0 is setup to copy 4 words from the ADC FIFO into a 15 word buffer at 0x1000. It repeats 4 times to fill the entire 16 word buffer. An interrupt will fire when the entire 16 words have been transferred.

Initialize the Linked descriptor in memory as follows:

- Configure CTRL member
  - DSTMODE = 0 (absolute)
  - SRCMODE = 0 (absolute)
  - SIZE = WORD
  - DSTINC = 0 (1 WORD)
  - SRCINC = 3 (0 WORDS)
  - DECLOOPCNT=1 (decrement loop count)
  - REQMODE=1 (Use XFERCNT)
  - BLOCKSIZE = 4 (4 words)
  - BYTESWAP=0 (no swap)
  - XFERCNT= 4 (4 words)
  - STRUCTTPYE=0 (TRANSFER)
  - IGNORESREQ=1 (ignore single requests)
- Write the address ADC0_SINGLEDATA register to the SRC member
- Write 0x1000 address to DST member
- Configure the LINK member
  - LINK = 0 (stop after loop)
  - MODE = 1 (relative link address)
  - LINKADDR = 0 (point to ourself)
- Configure the Channel
  - Write the desired number of repeats to the LDMA_CH0_LOOP register
  - SOURCESEL in LDMA_CH0REQSEL = ADC0 (select the ADC)
  - SIG in LDMA_CH0REQSEL = ADC0SCAN (select the scan conversion request)
- Clear and enable interrupts
- Load the descriptor using LINKLOAD as described in 24.4.2 Descriptor Linked List

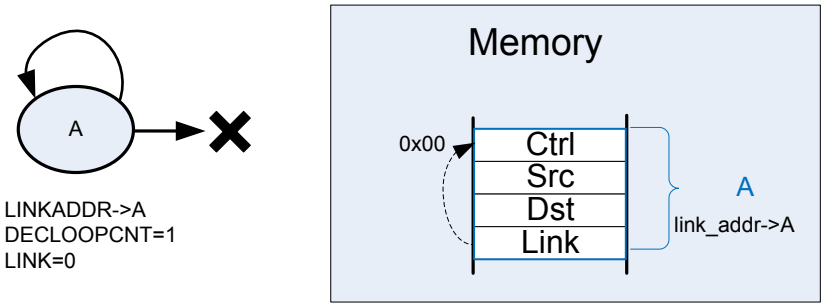


Figure 24.6. Single Descriptor Looped Transfer

Note that the looping descriptor must be stored in memory, because it must load itself from the link address in memory on each iteration.

24.4.4 Descriptor List with Looping

This example uses a descriptor list in memory with looping over multiple descriptors. This example also uses the looping feature and continues on with the next sequential descriptor after looping completes.

The descriptor list in memory is shown in figure Figure 24.7 Descriptor List with Looping on page 992. Descriptor A links to descriptor B. Descriptor B has the DECLOOPCNT bit enabled and loops back to the start of descriptor A. The LINK address of descriptor B is used for the loop address. The LINK bit is set to indicate that execution will continue after completion of looping. Once the LOOPCNT reaches zero, the LDMA will load descriptor C. Descriptor C must be located immediately following descriptor B.

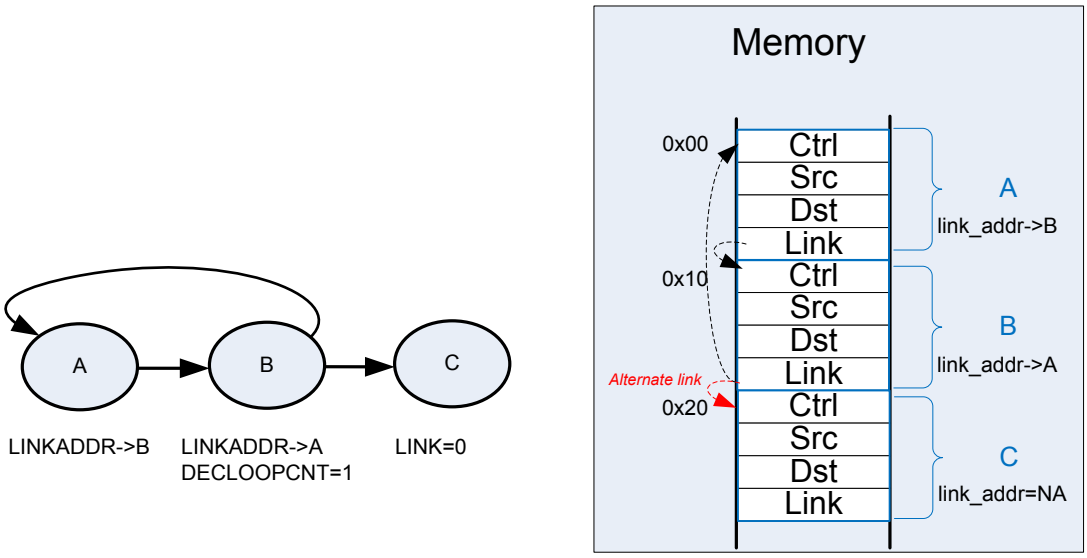


Figure 24.7. Descriptor List with Looping

Initialization is similar to the single looping descriptor with the following modifications.

- Set the LINK bit in descriptors A and B
- write the address of descriptor A into the LIKADDRESS of descriptor B
- write the address of descriptor B into the LIKADDRESS of descriptor A
- Descriptor C must be located immediately after descriptor B in memory

24.4.5 Simple Inter-Channel Synchronization

The LDMA controller features synchronization structures which allow differing channels and/or hardware events to pause a DMA sequence, and wait for a synchronizing event to restart it.

In this example DMA channel 0 and 1 are tasked with the transfer of different sets of data. Channel 0 has two transfer structures, and channel 1 just one, but channel 0 must wait until channel 1 has completed its transfer before it starts its second transfer structure.

Pausing channel 0 is accomplished by inserting a sync wait structure between the two transfer structures. This sync structure waits on SYNCTRIG[7] to be set by a sync set/clear structure which is controlled by channel 1. Sync structures do not transfer data, they can only set, clear, or wait to match the SYNCTRIG[7:0] bits. Note that sync structures cannot decrement loop counter.

```
LDMA_SYNC
    SYNCTRIG=0x0 (at time 0)

LDMA_CH0

    Structure A @ 0x00              Structure B @ 0x10                  Structure C @ 0x20
    CTRL                                CTRL                                CTRL
        STRUCTTYPE=XFER                     STRUCTTYPE=SYNC                     STRUCTTYPE=XFER
    LINK                                LINK                                LINK
        LINKADDR[29:0]=0x00000004           LINKADDR[29:0]=0x00000008           LINKADDR[29:0]=NA
        LINK=1                              LINK=1                              LINK=0
                                        DST
                                            MATCHEN=0x80
                                            MATCHVAL=0x80 (waits for SYNCTRIG[7]=1)

LDMA_CH1

    Structure Y @ 0x30              Structure Z @ 0x40

    CTRL                                CTRL
        STRUCTTYPE=XFER                     STRUCTTYPE=SYNC
    LINK                                LINK
        LINKADDR[29:0]=0x00000010           LINKADDR=NA
        LINK=1                              LINK=0
                                        SRC
                                            SRCCLR=0x0
                                            SRCSET=0x80 (sets SYNCTRIG[7])
```

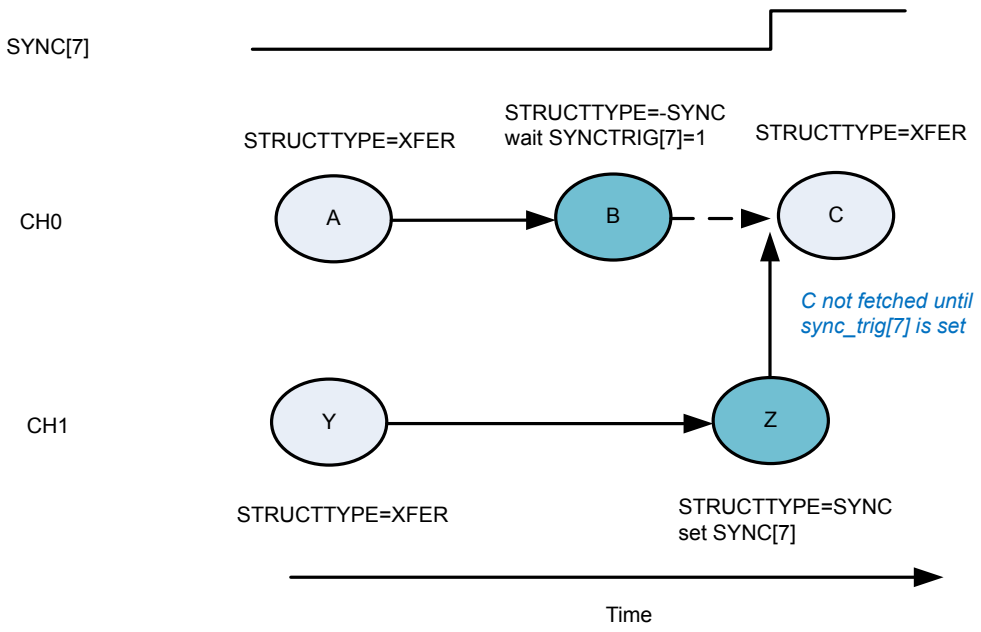


Figure 24.8. Simple Intra-channel Synchronization Example

Both A and Y effectively start at the same time. A finishes earlier, then it links to B, which waits for the SYNCTRIG[7] bit to be set before loading C. Y finishes after B is loaded, and it links to sync structure Z, which sets the SYNCTRIG[7] bit. Channel 0 responds to the trigger set by loading C for the final data transfer.

24.4.6 2D Copy

The LDMA can easily perform a 2D copy using a descriptor list with looping. This set up is visualized in Figure 24.9 2D Copy on page 995.

For an application working with graphics, this would mean the ability to copy a rectangle of a given width and height from one picture to another.

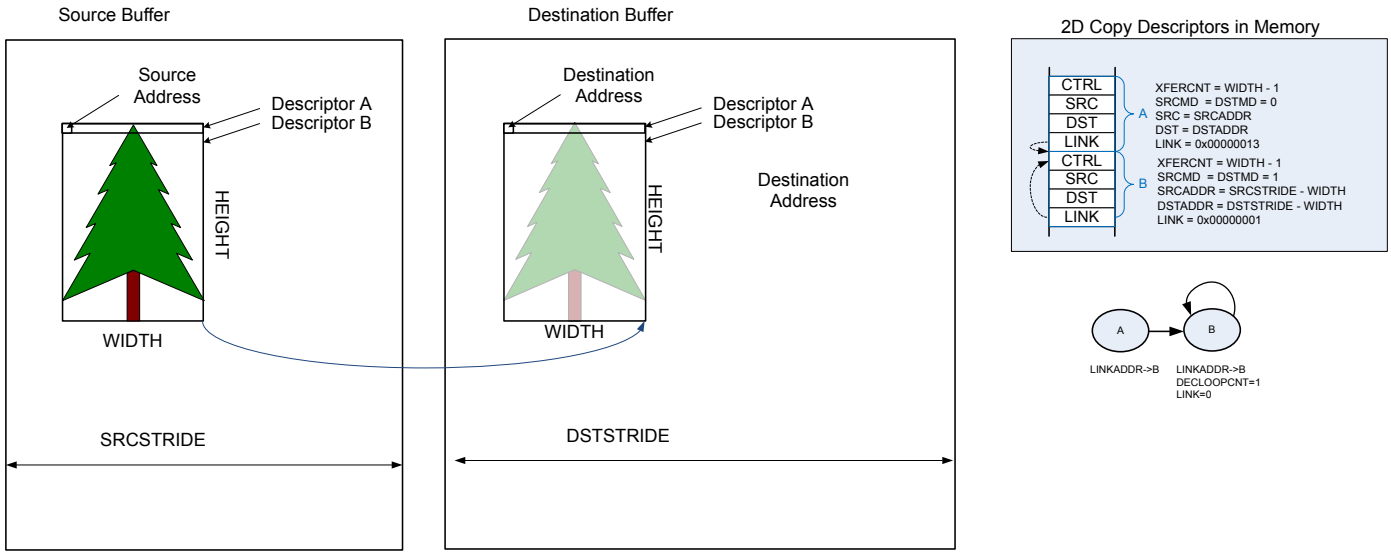


Figure 24.9. 2D Copy

The first descriptor will use absolute addressing mode and the source and destination addresses should point to the desired target addresses. The first descriptor will copy only the first row. The XFERCNT of the first descriptor is set to the desired width minus one.

- CTRL
  - XFERCNT = WIDTH - 1
  - SRCMD = 0 (absolute)
  - DSTMD = 0 (absolute)
- SRCADDR = target source address
- DSTADDR = target destination address
- LINK = 0x00000013
  - LINK=1
  - LINKMD=1
  - LINKADDR=0x00000010 (point to next descriptor)

The second descriptor will use relative addressing and the source and destination addresses are set to the desired offset. After the completion of the first descriptor, the address registers will point to the last address transferred. Thus, the width must be subtracted from the stride to get the offset. The second descriptor uses looping and the link register has not offset.

- CTRL
  - XFERCNT = WIDTH - 1
  - SRCMD = 1 (relative)
  - DSTMD = 1 (relative)
  - DECLOOPCNT = 1
- SRCADDR = desired source offset (SRCSTRIDE-WIDTH)
- DSTADDR = desired destination offset (DSTSTRIDE-WIDTH)
- LINK = 0x00000001
  - LINK=0
  - LINKMD=1 (relative)
  - LINKADDR=0x000000000 (no offset)

Because the first descriptor already transferred one row, the number of looping repeats should be the desired height minus two. Therefore, LOOPCNT should be set to HEIGHT minus two before initiating the transfer.

This same method is easily extended to copy multiple rectangles by linking descriptors together. To initialize the LDMA_CHx_LOOP register, precede each descriptor pair described above with a write immediate descriptor which writes the desired value to the LOOPCNT field of the LDMA_CHx_LOOP register.

24.4.7 Ping-Pong

Communication peripherals often use ping-pong buffers. Ping-pong buffers allow the CPU to process data in one buffer while a peripheral transmits or receives data in the other buffer.

Both transmit and receive ping-pong buffers are easily implemented using the LDMA. In either case, this requires two descriptors as shown in Figure 24.10 Infinite Ping-Pong Example on page 997. The LINKADDR field of the LINK member should point to the other descriptor. Using two adjacent descriptors and relative link addressing ensures the descriptors are easily reloadable.

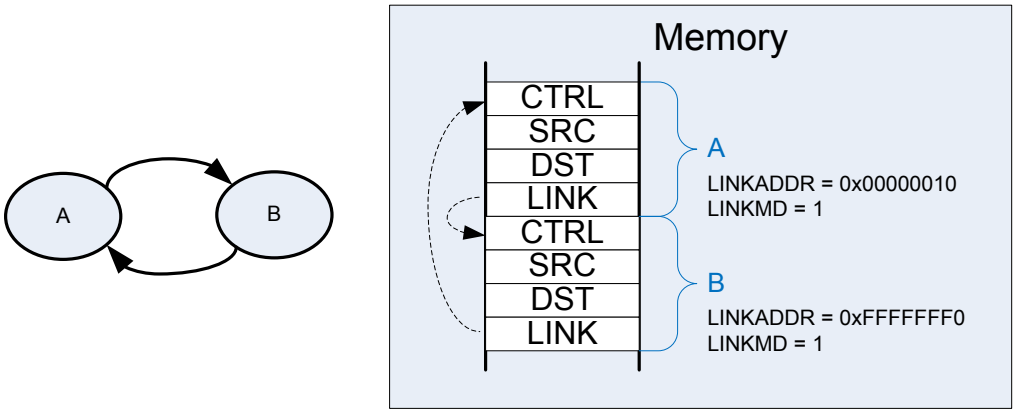


Figure 24.10. Infinite Ping-Pong Example

A **receiver** ping-pong buffer controller consists of two buffers and two descriptors stored in memory that point to the two buffers. Once initialized, as the peripheral receives data, it will fill the first buffer. Once the first buffer is full, it will link automatically to the second buffer and generate an interrupt. Software will then process the data in the first buffer while the LDMA is transferring data to the second buffer. For a receiver ping-pong buffer each descriptor should link to the other descriptor. The link bit should be set to provide infinite ping pong between the two buffers. The DONIFS bit in each descriptor should be set to generate an interrupt on the completion of each descriptor.

- Descriptor A
  - CTRL
    - DONEIFS = 1
    - other settings as desired
  - SRCADDR = peripheral source address
  - DSTADDR = memory destination address
  - LINK = 0x00000013
    - LINKADDR = 0x00000010 (next descriptor)
    - LINK = 1 (link to next descriptor)
    - LINKMD = 1 (relative addressing)
- Descriptor B
  - CTRL
    - DONEIFS = 1
    - other settings as desired
  - SRCADDR = peripheral source address
  - DSTADDR = memory destination address
  - LINK = 0xFFFFFFF3
    - LINKADDR = 0xFFFFFFF0 (previous descriptor)
    - LINK = 1 (link to previous descriptor)
    - LINKMD = 1 (relative addressing)

For **transmitter** ping-pong buffer, software will fill the first buffer and then initiate the DMA transfer. The LDMA will transmit the first buffer data while software is filling the second buffer. In this case, the two descriptors should point to each other, but not automatically

continue to the second buffer. The LINK bit should be cleared to zero. Once software has loaded the first buffer, it will use the LINK-LOAD bit to load the first descriptor and transmit the data. The DONIFS need not be set in each descriptor. The DMA will stop and then generate an interrupt at the completion of each descriptor.

- Descriptor A
  - CTRL
    - DONEIFS = 0
    - other settings as desired
  - SRCADDR = memory source address
  - DSTADDR = peripheral destination address
  - LINK = 0x00000013
    - LINKADDR = 0x00000010 (next descriptor)
    - LINK = 0 (link to next descriptor)
    - LINKMD = 1 (relative addressing)
- Descriptor B
  - CTRL
    - DONEIFS = 0
    - other settings as desired
  - SRCADDR = memory source address
  - DSTADDR = peripheral destination address
  - LINK = 0xFFFFFFF3
    - LINKADDR = 0xFFFFFFF0 (previous descriptor)
    - LINK = 0 (link to previous descriptor)
    - LINKMD = 1 (relative addressing)

24.4.8 Scatter-Gather

Scatter-Gather in general refers to a process that copies data from multiple locations scattered in memory and gathers the data to a single location in memory, or vice versa. A simple descriptor list allows data gathering. For example, data from a discontiguous list of buffers might be copied to a contiguous sequential array of buffers. The inverse is also possible when a sequential array of buffers is scattered to a discontiguous list of available buffers. See section 24.4.2 Descriptor Linked List.

Some DMAs which only have two descriptors implement scatter-gather by using one descriptor to modify the other descriptor. While it is possible to implement this same behavior using the LDMA, it is much more straight-forward to just use a simple descriptor list.

24.5 LDMA Source Selection Details

24.5.1 LDMA Source Selection Details

Table 24.3. LDMA Source Selection Details

SOURCESEL	Source Name	SIGSEL	Request Signal Name
0x1	LDMAXBAR	0x0	LDMAXBARPRSREQ0
		0x1	LDMAXBARPRSREQ1
0x2	TIMER0	0x0	TIMER0CC0
		0x1	TIMER0CC1
		0x2	TIMER0CC2
		0x3	TIMER0UFOF
0x3	TIMER1	0x0	TIMER1CC0
		0x1	TIMER1CC1
		0x2	TIMER1CC2
		0x3	TIMER1UFOF
0x4	USART0	0x0	USART0RXDATAV
		0x1	USART0RXDATAVRIGHT
		0x2	USART0TXBL
		0x3	USART0TXBLRIGHT
		0x4	USART0TXEMPTY
0x5	I2C0	0x0	I2C0RXDATAV
		0x1	I2C0TXBL
0x6	I2C1	0x0	I2C1RXDATAV
		0x1	I2C1TXBL
0xA	IADC0	0x0	IADC0IADC_SCAN
		0x1	IADC0IADC_SINGLE
0xB	MSC	0x0	MSCWDATA
0xC	TIMER2	0x0	TIMER2CC0
		0x1	TIMER2CC1
		0x2	TIMER2CC2
		0x3	TIMER2UFOF
0xD	TIMER3	0x0	TIMER3CC0
		0x1	TIMER3CC1
		0x2	TIMER3CC2
		0x3	TIMER3UFOF
0xE	TIMER4	0x0	TIMER4CC0
		0x1	TIMER4CC1
		0x2	TIMER4CC2
		0x3	TIMER4UFOF

SOURCESEL	Source Name	SIGSEL	Request Signal Name
0xF	VDAC0	0x0	VDAC0CH0_REQ
		0x1	VDAC0CH1_REQ
0x10	EUSART0	0x0	EUSART0RXFL
		0x1	EUSART0TXFL
0x11	EUSART1	0x0	EUSART1RXFL
		0x1	EUSART1TXFL
0x12	EUSART2	0x0	EUSART2RXFL
		0x1	EUSART2TXFL
0x13	LESENSE	0x0	LESENSEFIFO
0x14	LCD	0x0	LCD
0x15	MVP	0x0	MVPREQ

24.6 LDMA Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	LDMA_IPVERSION	R	DMA Channel Request Clear Register
0x004	LDMA_EN	RW	DMA Module Enable Disable Register
0x008	LDMA_CTRL	RW	DMA Control Register
0x00C	LDMA_STATUS	RH	DMA Status Register
0x010	LDMA_SYNCSWSET	W	DMA Sync Trig Sw Set Register
0x014	LDMA_SYNCSWCLR	W	DMA Sync Trig Sw Clear Register
0x018	LDMA_SYNCHWEN	RW	DMA Sync HW Trigger Enable Register
0x01C	LDMA_SYNCHWSEL	RW	DMA Sync HW Trigger Selection Register
0x020	LDMA_SYNCSTATUS	RH	DMA Sync Trigger Status Register
0x024	LDMA_CHEN	W	DMA Channel Enable Register
0x028	LDMA_CHDIS	W	DMA Channel Disable Register
0x02C	LDMA_CHSTATUS	RH	DMA Channel Status Register
0x030	LDMA_CHBUSY	RH	DMA Channel Busy Register
0x034	LDMA_CHDONE	RWH INTFLAG	DMA Channel Linking Done Register
0x038	LDMA_DBGHALT	RW	DMA Channel Debug Halt Register
0x03C	LDMA_SWREQ	W	DMA Channel Software Transfer Request
0x040	LDMA_REQDIS	RW	DMA Channel Request Disable Register
0x044	LDMA_REQPEND	RH	DMA Channel Requests Pending Register
0x048	LDMA_LINKLOAD	W	DMA Channel Link Load Register
0x04C	LDMA_REQCLEAR	W	DMA Channel Request Clear Register
0x050	LDMA_IF	RWH INTFLAG	Interrupt Flag Register
0x054	LDMA_IEN	RW	Interrupt Enable Register
0x05C	LDMA_CHx_CFG	RW	Channel Configuration Register
0x060	LDMA_CHx_LOOP	RWH	Channel Loop Counter Register
0x064	LDMA_CHx_CTRL	RWH	Channel Descriptor Control Word Register
0x068	LDMA_CHx_SRC	RWH	Channel Descriptor Source Address
0x06C	LDMA_CHx_DST	RWH	Channel Descriptor Destination Address
0x070	LDMA_CHx_LINK	RWH	Channel Descriptor Link Address
0x1000	LDMA_IPVERSION_SET	R	DMA Channel Request Clear Register
0x1004	LDMA_EN_SET	RW	DMA Module Enable Disable Register
0x1008	LDMA_CTRL_SET	RW	DMA Control Register
0x100C	LDMA_STATUS_SET	RH	DMA Status Register
0x1010	LDMA_SYNCSWSET_SET	W	DMA Sync Trig Sw Set Register
0x1014	LDMA_SYNCSWCLR_SET	W	DMA Sync Trig Sw Clear Register
0x1018	LDMA_SYNCHWEN_SET	RW	DMA Sync HW Trigger Enable Register

Offset	Name	Type	Description
0x101C	LDMA_SYNCHWSEL_SET	RW	DMA Sync HW Trigger Selection Register
0x1020	LDMA_SYNCSTATUS_SET	RH	DMA Sync Trigger Status Register
0x1024	LDMA_CHEN_SET	W	DMA Channel Enable Register
0x1028	LDMA_CHDIS_SET	W	DMA Channel Disable Register
0x102C	LDMA_CHSTATUS_SET	RH	DMA Channel Status Register
0x1030	LDMA_CHBUSY_SET	RH	DMA Channel Busy Register
0x1034	LDMA_CHDONE_SET	RWH INTFLAG	DMA Channel Linking Done Register
0x1038	LDMA_DBGHALT_SET	RW	DMA Channel Debug Halt Register
0x103C	LDMA_SWREQ_SET	W	DMA Channel Software Transfer Request
0x1040	LDMA_REQDIS_SET	RW	DMA Channel Request Disable Register
0x1044	LDMA_REQPEND_SET	RH	DMA Channel Requests Pending Register
0x1048	LDMA_LINKLOAD_SET	W	DMA Channel Link Load Register
0x104C	LDMA_REQCLEAR_SET	W	DMA Channel Request Clear Register
0x1050	LDMA_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1054	LDMA_IEN_SET	RW	Interrupt Enable Register
0x105C	LDMA_CHx_CFG_SET	RW	Channel Configuration Register
0x1060	LDMA_CHx_LOOP_SET	RWH	Channel Loop Counter Register
0x1064	LDMA_CHx_CTRL_SET	RWH	Channel Descriptor Control Word Register
0x1068	LDMA_CHx_SRC_SET	RWH	Channel Descriptor Source Address
0x106C	LDMA_CHx_DST_SET	RWH	Channel Descriptor Destination Address
0x1070	LDMA_CHx_LINK_SET	RWH	Channel Descriptor Link Address
0x2000	LDMA_IPVERSION_CLR	R	DMA Channel Request Clear Register
0x2004	LDMA_EN_CLR	RW	DMA Module Enable Disable Register
0x2008	LDMA_CTRL_CLR	RW	DMA Control Register
0x200C	LDMA_STATUS_CLR	RH	DMA Status Register
0x2010	LDMA_SYNCSWSET_CLR	W	DMA Sync Trig Sw Set Register
0x2014	LDMA_SYNCSWCLR_CLR	W	DMA Sync Trig Sw Clear Register
0x2018	LDMA_SYNCHWEN_CLR	RW	DMA Sync HW Trigger Enable Register
0x201C	LDMA_SYNCHWSEL_CLR	RW	DMA Sync HW Trigger Selection Register
0x2020	LDMA_SYNCSTATUS_CLR	RH	DMA Sync Trigger Status Register
0x2024	LDMA_CHEN_CLR	W	DMA Channel Enable Register
0x2028	LDMA_CHDIS_CLR	W	DMA Channel Disable Register
0x202C	LDMA_CHSTATUS_CLR	RH	DMA Channel Status Register
0x2030	LDMA_CHBUSY_CLR	RH	DMA Channel Busy Register
0x2034	LDMA_CHDONE_CLR	RWH INTFLAG	DMA Channel Linking Done Register
0x2038	LDMA_DBGHALT_CLR	RW	DMA Channel Debug Halt Register
0x203C	LDMA_SWREQ_CLR	W	DMA Channel Software Transfer Request

Offset	Name	Type	Description
0x2040	LDMA_REQDIS_CLR	RW	DMA Channel Request Disable Register
0x2044	LDMA_REQPEND_CLR	RH	DMA Channel Requests Pending Register
0x2048	LDMA_LINKLOAD_CLR	W	DMA Channel Link Load Register
0x204C	LDMA_REQCLEAR_CLR	W	DMA Channel Request Clear Register
0x2050	LDMA_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2054	LDMA_IEN_CLR	RW	Interrupt Enable Register
0x205C	LDMA_CHx_CFG_CLR	RW	Channel Configuration Register
0x2060	LDMA_CHx_LOOP_CLR	RWH	Channel Loop Counter Register
0x2064	LDMA_CHx_CTRL_CLR	RWH	Channel Descriptor Control Word Register
0x2068	LDMA_CHx_SRC_CLR	RWH	Channel Descriptor Source Address
0x206C	LDMA_CHx_DST_CLR	RWH	Channel Descriptor Destination Address
0x2070	LDMA_CHx_LINK_CLR	RWH	Channel Descriptor Link Address
0x3000	LDMA_IPVERSION_TGL	R	DMA Channel Request Clear Register
0x3004	LDMA_EN_TGL	RW	DMA Module Enable Disable Register
0x3008	LDMA_CTRL_TGL	RW	DMA Control Register
0x300C	LDMA_STATUS_TGL	RH	DMA Status Register
0x3010	LDMA_SYNCSWSET_TGL	W	DMA Sync Trig Sw Set Register
0x3014	LDMA_SYNCSWCLR_TGL	W	DMA Sync Trig Sw Clear Register
0x3018	LDMA_SYNCHWEN_TGL	RW	DMA Sync HW Trigger Enable Register
0x301C	LDMA_SYNCHWSEL_TGL	RW	DMA Sync HW Trigger Selection Register
0x3020	LDMA_SYNCSTATUS_TGL	RH	DMA Sync Trigger Status Register
0x3024	LDMA_CHEN_TGL	W	DMA Channel Enable Register
0x3028	LDMA_CHDIS_TGL	W	DMA Channel Disable Register
0x302C	LDMA_CHSTATUS_TGL	RH	DMA Channel Status Register
0x3030	LDMA_CHBUSY_TGL	RH	DMA Channel Busy Register
0x3034	LDMA_CHDONE_TGL	RWH INTFLAG	DMA Channel Linking Done Register
0x3038	LDMA_DBGHALT_TGL	RW	DMA Channel Debug Halt Register
0x303C	LDMA_SWREQ_TGL	W	DMA Channel Software Transfer Request
0x3040	LDMA_REQDIS_TGL	RW	DMA Channel Request Disable Register
0x3044	LDMA_REQPEND_TGL	RH	DMA Channel Requests Pending Register
0x3048	LDMA_LINKLOAD_TGL	W	DMA Channel Link Load Register
0x304C	LDMA_REQCLEAR_TGL	W	DMA Channel Request Clear Register
0x3050	LDMA_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3054	LDMA_IEN_TGL	RW	Interrupt Enable Register
0x305C	LDMA_CHx_CFG_TGL	RW	Channel Configuration Register
0x3060	LDMA_CHx_LOOP_TGL	RWH	Channel Loop Counter Register
0x3064	LDMA_CHx_CTRL_TGL	RWH	Channel Descriptor Control Word Register

Offset	Name	Type	Description
0x3068	LDMA_CHx_SRC_TGL	RWH	Channel Descriptor Source Address
0x306C	LDMA_CHx_DST_TGL	RWH	Channel Descriptor Destination Address
0x3070	LDMA_CHx_LINK_TGL	RWH	Channel Descriptor Link Address

24.7 LDMA Register Description

24.7.1 LDMA_IPVERSION - DMA Channel Request Clear Register

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0				
Access																												R				
Name																												IPVERSION				

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	IPVERSION	0x0	R	**DMA Request Clear**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

24.7.2 LDMA_EN - DMA Module Enable Disable Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																EN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	EN	0x0	RW	**LDMA module enable and disable register**
	The ENABLE bit enables the module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit.			

24.7.3 LDMA_CTRL - DMA Control Register

Offset	Bit Position																															
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0			0x1E																												
Access	RW			RW																												
Name	CORERST			NUMFIXED																												

Bit	Name	Reset	Access	Description
31	CORERST	0x0	RW	**Reset DMA controller**
	Trigger a reset of the LDMA controller core without losing register configuration			
30:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28:24	NUMFIXED	0x1E	RW	**Number of Fixed Priority Channels**
	This field defines the number of Fixed Priority Arbitration channels. Channels CH0 though CH(n-1) are fixed, and channels CH(n) through CH7 are round robin, where n is the field value. The reset value will give all fixed channels.			
23:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

24.7.4 LDMA_STATUS - DMA Status Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x8										0x10							0x0				0x0						0x0	0x0
Access				R										R							R				R						R	R
Name				CHNUM										FIFOLEVEL							CHERROR				CHGRANT						ANYREQ	ANYBUSY

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28:24	CHNUM	0x8	R	**Number of Channels**
	The value of CHNUM always reads the total number of channels present for this instance of the DMA controller module.			
23:21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20:16	FIFOLEVEL	0x10	R	**FIFO Level**
	The value of FIFOLEVEL indicates the number of entries currently in the FIFO. (Note when all channels are disabled, this register will read the total number of entries in the FIFO.)			
15:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12:8	CHERROR	0x0	R	**Errant Channel Number**
	When the ERROR flag is set in the LDMA_IF register, the CHERROR field will indicate the most recent channel to have a transfer error.			
7:3	CHGRANT	0x0	R	**Granted Channel Number**
	The value of this field indicates the currently active channel or last active channel. Note that the reset value for this field is zero.			
2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	ANYREQ	0x0	R	**Any DMA Channel Request Pending**
	The value of this bit will be TRUE (1) if any requests are pending			
0	ANYBUSY	0x0	R	**Any DMA Channel Busy**
	The value of this bit will be TRUE (1) if one or more DMA channels are actively transferring data			

24.7.5 LDMA_SYNCSWSET - DMA Sync Trig Sw Set Register

Offset	Bit Position																															
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W							
Name																									SYNCSWSET							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	SYNCSWSET	0x0	W	**DMA SYNC Software Trigger Set**
	Sets the corresponding bit in the SYNCSTATUS.SYNCTRIG field to value 1.			

24.7.6 LDMA_SYNCSWCLR - DMA Sync Trig Sw Clear Register

Offset	Bit Position																															
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W							
Name																									SYNCSWCLR							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	SYNCSWCLR	0x0	W	**DMA SYNC Software Trigger Clear**
	Clears the corresponding bit in the SYNCSTATUS.SYNCTRIG field to value 0.			

24.7.7 LDMA_SYNCHWEN - DMA Sync HW Trigger Enable Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																0x0							
Access									RW																RW							
Name									SYNCCLREN																SYNCSETEN							

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:16	SYNCCLREN	0x0	RW	**Hardware Sync Trigger Clear Enable**
	Enables the corresponding bit in the SYNCSTATUS.SYNCTRIG field to be cleared by PRS channel 7-0, mapping to bits [23:16].			
15:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	SYNCSETEN	0x0	RW	**Hardware Sync Trigger Set Enable**
	Enables the corresponding bit in the SYNCSTATUS.SYNCTRIG field to be set by PRS channel 7-0, mapping to bits [7:0].			

24.7.8 LDMA_SYNCHWSEL - DMA Sync HW Trigger Selection Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset												0x0																0x0				
Access												RW																RW				
Name												SYNCCLREDGE																SYNCSETEDGE				

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:16	SYNCCLREDGE	0x0	RW	**Hardware Sync Trigger Clear Edge Select**

Select rising or falling edge detection on PRS to clear trigger.

Value	Mode	Description
0	RISE	Use rising edge detection
1	FALL	Use falling edge detection

Bit	Name	Reset	Access	Description
15:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	SYNCSETEDGE	0x0	RW	**Hardware Sync Trigger Set Edge Select**

Select rising or falling edge detection on PRS to set trigger.

Value	Mode	Description
0	RISE	Use rising edge detection
1	FALL	Use falling edge detection

24.7.9 LDMA_SYNCSTATUS - DMA Sync Trigger Status Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0				
Access																												R				
Name																												SYNCTRIG				

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	SYNCTRIG	0x0	R	**sync trig status**
	Reflects the status of setting and clearing by software (SYNCSWSET/SYNCSWCLR), hardware (PRS), and loading SYNC structures. Setting a bit always takes prescedence over clearing.			

24.7.10 LDMA_CHEN - DMA Channel Enable Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0				
Access																												W				
Name																												CHEN				

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	CHEN	0x0	W	**Channel Enables**
	Setting one of these bits will enable the respective DMA channel, writing zeros has no effect			

24.7.11 LDMA_CHDIS - DMA Channel Disable Register

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W							
Name																									CHDIS							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	CHDIS	0x0	W	**DMA Channel disable**
	Setting one of these bits will disable of the channels, wrting zero has no effect			

24.7.12 LDMA_CHSTATUS - DMA Channel Status Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									CHSTATUS							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	CHSTATUS	0x0	R	**DMA Channel Status**
	The value of this bit will be TRUE (1) if one or more DMA channels are enabled			

24.7.13 LDMA_CHBUSY - DMA Channel Busy Register

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0				
Access																												R				
Name																												BUSY				

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	BUSY	0x0	R	**Channels Busy** The bits of this field read 1 when the corresponding channel is busy.

24.7.14 LDMA_CHDONE - DMA Channel Linking Done Register

Offset	Bit Position																													
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																									RW	RW	RW	RW	RW	RW	RW	RW
Name																									CHDONE7	CHDONE6	CHDONE5	CHDONE4	CHDONE3	CHDONE2	CHDONE1	CHDONE0

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	CHDONE7	0x0	RW	**DMA Channel Link done intr flag**
	Each DMA channel sets the corresponding bit in this register when the entire transfer is done.			
6	CHDONE6	0x0	RW	**DMA Channel Link done intr flag**
	Each DMA channel sets the corresponding bit in this register when the entire transfer is done.			
5	CHDONE5	0x0	RW	**DMA Channel Link done intr flag**
	Each DMA channel sets the corresponding bit in this register when the entire transfer is done.			
4	CHDONE4	0x0	RW	**DMA Channel Link done intr flag**
	Each DMA channel sets the corresponding bit in this register when the entire transfer is done.			
3	CHDONE3	0x0	RW	**DMA Channel Link done intr flag**
	Each DMA channel sets the corresponding bit in this register when the entire transfer is done.			
2	CHDONE2	0x0	RW	**DMA Channel Link done intr flag**
	Each DMA channel sets the corresponding bit in this register when the entire transfer is done.			
1	CHDONE1	0x0	RW	**DMA Channel Link done intr flag**
	Each DMA channel sets the corresponding bit in this register when the entire transfer is done.			
0	CHDONE0	0x0	RW	**DMA Channel Link done intr flag**
	Each DMA channel sets the corresponding bit in this register when the entire transfer is done.			

24.7.15 LDMA_DBGHALT - DMA Channel Debug Halt Register

Offset	Bit Position																													
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0				
Access																												RW				
Name																												DBGHALT				

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	DBGHALT	0x0	RW	**DMA Debug Halt**
	Setting one of these bits will mask the corresponding DMA channel's peripheral request when debugging and the CPU is halted. This may be useful for debugging DMA software.			

24.7.16 LDMA_SWREQ - DMA Channel Software Transfer Request

Offset	Bit Position																													
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0				
Access																												W				
Name																												SWREQ				

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	SWREQ	0x0	W	**Software Transfer Requests**
	Setting one of these bits will trigger a DMA transfer for the corresponding channel. Writing zeros has no effect.			

24.7.17 LDMA_REQDIS - DMA Channel Request Disable Register

Offset	Bit Position																													
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									REQDIS							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	REQDIS	0x0	RW	**DMA Request Disables**
	Setting one of these bits will disable peripheral requests for the corresponding channel. When cleared any pending peripheral requests will be serviced.			

24.7.18 LDMA_REQPEND - DMA Channel Requests Pending Register

Offset	Bit Position																													
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									REQPEND							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	REQPEND	0x0	R	**DMA Requests Pending**
	When a DMA channel has a pending peripheral request the corresponding REQPEND bit will read 1.			

24.7.19 LDMA_LINKLOAD - DMA Channel Link Load Register

Offset	Bit Position																															
0x048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W							
Name																									LINKLOAD							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	LINKLOAD	0x0	W	**DMA Link Loads**
	Setting one of these bits will force the corresponding DMA channel to load the next DMA structure and enable the channel. This empowers software to step through a sequence of descriptors.			

24.7.20 LDMA_REQCLEAR - DMA Channel Request Clear Register

Offset	Bit Position																															
0x04C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W							
Name																									REQCLEAR							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	REQCLEAR	0x0	W	**DMA Request Clear**
	Setting one of these bits will clear any internally registered transfer requests for the corresponding channel.			

24.7.21 LDMA_IF - Interrupt Flag Register

Offset	Bit Position																													
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access	RW																								RW	RW	RW	RW	RW	RW	RW	RW
Name	ERROR																								DONE7	DONE6	DONE5	DONE4	DONE3	DONE2	DONE1	DONE0

Bit	Name	Reset	Access	Description
31	ERROR	0x0	RW	**Error Flag**
	Set to 1 on an Error			
30:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	DONE7	0x0	RW	**DMA Structure Operation Done**
	When a channel completes a transfer or sync operation, the corresponding DONE bit is set in the LDMA_IF register.			
6	DONE6	0x0	RW	**DMA Structure Operation Done**
	When a channel completes a transfer or sync operation, the corresponding DONE bit is set in the LDMA_IF register.			
5	DONE5	0x0	RW	**DMA Structure Operation Done**
	When a channel completes a transfer or sync operation, the corresponding DONE bit is set in the LDMA_IF register.			
4	DONE4	0x0	RW	**DMA Structure Operation Done**
	When a channel completes a transfer or sync operation, the corresponding DONE bit is set in the LDMA_IF register.			
3	DONE3	0x0	RW	**DMA Structure Operation Done**
	When a channel completes a transfer or sync operation, the corresponding DONE bit is set in the LDMA_IF register.			
2	DONE2	0x0	RW	**DMA Structure Operation Done**
	When a channel completes a transfer or sync operation, the corresponding DONE bit is set in the LDMA_IF register.			
1	DONE1	0x0	RW	**DMA Structure Operation Done**
	When a channel completes a transfer or sync operation, the corresponding DONE bit is set in the LDMA_IF register.			
0	DONE0	0x0	RW	**DMA Structure Operation Done**
	When a channel completes a transfer or sync operation, the corresponding DONE bit is set in the LDMA_IF register.			

24.7.22 LDMA_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																											0x0				
Access	RW																											RW				
Name	ERROR																											CHDONE				

Bit	Name	Reset	Access	Description
31	ERROR	0x0	RW	**Enable or disable the error interrupt**
	Enables the AHB bus error interrupt			
30:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	CHDONE	0x0	RW	**Enable or disable the done interrupt**
	Enables done interrupts			

24.7.23 LDMA_CHx_CFG - Channel Configuration Register

Offset	Bit Position																													
0x05C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0	0x0			0x0																	
Access											RW	RW			RW																	
Name											DSTINCSIGN	SRCINCSIGN			ARBSLOTS																	

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	DSTINCSIGN	0x0	RW	**Destination Address Increment Sign**
	0: Increment destination address, 1: Decrement destination address			
	Value	Mode		Description
	0	POSITIVE		Increment destination address
	1	NEGATIVE		Decrement destination address
20	SRCINCSIGN	0x0	RW	**Source Address Increment Sign**
	0: Increment source address, 1: Decrement source address			
	Value	Mode		Description
	0	POSITIVE		Increment source address
	1	NEGATIVE		Decrement source address
19:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17:16	ARBSLOTS	0x0	RW	**Arbitration Slot Number Select**
	For channels using round robin arbitration, this bit-field is used to select the number of slots in the round robin queue.			
	Value	Mode		Description
	0	ONE		One arbitration slot selected
	1	TWO		Two arbitration slots selected
	2	FOUR		Four arbitration slots selected
	3	EIGHT		Eight arbitration slots selected
15:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

24.7.24 LDMA_CHx_LOOP - Channel Loop Counter Register

Offset	Bit Position																													
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									LOOPCNT							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	LOOPCNT	0x0	RW	**Linked Structure Sequence Loop Counter**
	This bit-field specifies the number of iterations when using looping descriptors. Software should write to LOOPCNT before using a looping descriptor.			

24.7.25 LDMA_CHx_CTRL - Channel Descriptor Control Word Register

Offset	Bit Position																															
0x064	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0		0x0		0x0		0x0	0x0	0x0	0x0	0x0				0x0	0x0											0x0		0x0	
Access	R	R	RW		RW		RW		RW	RW	RW	RW	RW				RW	RW											R		RW	
Name	DSTMODE	SRCMODE	DSTINC		SIZE		SRCINC		IGNORESREQ	DECLOOPCNT	REQMODE	DONEIEN	BLOCKSIZE				BYTESWAP	XFERCNT											STRUCTREQ		STRUCTTYPE	

Bit	Name	Reset	Access	Description
31	DSTMODE	0x0	R	**Destination Addressing Mode**

This field specifies the destination addressing mode of linked descriptors. After loading a linked descriptor, reading this field will indicate the destination addressing mode of the linked descriptor. Note that the first descriptor always uses absolute addressing mode.

Value	Mode	Description
0	ABSOLUTE	The DSTADDR field of LDMA_CHx_DST contains the absolute address of the destination data.
1	RELATIVE	The DSTADDR field of LDMA_CHx_DST contains the relative offset of the destination data.

Bit	Name	Reset	Access	Description
30	SRCMODE	0x0	R	**Source Addressing Mode**

This field specifies the source addressing mode of linked descriptors. After loading a linked descriptor, reading this field will indicate the source addressing mode of the linked descriptor. Note that the first descriptor always uses absolute addressing mode.

Value	Mode	Description
0	ABSOLUTE	The SRCADDR field of LDMA_CHx_SRC contains the absolute address of the source data.
1	RELATIVE	The SRCADDR field of LDMA_CHx_SRC contains the relative offset of the source data.

Bit	Name	Reset	Access	Description
29:28	DSTINC	0x0	RW	**Destination Address Increment Size**

This bit-field specifies the stride or number of unit data addresses to increment the destination address after each unit of data is transferred. The unit data width is controlled by the SIZE bit-field and can be a byte, half-word or word.

Value	Mode	Description
0	ONE	Increment destination address by one unit data size after each write
1	TWO	Increment destination address by two unit data sizes after each write
2	FOUR	Increment destination address by four unit data sizes after each write
3	NONE	Do not increment the destination address. Writes are made to a fixed destination address, for example writing to a FIFO.

Bit	Name	Reset	Access	Description
27:26	SIZE	0x0	RW	**Unit Data Transfer Size**

This field specifies the size of data transferred.

Value	Mode	Description
0	BYTE	Each unit transfer is a byte
1	HALFWORD	Each unit transfer is a half-word
2	WORD	Each unit transfer is a word

Bit	Name	Reset	Access	Description
25:24	SRCINC	0x0	RW	**Source Address Increment Size**

This bit-field specifies the stride or number of unit data addresses to increment the source address after each unit of data is transferred. The unit data width is controlled by the SIZE bit-field and can be a byte, half-word or word.

Value	Mode	Description
0	ONE	Increment source address by one unit data size after each read
1	TWO	Increment source address by two unit data sizes after each read
2	FOUR	Increment source address by four unit data sizes after each read
3	NONE	Do not increment the source address. In this mode reads are made from a fixed source address, for example reading FIFO.

Bit	Name	Reset	Access	Description
23	IGNORESREQ	0x0	RW	**Ignore Sreq**

The channel arbiter will ignore single requests (SREQ) and only respond to multiple requests (REQ) when this bit is set.

Bit	Name	Reset	Access	Description
22	DECLOOPCNT	0x0	RW	**Decrement Loop Count**

When using looping, setting this bit will decrement the LOOPCNT field in the LDMA_CHx_LOOP register after each descriptor execution.

Bit	Name	Reset	Access	Description
21	REQMODE	0x0	RW	**DMA Request Transfer Mode Select**

Selects the DMA Request Transfer mode.

Value	Mode	Description
0	BLOCK	The LDMA transfers one BLOCKSIZE per transfer request.
1	ALL	One transfer request transfers all units as defined by the XFRCNT field.

Bit	Name	Reset	Access	Description
20	DONEIEN	0x0	RW	**DMA Operation Done Interrupt Flag Set En**

Setting this bit will set the interrupt flag when the transfer is done, or linked in the case where the LINK bit is set, or synchronized in the case of a SYNC transfer.

Bit	Name	Reset	Access	Description
19:16	BLOCKSIZE	0x0	RW	**Block Transfer Size**

This bit-field controls the number of unit data transfers per arbitration cycle

Value	Mode	Description
0	UNIT1	One unit transfer per arbitration
1	UNIT2	Two unit transfers per arbitration
2	UNIT3	Three unit transfers per arbitration
3	UNIT4	Four unit transfers per arbitration

Bit	Name	Reset	Access	Description
	4	UNIT6		Six unit transfers per arbitration
	5	UNIT8		Eight unit transfers per arbitration
	7	UNIT16		Sixteen unit transfers per arbitration
	9	UNIT32		32 unit transfers per arbitration
	10	UNIT64		64 unit transfers per arbitration
	11	UNIT128		128 unit transfers per arbitration
	12	UNIT256		256 unit transfers per arbitration
	13	UNIT512		512 unit transfers per arbitration
	14	UNIT1024		1024 unit transfers per arbitration
	15	ALL		Transfer all units as specified by the XFRCNT field
15	BYTESWAP	0x0	RW	**Endian Byte Swap**
	For word and half-word transfers, setting this bit will swap all bytes of each word or half-word.			
14:4	XFERCNT	0x0	RW	**DMA Unit Data Transfer Count**
	Specifies number of unit data (words, half-words, or bytes) to transfer, as determined by the SIZE field. The value written should be one less than the desired transfer count.			
3	STRUCTREQ	0x0	R	**Structure DMA Transfer Request**
	Structure Transfer Request			
2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	STRUCTTYPE	0x0	RW	**DMA Structure Type**
	DMA Structure type			

Value	Mode	Description
0	TRANSFER	DMA transfer structure type selected.
1	SYNCHRONIZE	Synchronization structure type selected.
2	WRITE	Write immediate value structure type selected.

24.7.26 LDMA_CHx_SRC - Channel Descriptor Source Address

Offset	Bit Position																															
0x068	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	SRCADDR																															

Bit	Name	Reset	Access	Description
31:0	SRCADDR	0x0	RW	**Source Data Address**
	Writing to this register sets the source address. Reading from this register during a DMA transfer will indicate the next source read address. The value of this register is incremented or decremented with each source read.			

24.7.27 LDMA_CHx_DST - Channel Descriptor Destination Address

Offset	Bit Position																															
0x06C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	DSTADDR																															

Bit	Name	Reset	Access	Description
31:0	DSTADDR	0x0	RW	**Destination Data Address**
	Writing to this register sets the destination address. Reading from this register during a DMA transfer will indicate the next destination write address. This value of this register is incremented or decremented with each destination write.			

24.7.28 LDMA_CHx_LINK - Channel Descriptor Link Address

Offset	Bit Position																													
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																														0x0	0x0
Access	RW																														RW	R
Name	LINKADDR																														LINK	LINKMODE

Bit	Name	Reset	Access	Description
31:2	LINKADDR	0x0	RW	**Link Structure Address**
	To use linking, write the address of the the first linked descriptor to this register. When a linked descriptor is loaded, it may also be linked to another descriptor. Reading this register will reflect the address of the next linked descriptor.			
1	LINK	0x0	RW	**Link Next Structure**
	After completing the initial transfer, if this bit is NOT set, the DMA will load the next linked descriptor. If the next linked descriptor also has this bit set, the DMA will load the next linked descriptor.			
0	LINKMODE	0x0	R	**Link Structure Addressing Mode**
	This field specifies the addressing mode of linked descriptors. After loading a linked descriptor, reading this field will indicate the addressing mode of the loaded linked descriptor. Note that the first descriptor always uses absolute addressing mode.			

Value	Mode	Description
0	ABSOLUTE	The LINKADDR field of LDMA_CHx_LINK contains the absolute address of the linked descriptor.
1	RELATIVE	The LINKADDR field of LDMA_CHx_LINK contains the relative offset of the linked descriptor.

24.8 LDMAXBAR Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	LDMAXBAR_IPVERSION	R	IP Veersion ID
0x004	LDMAXBAR_CHx_REQSEL	RW	Channel Peripheral Request Select Reg...
0x1000	LDMAXBAR_IPVERSION_SET	R	IP Veersion ID
0x1004	LDMAXBAR_CHx_REQSEL_SET	RW	Channel Peripheral Request Select Reg...
0x2000	LDMAXBAR_IPVERSION_CLR	R	IP Veersion ID
0x2004	LDMAXBAR_CHx_REQSEL_CLR	RW	Channel Peripheral Request Select Reg...
0x3000	LDMAXBAR_IPVERSION_TGL	R	IP Veersion ID
0x3004	LDMAXBAR_CHx_REQSEL_TGL	RW	Channel Peripheral Request Select Reg...

24.9 LDMAXBAR Register Description

24.9.1 LDMAXBAR_IPVERSION - IP Veersion ID

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x6																															
Access	R																															
Name	IPVERSION																															

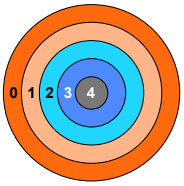
Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x6	R	**IP Version ID**
	ID indicating version of IP			

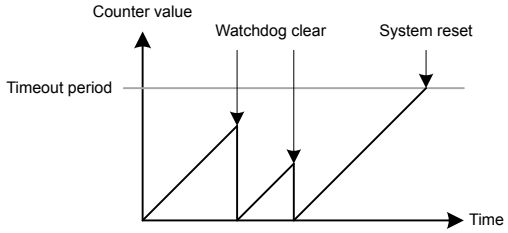
24.9.2 LDMAXBAR_CHx_REQSEL - Channel Peripheral Request Select Reg...

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0																		0x0			
Access											RW																		RW			
Name											SOURCESEL																		SIGSEL			

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:16	SOURCESEL	0x0	RW	**Source Select**
	Select input source to DMA channel.			
15:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	SIGSEL	0x0	RW	**Signal Select**
	Select input signal to DMA channel.			

25. WDOG - Watch Dog Timer





Quick Facts

What?

The WDOG (Watchdog Timer) resets the system in case of a fault condition, and can be enabled in all energy modes as long as the low frequency clock source is available.

Why?

If a software failure or external event renders the MCU unresponsive, a Watchdog timeout will reset the system to a known, safe state.

How?

An enabled Watchdog Timer implements a configurable timeout period. If the CPU fails to re-start the Watchdog Timer before it times out, a full system reset will be triggered. The Watchdog consumes insignificant power, and allows the device to remain safely in low energy modes for up to 256 seconds at a time.

25.1 Introduction

The purpose of the watchdog timer is to generate a reset in case of a system failure to increase application reliability. The failure can be caused by a variety of events, such as an ESD pulse or a software failure.

25.2 Features

- Clock input from selectable oscillators
  - Internal 32 kHz LFRCO oscillator
  - Internal 1 kHz ULFRCO oscillator
  - External 32.768 kHz LFXO XTAL oscillator
  - HCLK divided by 1024
- Configurable timeout period from 9 to 256k watchdog clock cycles
- Individual selection to keep running or freeze when entering EM1 Sleep, EM2 DeepSleep, or EM3 Stop
- Selection to keep running or freeze when entering debug mode
- Selection to block the CPU from entering Energy Mode 4
- Configurable warning interrupt at 25%, 50%, or 75% of the timeout period
- Configurable window interrupt at 12.5%, 25%, 37.5%, 50%, 62.5%, 75%, 87.5% of the timeout period
- Timeout interrupt
- PRS as a watchdog clear
- Interrupt for the event where a PRS rising edge is absent before a software reset

25.3 Functional Description

The watchdog is enabled by setting the EN bit in WDOGn_EN. When enabled, the watchdog counts up to the period value configured through the PERSEL field in WDOGn_CFG. If the watchdog timer is not cleared to 0 (by writing a 1 to the CLEAR bit in WDOGn_CMD) before the period is reached, the chip is reset. If a timely clear command is issued, the timer starts counting up from 0 again. The watchdog can optionally be locked by writing anything other than UNLOCK code in WDOGn_LOCK. Once locked, it cannot be disabled or reconfigured by software.

Note: If the WDOG is configured to halt during EM1, EM2, or EM3 and EM1/EM2/EM3 is entered on the clock cycle before the specified timeout, a timeout event will occur upon the EM1/EM2/EM3 wake event.

When the EN bit in WDOGn_EN is cleared to 0, the watchdog counter is reset. Any pending interrupt flags will remain active until cleared.

25.3.1 Clock Source

Four clock sources are available for use with the watchdog, through the CLKSEL field in CMU_WDOGn_CFG. The selected oscillator source automatically starts when the watchdog is enabled. To prevent accidental change of the clock selection, CMU_WDOGLOCK can be written anything other than UNLOCK code. Also, respective oscillator has locks to prevent accidental disabling of oscillators. The PERSEL field in WDOGn_CFG is used to divide the selected watchdog clock, and the timeout for the watchdog timer can be calculated with the formula:

$$T_{TIMEOUT} = [2^{(PERSEL+3)} + 1] / f$$

where f is the frequency of the selected clock.

Users must clear EM2RUN and EM3RUN when the selected clock source is HCLKDIV1024.

25.3.2 Debug Functionality

The watchdog timer can either keep running or be frozen when the device is halted by a debugger. This configuration is done through the DEBUGRUN bit in WDOGn_CFG. When code execution is resumed, the watchdog will continue counting where it left off.

25.3.3 Energy Mode Handling

The watchdog timer can be configured to either keep on running or freeze when entering EM1 Sleep, EM2 DeepSleep, or EM3 Stop. The configuration is done individually for each energy mode in the EM1RUN, EM2RUN, and EM3RUN bits in WDOGn_CFG. When the watchdog has been frozen and is re-entering an energy mode where it is running, the watchdog timer will continue counting where it left off. The watchdog does not run in EM4. If EM4BLOCK in WDOGn_CFG is set, the CPU will be prevented from entering EM4 by software request.

25.3.4 Warning Interrupt

The watchdog implements a warning interrupt which can be configured to occur at approximately 25%, 50%, or 75% of the timeout period through the WARNSEL field of the WDOGn_CFG register. This interrupt can be used to wake up the cpu for clearing the watchdog. The warning point for the watchdog timer can be calculated with the formula:

$$T_{WARNING} = [2^{(PERSEL+3)} +1]*WARNSEL / 4 / f$$

where f is the frequency of the selected clock.

When the watchdog is enabled, it is recommended to clear the watchdog before changing WARNSEL.

25.3.5 Window Interrupt

This interrupt occurs when the watchdog is cleared below a certain threshold. This threshold is given by the formula:

$$T_{WINDOW} = [2^{(PERSEL+3)} + 1]*WARNSEL / 8 / f$$

where f is the frequency of the selected clock.

This value will be approximately 12.5%, 25%, 37.5%, 50%, 62.5%, 75%, or 87.5% of the timeout value based on the WINSEL field of the WDOGn_CFG. Figure 25.1 WDOG Warning, Window, and Timeout on page 1029 illustrates the warning, the window, and the timeout interrupts. Also, it shows where the PRS rising edge needs to happen. The PRS edge detection feature is discussed later.

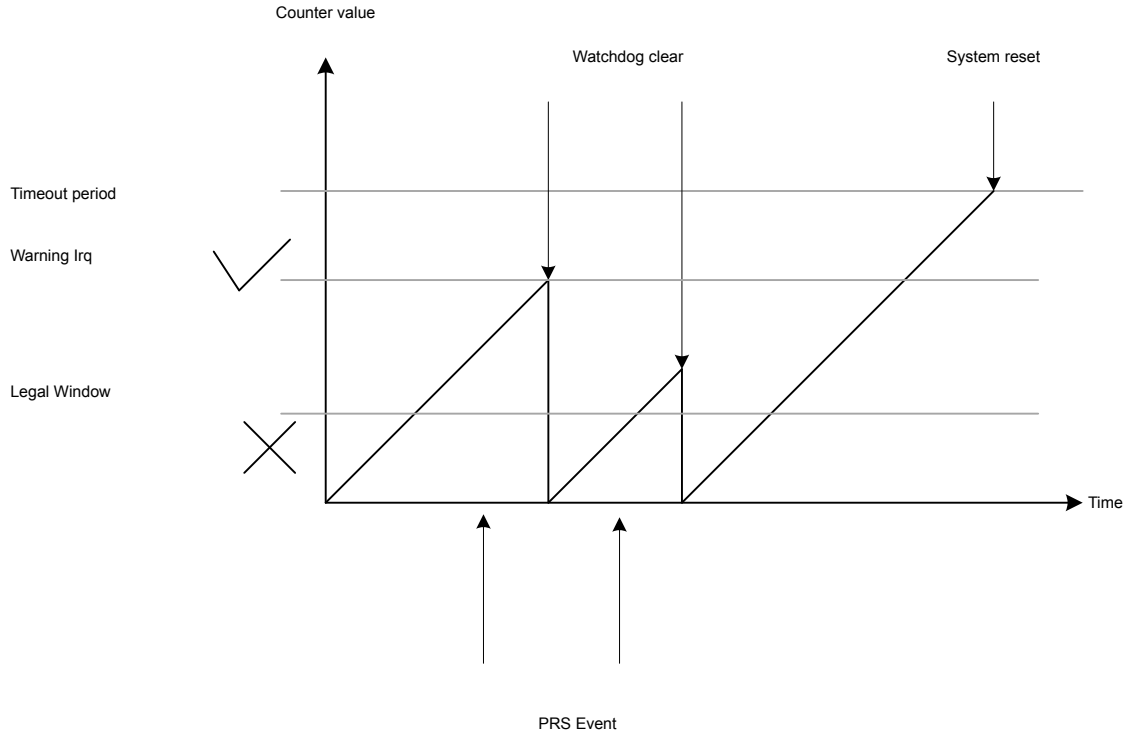


Figure 25.1. WDOG Warning, Window, and Timeout

25.3.6 PRS as Watchdog Clear

A PRS channel (selected by register PRS_CONSUMER_WDOGn_SRC0) can be used to clear the watchdog counter. To enable this feature, CLRSRC must be set to 1. Figure 25.2 PRS Clearing WDOG on page 1030 shows how the PRS channel takes over the WDOG clear function. Clearing the WDOG with the PRS is mutually exclusive of clearing the WDT by software.

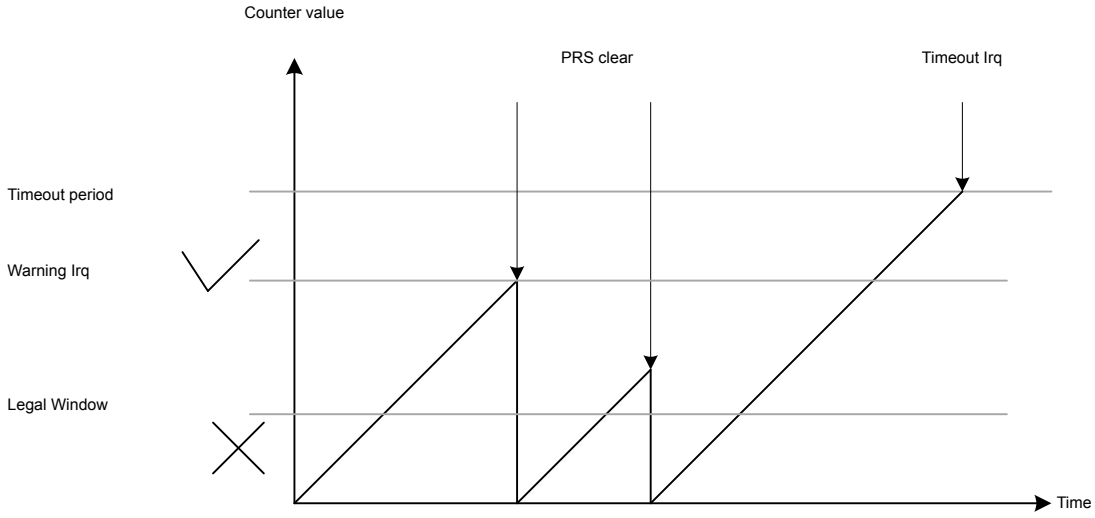


Figure 25.2. PRS Clearing WDOG

25.3.7 PRS Rising Edge Monitoring

PRS channels can be used to monitor multiple processes. The first and second channel are selected by PRS_CONSUMER_WDOGn_SRC0 and PRS_CONSUMER_WDOGn_SRC1, respectively. If enabled, every time the watch dog timer is cleared the PRS channels are checked and any channel which has not seen an event can trigger an interrupt.

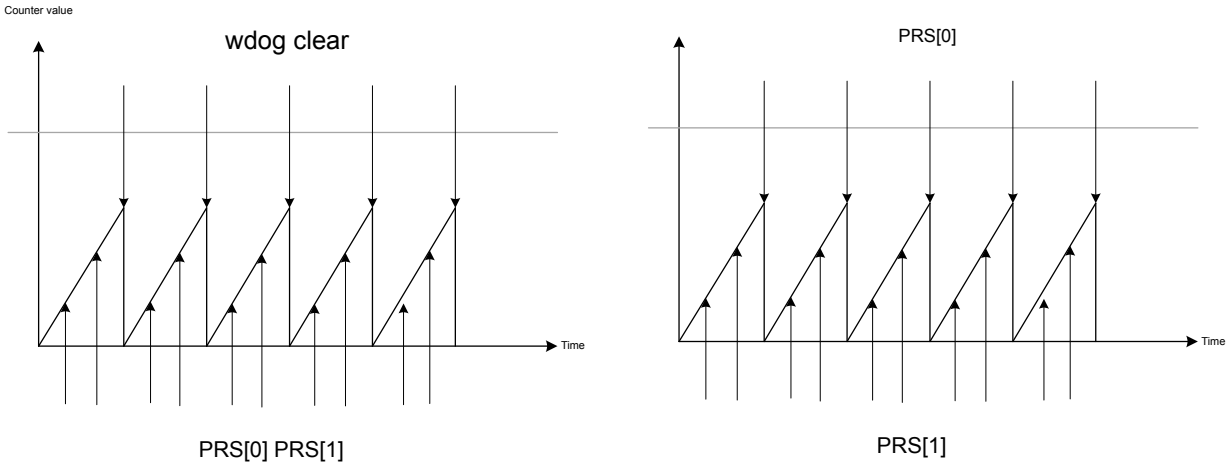


Figure 25.3. PRS Edge Monitoring in WDOG

25.4 WDOG Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	WDOG_IPVERSION	R	IP Version Register
0x004	WDOG_EN	RW ENABLE	Enable Register
0x008	WDOG_CFG	RW CONFIG	Configuration Register
0x00C	WDOG_CMD	W LFSYNC	Command Register
0x014	WDOG_STATUS	RH	Status Register
0x018	WDOG_IF	RWH INTFLAG	Interrupt Flag Register
0x01C	WDOG_IEN	RW	Interrupt Enable Register
0x020	WDOG_LOCK	W	Lock Register
0x024	WDOG_SYNCBUSY	RH	Synchronization Busy Register
0x1000	WDOG_IPVERSION_SET	R	IP Version Register
0x1004	WDOG_EN_SET	RW ENABLE	Enable Register
0x1008	WDOG_CFG_SET	RW CONFIG	Configuration Register
0x100C	WDOG_CMD_SET	W LFSYNC	Command Register
0x1014	WDOG_STATUS_SET	RH	Status Register
0x1018	WDOG_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x101C	WDOG_IEN_SET	RW	Interrupt Enable Register
0x1020	WDOG_LOCK_SET	W	Lock Register
0x1024	WDOG_SYNCBUSY_SET	RH	Synchronization Busy Register
0x2000	WDOG_IPVERSION_CLR	R	IP Version Register
0x2004	WDOG_EN_CLR	RW ENABLE	Enable Register
0x2008	WDOG_CFG_CLR	RW CONFIG	Configuration Register
0x200C	WDOG_CMD_CLR	W LFSYNC	Command Register
0x2014	WDOG_STATUS_CLR	RH	Status Register
0x2018	WDOG_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x201C	WDOG_IEN_CLR	RW	Interrupt Enable Register
0x2020	WDOG_LOCK_CLR	W	Lock Register
0x2024	WDOG_SYNCBUSY_CLR	RH	Synchronization Busy Register
0x3000	WDOG_IPVERSION_TGL	R	IP Version Register
0x3004	WDOG_EN_TGL	RW ENABLE	Enable Register
0x3008	WDOG_CFG_TGL	RW CONFIG	Configuration Register
0x300C	WDOG_CMD_TGL	W LFSYNC	Command Register
0x3014	WDOG_STATUS_TGL	RH	Status Register
0x3018	WDOG_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x301C	WDOG_IEN_TGL	RW	Interrupt Enable Register
0x3020	WDOG_LOCK_TGL	W	Lock Register

Offset	Name	Type	Description
0x3024	WDOG_SYNCBUSY_TGL	RH	Synchronization Busy Register

25.5 WDOG Register Description

25.5.1 WDOG_IPVERSION - IP Version Register

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP Version**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

25.5.2 WDOG_EN - Enable Register

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disabling busy status**
	When EN is cleared, DISABLING status is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs and FIFO			
0	EN	0x0	RW	**Module Enable**
	The ENABLE bit enables the module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit.			

25.5.3 WDOG_CFG - Configuration Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0x0					0x0							0xF								0x0	0x0	0x0			0x0	0x0	0x0	0x0	0x0	0x0
Access		RW					RW							RW								RW	RW	RW			RW	RW	RW	RW	RW	RW
Name		WINSEL					WARNSEL							PERSEL								PRS1MISSRSTEN	PRS0MISSRSTEN	WDOGRSTDIS			DEBUGRUN	EM4BLOCK	EM3RUN	EM2RUN	EM1RUN	CLRSRC

Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
30:28	WINSEL	0x0	RW	**WDOG Illegal Window Select**

Select WDOG illegal limit.

Value	Mode	Description
0	DIS	Disabled.
1	SEL1	Window timeout is 12.5% of the Timeout.
2	SEL2	Window timeout is 25% of the Timeout.
3	SEL3	Window timeout is 37.5% of the Timeout.
4	SEL4	Window timeout is 50% of the Timeout.
5	SEL5	Window timeout is 62.5% of the Timeout.
6	SEL6	Window timeout is 75.5% of the Timeout.
7	SEL7	Window timeout is 87.5% of the Timeout.

27:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25:24	WARNSEL	0x0	RW	**WDOG Warning Period Select**

Select WDOG warning timeout period.

Value	Mode	Description
0	DIS	Disable
1	SEL1	Warning timeout is 25% of the Timeout.
2	SEL2	Warning timeout is 50% of the Timeout.
3	SEL3	Warning timeout is 75% of the Timeout.

23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PERSEL	0xF	RW	**WDOG Timeout Period Select**

Select WDOG timeout period.

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	SEL0		Timeout period of 9 wdog cycles
	1	SEL1		Timeout period of 17 wdog cycles
	2	SEL2		Timeout period of 33 wdog cycles
	3	SEL3		Timeout period of 65 wdog cycles
	4	SEL4		Timeout period of 129 wdog cycles
	5	SEL5		Timeout period of 257 wdog cycles
	6	SEL6		Timeout period of 513 wdog cycles
	7	SEL7		Timeout period of 1k wdog cycles
	8	SEL8		Timeout period of 2k wdog cycles
	9	SEL9		Timeout period of 4k wdog cycles
	10	SEL10		Timeout period of 8k wdog cycles
	11	SEL11		Timeout period of 16k wdog cycles
	12	SEL12		Timeout period of 32k wdog cycles
	13	SEL13		Timeout period of 64k wdog cycles
	14	SEL14		Timeout period of 128k wdog cycles
	15	SEL15		Timeout period of 256k wdog cycles
15:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	PRS1MISSRSTEN	0x0	RW	**PRS Src1 Missing Event WDOG Reset**
	When set, a PRS Source 1 missing event will trigger a WDOG reset.			
9	PRS0MISSRSTEN	0x0	RW	**PRS Src0 Missing Event WDOG Reset**
	When set, a PRS Source 0 missing event will trigger a WDOG reset.			
8	WDOGRSTDIS	0x0	RW	**WDOG Reset Disable**
	Disable WDOG reset output.			
	Value	Mode		Description
	0	EN		A timeout will cause a WDOG reset
	1	DIS		A timeout will not cause a WDOG reset
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	DEBUGRUN	0x0	RW	**Debug Mode Run**
	Set to keep WDOG running in debug mode.			
	Value	Mode		Description
	0	DISABLE		WDOG timer is frozen in debug mode
	1	ENABLE		WDOG timer is running in debug mode
4	EM4BLOCK	0x0	RW	**EM4 Block**

Bit	Name	Reset	Access	Description
	Set to disallow EM4 entry by software.			
	Value	Mode		Description
	0	DISABLE		EM4 can be entered by software. See EMU for detailed description.
	1	ENABLE		EM4 cannot be entered by software.
3	EM3RUN	0x0	RW	**EM3 Run**
	Set to keep WDOG running in EM3.			
	Value	Mode		Description
	0	DISABLE		WDOG timer is frozen in EM3.
	1	ENABLE		WDOG timer is running in EM3.
2	EM2RUN	0x0	RW	**EM2 Run**
	Set to keep WDOG running in EM2.			
	Value	Mode		Description
	0	DISABLE		WDOG timer is frozen in EM2.
	1	ENABLE		WDOG timer is running in EM2.
1	EM1RUN	0x0	RW	**EM1 Run**
	Set to keep WDOG running in EM1.			
	Value	Mode		Description
	0	DISABLE		WDOG timer is frozen in EM1.
	1	ENABLE		WDOG timer is running in EM1.
0	CLRSRC	0x0	RW	**WDOG Clear Source**
	Select WDOG clear source.			
	Value	Mode		Description
	0	SW		A write to the clear bit will clear the WDOG counter
	1	PRSSRC0		A rising edge on the PRS Source 0 will clear the WDOG counter

25.5.4 WDOG_CMD - Command Register

Offset	Bit Position																															
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																W(nB)
Name																																CLEAR

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	CLEAR	0x0	W(nB)	**WDOG Timer Clear**

Clear WDOG timer. The bit must be written 4 WDOG cycles before the timeout.

Value	Mode	Description
0	UNCHANGED	WDOG timer is unchanged.
1	CLEARED	WDOG timer is cleared to 0.

25.5.5 WDOG_STATUS - Status Register

Offset	Bit Position																															
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	LOCK																															

Bit	Name	Reset	Access	Description
31	LOCK	0x0	R	**WDOG Configuration Lock Status**

Status of all lockable WDOG registers.

Value	Mode	Description
0	UNLOCKED	All WDOG lockable registers are unlocked.
1	LOCKED	All WDOG lockable registers are locked.

30:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*

25.5.6 WDOG_IF - Interrupt Flag Register

Offset	Bit Position																														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												PEM1	PEM0	WIN	WARN	TOUT

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	PEM1	0x0	RW	**PRS Src1 Event Missing Interrupt Flag**
	Set when a WDOG clear happens before a prs event has been detected on PRS Source one.			
3	PEM0	0x0	RW	**PRS Src0 Event Missing Interrupt Flag**
	Set when a WDOG clear happens before a prs event has been detected on PRS Source zero.			
2	WIN	0x0	RW	**WDOG Window Interrupt Flag**
	Set when a WDOG clear happens below the window limit value.			
1	WARN	0x0	RW	**WDOG Warning Timeout Interrupt Flag**
	Set when a WDOG warning timeout has occurred.			
0	TOUT	0x0	RW	**WDOG Timeout Interrupt Flag**
	Set when a WDOG timeout has occurred.			

25.5.7 WDOG_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												PEM1	PEM0	WIN	WARN	TOUT

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	PEM1	0x0	RW	**PRS Src1 Event Missing Interrupt Enable**
	Enable/disable the PEM1 interrupt.			
3	PEM0	0x0	RW	**PRS Src0 Event Missing Interrupt Enable**
	Enable/disable the PEM0 interrupt.			
2	WIN	0x0	RW	**WDOG Window Interrupt Enable**
	Enable/disable the WIN interrupt.			
1	WARN	0x0	RW	**WDOG Warning Timeout Interrupt Enable**
	Enable/disable the WARN interrupt.			
0	TOUT	0x0	RW	**WDOG Timeout Interrupt Enable**
	Enable/disable the TOUT interrupt.			

25.5.8 WDOG_LOCK - Lock Register

Offset	Bit Position																															
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0xABE8							
Access																									W							
Name																									LOCKKEY							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0xABE8	W	**WDOG Configuration Lock**

Write any other value than the unlock code to lock WDOG_EN, WDOD_CFG registers from editing. Write the unlock code to unlock.

Value	Mode	Description
0	LOCK	Lock WDOG lockable registers
44008	UNLOCK	Unlock WDOG lockable registers

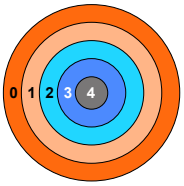
25.5.9 WDOG_SYNCBUSY - Synchronization Busy Register

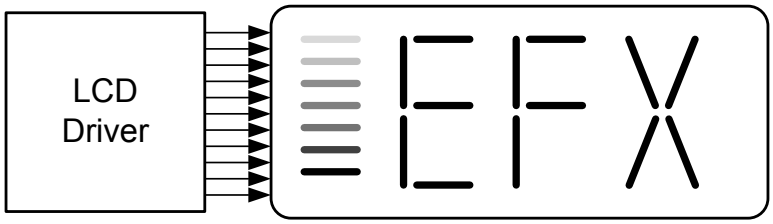
Offset	Bit Position																															
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																R
Name																																CMD

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	CMD	0x0	R	**Sync Busy for Cmd Register**

CMD bitfield sync is busy when set.

26. LCD - Liquid Crystal Display Driver





Quick Facts

What?

The LCD driver can drive LCD displays of up to 4x28 or 8x24 segments. The animation feature makes it possible to have active animations without CPU intervention.

Why?

Segmented LCD displays are a common way to display information. The extreme low-power LCD driver enables a lot of applications to utilize an LCD display even in energy critical systems.

How?

The low frequency clock signal, low-power waveform, animation and blink capabilities enable the LCD driver to run autonomously in EM2 for long periods. Adding the flexible frame rate setting, contrast control, and different multiplexing modes make the EFM32PG28 the optimal choice for battery-driven systems with LCD panels.

26.1 Introduction

The LCD driver is capable of driving a segmented LCD display in combinations of up to 8 x 24 segments. A charge pump function enables it to provide the LCD display with higher voltage than the supply voltage for the device. In addition, an animation feature can run custom animations on the LCD display without any CPU intervention. The LCD driver can also remain active even in Energy Mode 2 and provides a Frame Counter interrupt or a Display Counter interrupt that can wake-up the device on a regular basis for updating data.

26.2 Features

- Up to 4x28 or 8x24 segments.
- Configurable multiplexing (1, 2, 3, 4, 6, 8)
- Supports COM/SEG combinations of: 1x28, 2x28, 3x28, 4x28, 6x26, and 8x24
- Configurable bias/voltage levels settings
- Configurable clock source prescaler
- Configurable Frame rate
- Segment lines can be enabled or disabled individually
- Blink capabilities
- Integrated animation functionality
  - Available on SEG0-SEG7 or SEG8-SEG15
- Charge Redistribution
- Charge pump capabilities
- Possible to run on external power
- Programmable contrast
- Frame Counter
- LCD frame interrupt
- Display Counter
- Display Counter interrupt
- Synchronization Done interrupt
- Direct segment control

26.3 Functional Description

An overview of the LCD module is shown in Figure 26.1 LCD Block Diagram on page 1041. In its simplest form, an LCD driver would apply a voltage above a certain threshold voltage in order to darken a segment and a voltage below threshold to make a segment clear. However, the LCD display segment will degrade if the applied voltage has a DC-component. To avoid this, the applied waveforms are arranged such that the differential voltage seen by each segment has an average value of zero, and such that the RMS voltage (or differential sum of the two waveforms for fast response LCDs) is below the segment threshold voltage if the segment shall be transparent, and above the segment threshold voltage when the segment shall be dark. The segment lines to support up to 192 different LCD segments using 8 common lines by 24 segment lines. The common lines and segment lines can be enabled or disabled individually to prevent the LCD driver from occupying more I/O resources than required.

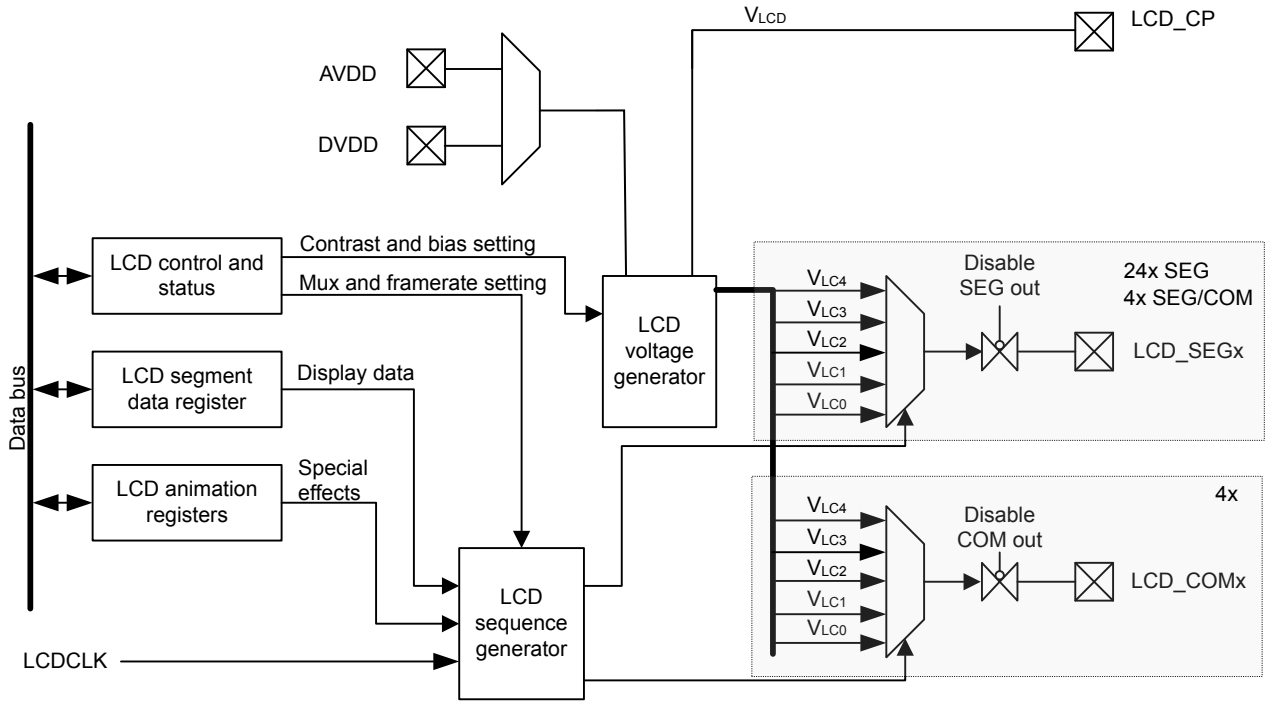


Figure 26.1. LCD Block Diagram

The LCD peripheral clock, LCDCLK, is selected in the CMU. Within this chapter it may be referred to as LCDCLK or by the more generic name CLK_PER (peripheral clock). The register interface is clocked by the selected bus clock, referred to as either PCLK or CLK_BUS.

Setting the EN bit in LCD_EN enables the LCD driver. The MUX bit-field in LCD_DISPCTRL determines which COM lines are driven by the LCD driver. By default, LCD.COM0 is driven whenever the LCD driver is enabled.

Refer to the GPIO module to enable GPIOs for individual segment and common lines.

Each LCD segment pin can also be individually disabled by setting the pin to any other state than DISABLED in the GPIO pin configuration.

26.3.1 LCD Frame Rates

LCD Frame rates are usually set between 30 to 100 frames per second (FPS). The LCD peripheral clock LCDCLK can be prescaled which reduces power by slowing down the internal LCD logic clocking. Charge redistribution can also be used to reduce power by briefly connecting LCD pins to the same mid-voltage driver before switching each LCD pin to its next voltage driver.

The 32.768 kHz clock should remain stable during LCD usage. If the clock is switched between LFXO and LFRCO, the LCD should first be disabled before switching the clock.

Each multiplexed line will have two phase periods per frame. For example, a static mux configuration (one COM line) will have two phase periods per frame, while a Quadruplex mux configuration (four COM lines) will have 8 phase periods per frame. CHGRDST in LCD_DISPCTRL is used to select the number of prescaled LCDCLK cycles used for charge redistribution. FRDIV in LCD_FRAMERATE is used to provide an an additional increase of the phase period. The actual frame rate divider is FRDIV + 1. The charge redistribution cycle percentage (CHGRDST PERCENT) should generally be 5% or less to keep the reduction in LCD pad RMS voltage to a minimum. If charge redistribution is not used, a larger prescaling value is recommended to reduce power. Charge redistribution is on by default, but it can be disabled by setting CHGRDST to disable. in LCD_DISPCTRL. Table 26.1 LCD Frame Rate on page 1042 shows several example settings and the resulting percentage of time that charge redistribution is on.

Table 26.1. LCD Frame Rate

MUX	CHGRDST	Prescale	Actual FR Divider	FPS	CHGRDST PERCENT
Static	4	1	512	32.0	0.8%
Static	4	1	163	100.5	2.5%
Static	1	16	32	32.0	3.1%
Static	1	8	20	102.4	5.0%
Static	0	128	4	32.0	0.0%
Static	0	32	5	102.4	0.0
Quadruplex	4	1	136	30.1	2.9%
Quadruplex	2	1	40	102.4	5.0%
Quadruplex	1	4	34	30.1	2.9%
Quadruplex	1	2	20	102.4	5.0%
Quadruplex	0	16	8	32.0	0.0%
Quadruplex	0	4	10	102.4	0.0%
Octaplex	2	1	68	30.1	2.9%
Octaplex	1	1	20	102.4	5.0%
Octaplex	1	2	34	30.1	2.9%
Octaplex	0	64	1	32.0	0.0%

26.3.2 LCD Multiplexing, Bias, and Wave Settings

The LCD driver supports different multiplexing and bias settings, and these can be set individually in the LCD_DISPCTRL.MUX and LCD_DISPCTRL.BIAS respectively, see Table 26.2 LCD Mux Settings on page 1043 and Table 26.3 LCD BIAS Settings on page 1043.

The MUX setting determines the number of LCD COM lines that are enabled, beginning with COM0. When static multiplexing is selected, LCD output is enabled on LCD.COM0, when duplex multiplexing is used, LCD.COM0-LCD.COM1 are used, when triplex multiplexing is selected, LCD.COM0-LCD.COM2 are used, and when quadruplex multiplexing is selected, LCD.COM0-LCD.COM3 are used. For sextaplex and octaplex multipliexing, certain SEG lines cannot be used as the pins are shared with the COM4, 5, 6, and 7 signals. When sextaplex multiplexing is selected, LCD.COM0-LCD.COM5 act as common pins, reducing the number of available segment pins to 26 (LCD.SEG24 and LCD.SEG25 are not available) . Finally when octaplex multiplexing is selected, LCD.COM0-LCD.COM7 act as common pins, reducing the number of available segment pins to 24, with LCD.SEG26 and LCD.SEG27 unavailable.

The waveforms generated by the LCD controller can be generated in two different versions, type A and type B. The type B waveforms have a lower switching frequency than the type A waveforms, and thus consume less power. LCD_DISPCTRL.WAVE decides which waveforms to generate. An example of a type A waveform is shown in Figure 26.2 LCD Type A Waveform for LCD.COM0 in Quadruplex Multiplex Mode, 1/3 Bias on page 1044, and an example of a type B waveform is shown in Figure 26.3 LCD Type B Waveform for LCD.COM0 in Quadruplex Multiplex Mode, 1/3 Bias on page 1044. For COM waveforms, a green dotted lines indicates where the SEG waveform would be for an 'on' level while the red dotted lines indicate where the SEG waveform would be for an 'off' level.

Table 26.2. LCD Mux Settings

MUX	Mode	Multiplexing
000	Static	Static (segments can be multiplexed with LCD_COM[0])
001	Duplex	Duplex (segments can be multiplexed with LCD_COM[1:0])
010	Triplex	Triplex (segments can be multiplexed with LCD_COM[2:0])
011	Quadruplex	Quadruplex (segments can be multiplexed with LCD_COM[3:0])
101	Sextaplex	Sextaplex (segments can be multiplexed with LCD_COM[5:0])
111	Octaplex	Octaplex (segments can be multiplexed with LCD_COM[7:0])

Table 26.3. LCD BIAS Settings

BIAS	Mode	Bias setting
00	Static	Static (2 levels)
01	Half Bias	1/2 Bias (3 levels)
10	Third Bias	1/3 Bias (4 levels)
11	Fourth Bias	1/4 Bias (5 levels)

Table 26.4. LCD Wave Settings

WAVE	Mode	Wave mode
0	Type B	Low power optimized waveform output
1	Type A	Regular waveform output

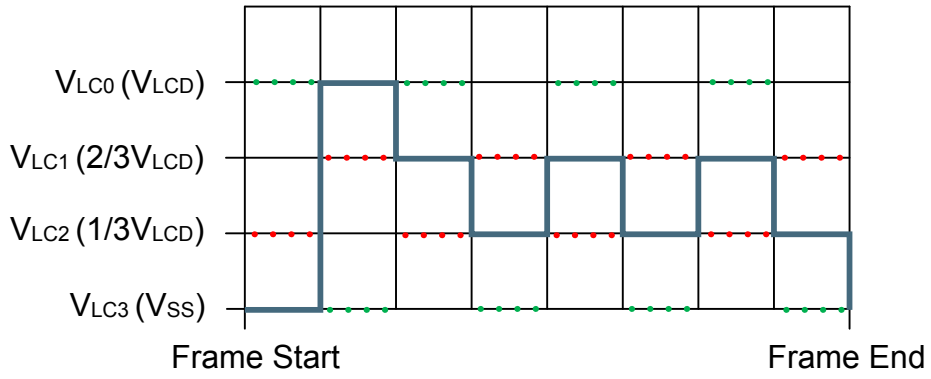


Figure 26.2. LCD Type A Waveform for LCD.COM0 in Quadruplex Multiplex Mode, 1/3 Bias

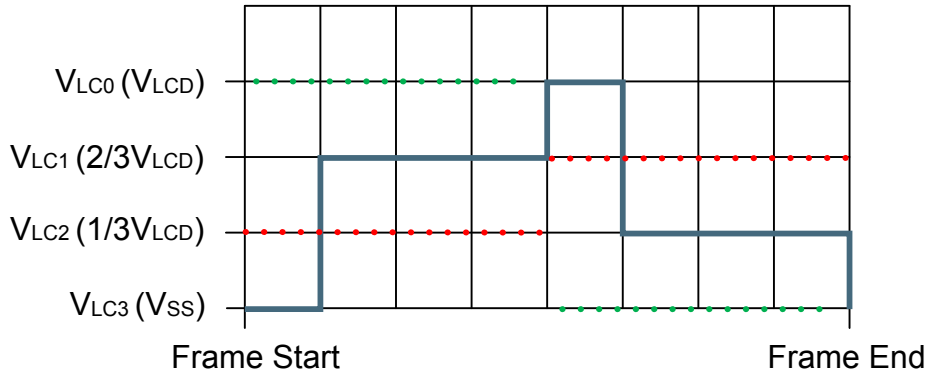


Figure 26.3. LCD Type B Waveform for LCD.COM0 in Quadruplex Multiplex Mode, 1/3 Bias

26.3.3 LCD Waveform Examples

The numbers on the illustration's y-axes in the following sections only indicate different voltage levels. All examples are shown with low-power waveforms.

26.3.3.1 Waveforms with Static Bias and Multiplexing

- With static bias and multiplexing, each segment line can be connected to LCD.COM0. When the segment line has the same waveform as LCD.COM0, the LCD panel pixel is turned off, while when the segment line has the opposite waveform, the LCD panel pixel is turned on.
- DC voltage = 0 (over one frame)
- V_{RMS} (on) = V_{LCD}
- V_{RMS} (off) = 0 (V_{SS})

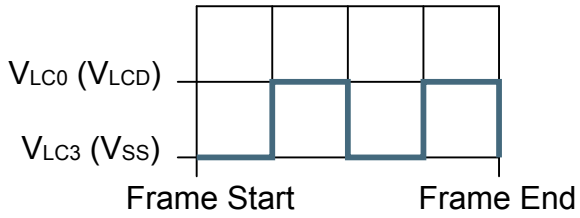


Figure 26.4. LCD Static Bias and Multiplexing - LCD.COM0

26.3.3.2 Waveforms with 1/2 Bias and Duplex Multiplexing

In this mode, each frame is divided into 4 periods. LCD_COM[1:0] lines can be multiplexed with all segment lines. Figure 26.5 LCD 1/2 Bias and Duplex Multiplexing - LCD.COM0 on page 1046 and Figure 26.6 LCD 1/2 Bias and Duplex Multiplexing - LCD.COM1 on page 1046 show 1/2 bias and duplex multiplexing (waveforms show two frames)

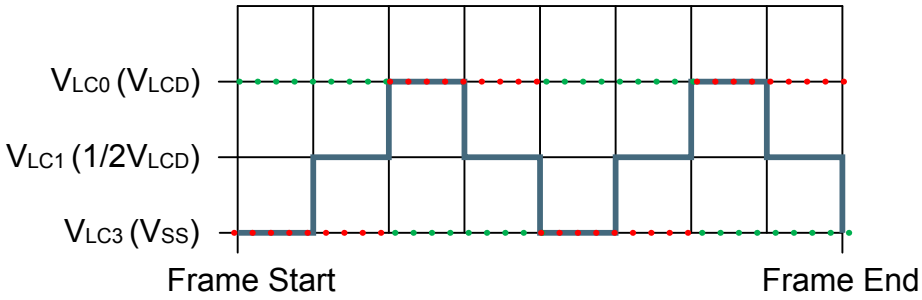


Figure 26.5. LCD 1/2 Bias and Duplex Multiplexing - LCD.COM0

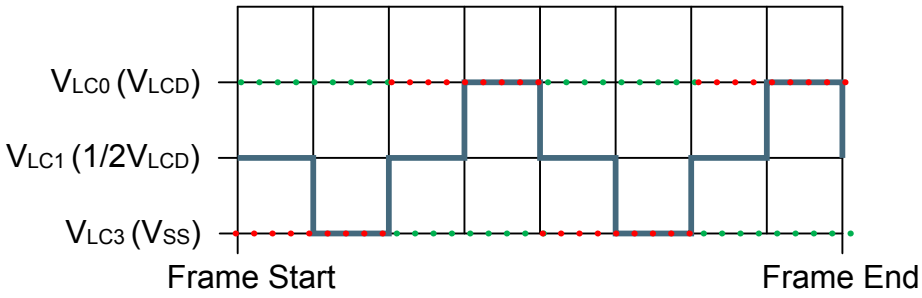


Figure 26.6. LCD 1/2 Bias and Duplex Multiplexing - LCD.COM1

1/2 bias and duplex multiplexing - LCD.SEG0

Figure 26.7 LCD 1/2 Bias and Duplex Multiplexing - LCD.SEG0 on page 1047 is an example to illustrate how different segment waveforms can be multiplexed with the LCD_COM lines in order to turn on and off LCD pixels. As illustrated in Figure 26.9 LCD 1/2 Bias and Duplex Multiplexing - LCD.SEG0-LCD.COM0 on page 1047 and Figure 26.10 LCD 1/2 Bias and Duplex Multiplexing - LCD.SEG0-LCD.COM1 on page 1048, this waveform will turn ON pixels connected to LCD.COM0, while pixels connected to LCD.COM1 will be turned OFF.

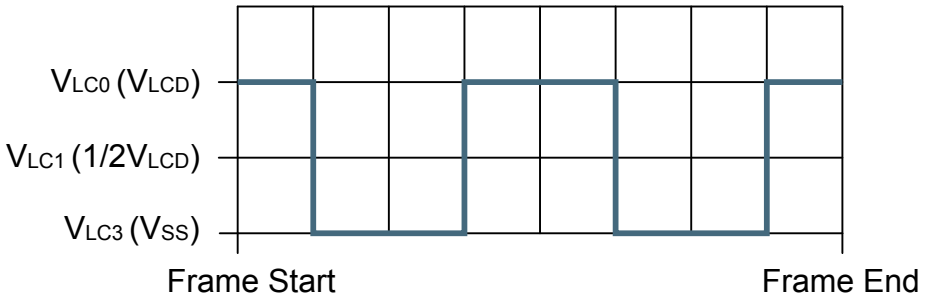


Figure 26.7. LCD 1/2 Bias and Duplex Multiplexing - LCD.SEG0

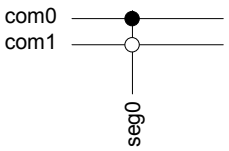


Figure 26.8. LCD 1/2 Bias and Duplex Multiplexing - LCD.SEG0 Connection

1/2 bias and duplex multiplexing - LCD.SEG0-LCD.COM0

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.79 \times V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM0 will be ON with this waveform.

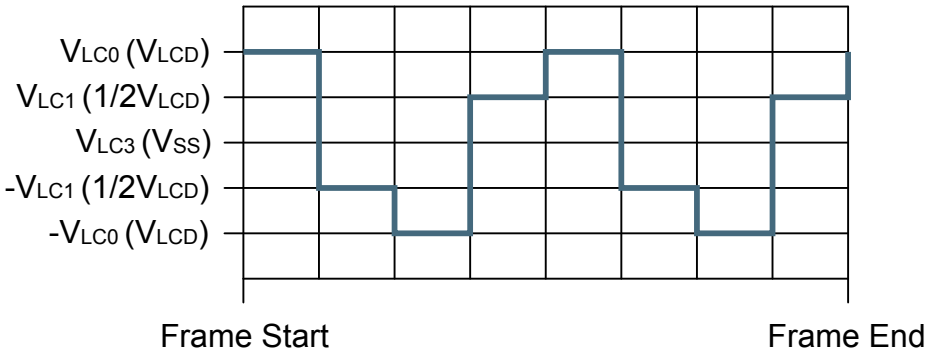


Figure 26.9. LCD 1/2 Bias and Duplex Multiplexing - LCD.SEG0-LCD.COM0

1/2 bias and duplex multiplexing - LCD.SEG0-LCD.COM1

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.35 \times V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM0 will be OFF with this waveform.

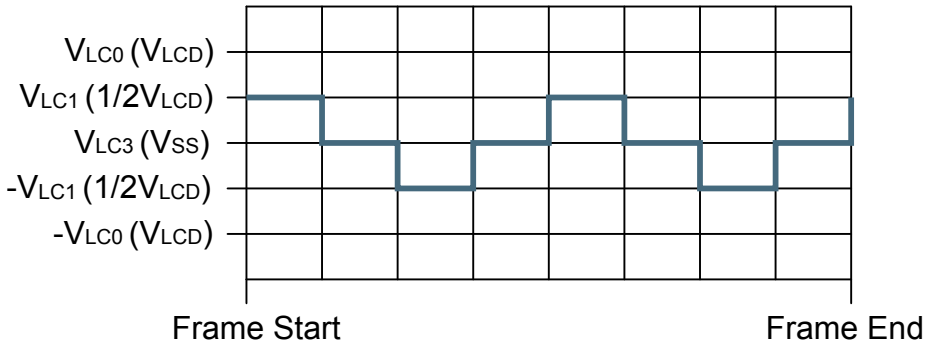


Figure 26.10. LCD 1/2 Bias and Duplex Multiplexing - LCD.SEG0-LCD.COM1

26.3.3.3 Waveforms with 1/3 Bias and Duplex Multiplexing

In this mode, each frame is divided into 4 periods. LCD_COM[1:0] lines can be multiplexed with all segment lines. Figure 26.11 LCD 1/3 Bias and Duplex Multiplexing - LCD.COM0 on page 1049 and Figure 26.12 LCD 1/3 Bias and Duplex Multiplexing - LCD.COM1 on page 1049 show 1/3 bias and duplex multiplexing (waveforms show two frames).

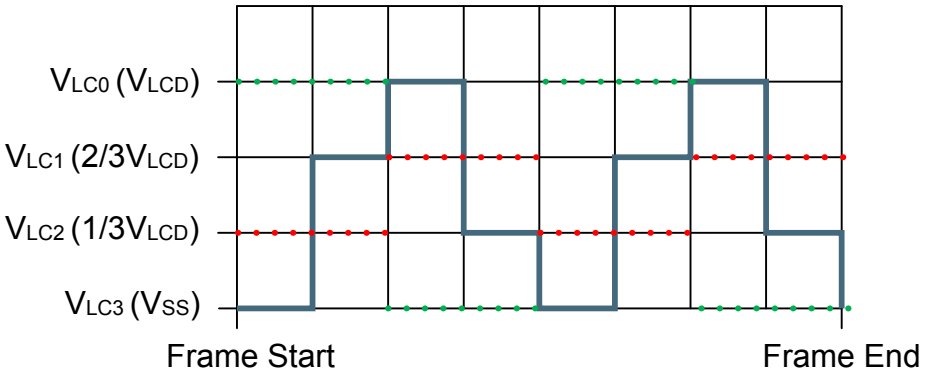


Figure 26.11. LCD 1/3 Bias and Duplex Multiplexing - LCD.COM0

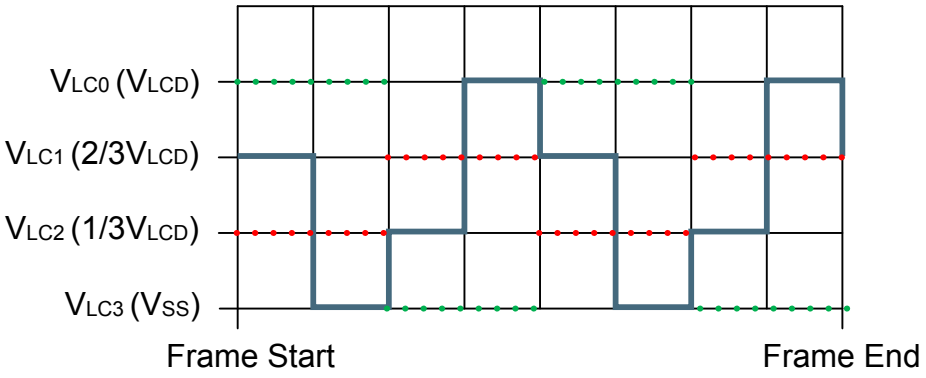


Figure 26.12. LCD 1/3 Bias and Duplex Multiplexing - LCD.COM1

1/3 bias and duplex multiplexing - LCD.SEG0

Figure 26.13 LCD 1/3 Bias and Duplex Multiplexing - LCD.SEG0 on page 1050 is just an example to illustrate how different segment waveforms can be multiplexed with the COM lines in order to turn on and off LCD pixels. As illustrated in Figure 26.15 LCD 1/3 Bias and Duplex Multiplexing - LCD.SEG0-LCD.COM0 on page 1050 and Figure 26.16 LCD 1/3 Bias and Duplex Multiplexing - LCD.SEG0-LCD.COM1 on page 1051, this waveform will turn ON pixels connected to LCD.COM0, while pixels connected to LCD.COM1 will be turned OFF.

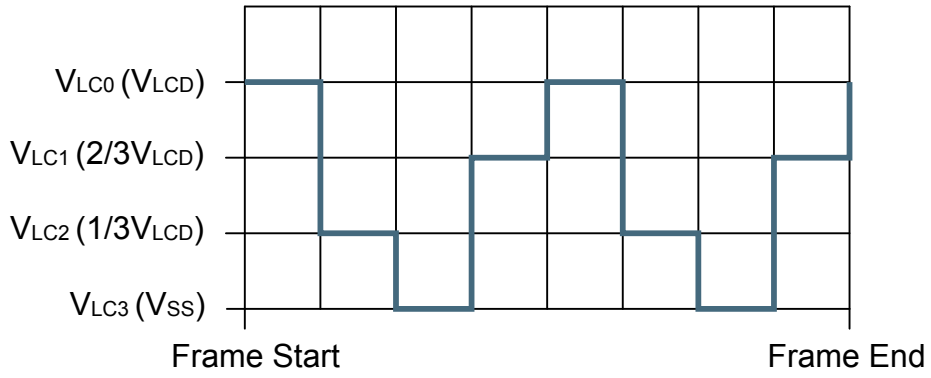


Figure 26.13. LCD 1/3 Bias and Duplex Multiplexing - LCD.SEG0

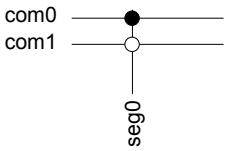


Figure 26.14. LCD 1/3 Bias and Duplex Multiplexing - LCD.SEG0 Connection

1/3 bias and duplex multiplexing - LCD.SEG0-LCD.COM0

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.75 \times V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM0 will be ON with this waveform.

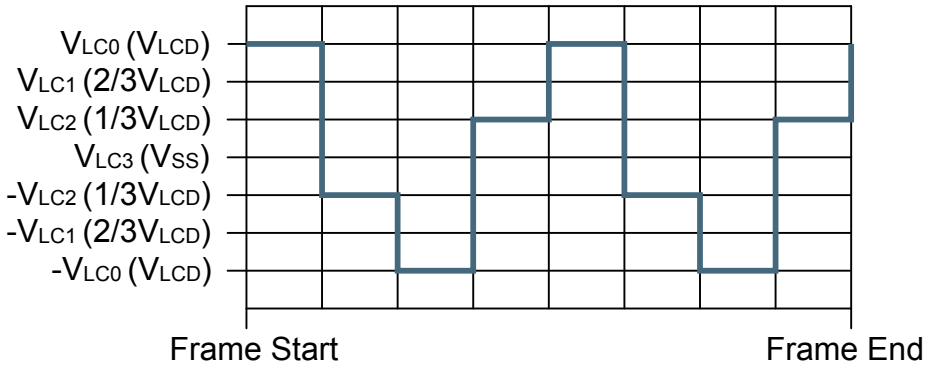


Figure 26.15. LCD 1/3 Bias and Duplex Multiplexing - LCD.SEG0-LCD.COM0

1/3 bias and duplex multiplexing - LCD.SEG0-LCD.COM0

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.33 \times V_{LCD}$

- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM1 will be OFF with this waveform.

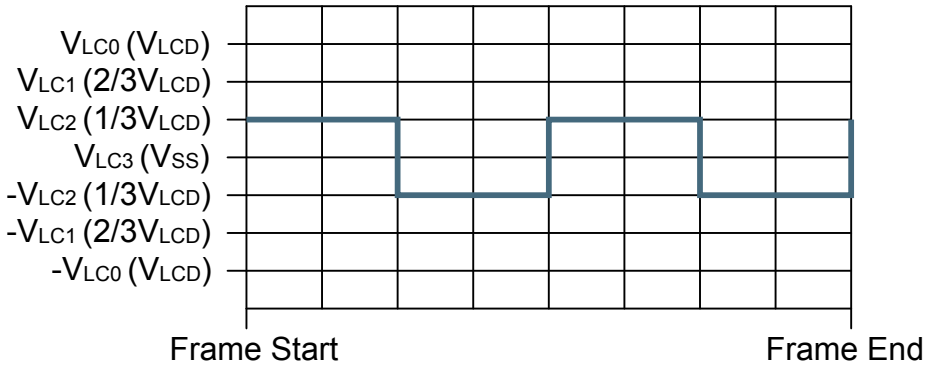


Figure 26.16. LCD 1/3 Bias and Duplex Multiplexing - LCD.SEG0-LCD.COM1

26.3.3.4 Waveforms with 1/2 Bias and Triplex Multiplexing

In this mode, each frame is divided into 6 periods. LCD_COM[2:0] lines can be multiplexed with all segment lines. Figure 26.17 LCD 1/2 Bias and Triplex Multiplexing - LCD.COM0 on page 1052 through Figure 26.19 LCD 1/2 Bias and Triplex Multiplexing - LCD.COM2 on page 1052 show 1/2 bias and triplex multiplexing (waveforms show two frames).

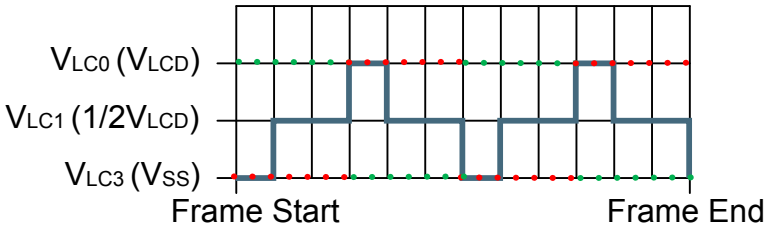


Figure 26.17. LCD 1/2 Bias and Triplex Multiplexing - LCD.COM0

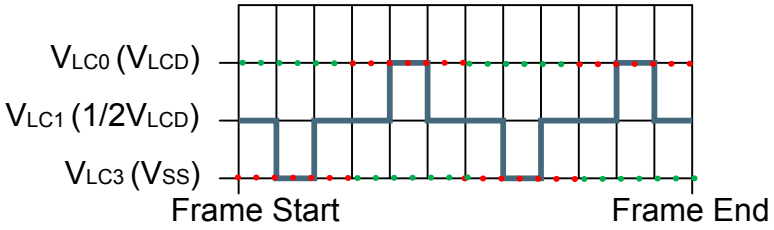


Figure 26.18. LCD 1/2 Bias and Triplex Multiplexing - LCD.COM1

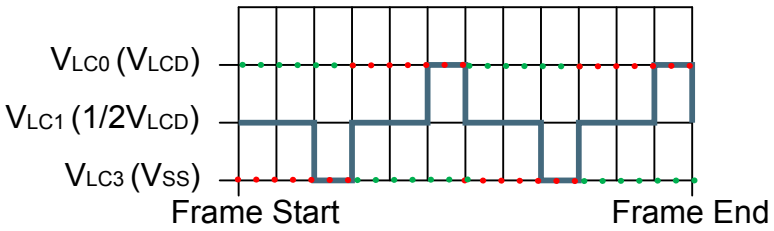


Figure 26.19. LCD 1/2 Bias and Triplex Multiplexing - LCD.COM2

1/2 bias and triplex multiplexing - LCD.SEG0

Figure 26.20 LCD 1/2 Bias and Triplex Multiplexing - LCD.SEG0 on page 1053 is just an example to illustrate how different segment waveforms can be multiplexed with the COM lines in order to turn on and off LCD pixels. As illustrated in Figure 26.22 LCD 1/2 Bias and Triplex Multiplexing - LCD.SEG0-LCD.COM0 on page 1053 through Figure 26.24 LCD 1/2 Bias and Triplex Multiplexing - LCD.SEG0-LCD.COM2 on page 1054, this waveform will turn ON pixels connected to LCD.COM1, while pixels connected to LCD.COM0 and LCD.COM2 will be turned OFF.

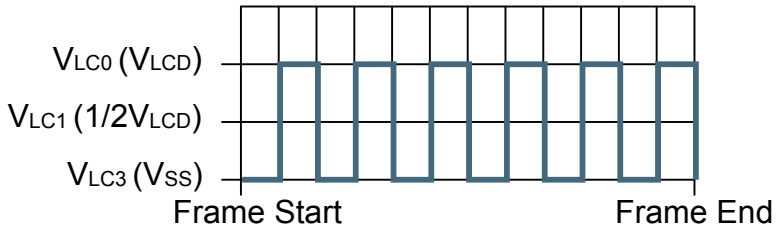


Figure 26.20. LCD 1/2 Bias and Triplex Multiplexing - LCD.SEG0

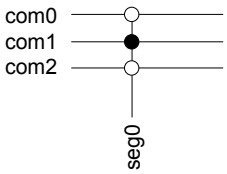


Figure 26.21. LCD 1/2 Bias and Triplex Multiplexing - LCD.SEG0 Connection

1/2 bias and triplex multiplexing - LCD.SEG0-LCD.COM0

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.4 \times V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM0 will be OFF with this waveform.

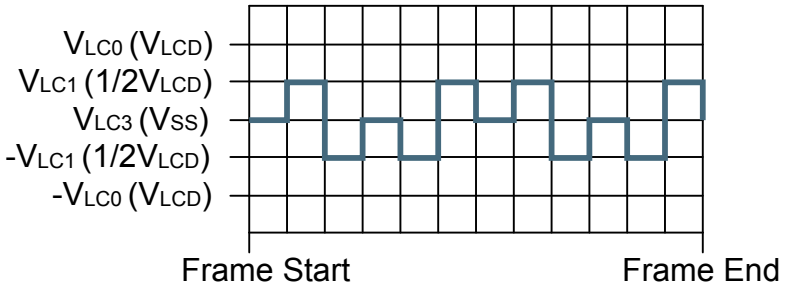


Figure 26.22. LCD 1/2 Bias and Triplex Multiplexing - LCD.SEG0-LCD.COM0

1/2 bias and triplex multiplexing - LCD.SEG0-LCD.COM1

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.7\ V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM1 will be ON with this waveform.

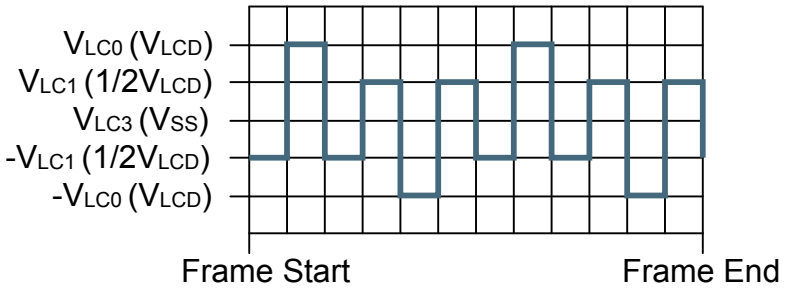


Figure 26.23. LCD 1/2 Bias and Triplex Multiplexing - LCD.SEG0-LCD.COM1

1/2 bias and triplex multiplexing - LCD.SEG0-LCD.COM2

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.4 \times V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM2 will be OFF with this waveform.

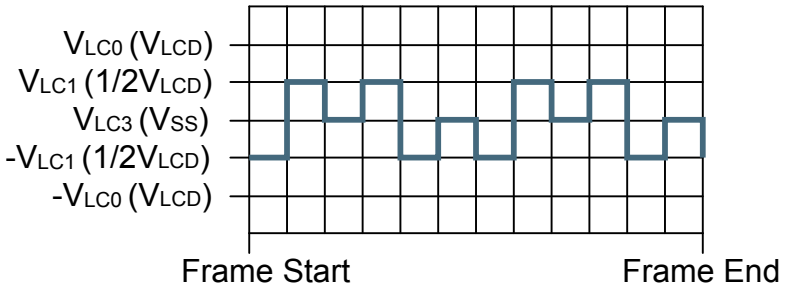


Figure 26.24. LCD 1/2 Bias and Triplex Multiplexing - LCD.SEG0-LCD.COM2

26.3.3.5 Waveforms with 1/3 Bias and Triplex Multiplexing

In this mode, each frame is divided into 6 periods. LCD_COM[2:0] lines can be multiplexed with all segment lines. Figure 26.25 LCD 1/3 Bias and Triplex Multiplexing - LCD.COM0 on page 1055 through Figure 26.27 LCD 1/3 Bias and Triplex Multiplexing - LCD.COM2 on page 1055 show 1/3 bias and triplex multiplexing (waveforms show two frames).

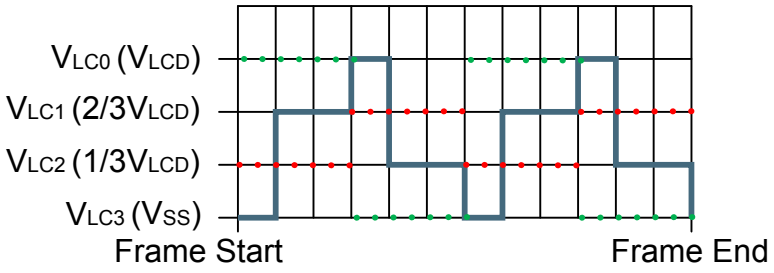


Figure 26.25. LCD 1/3 Bias and Triplex Multiplexing - LCD.COM0

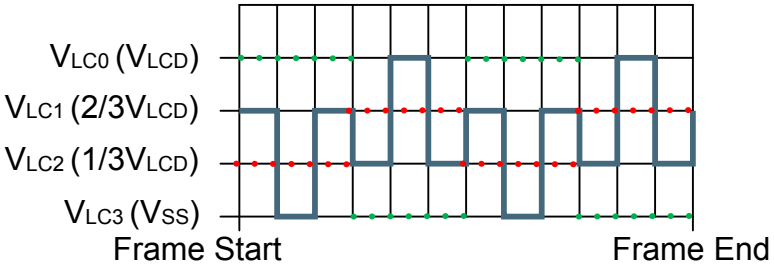


Figure 26.26. LCD 1/3 Bias and Triplex Multiplexing - LCD.COM1

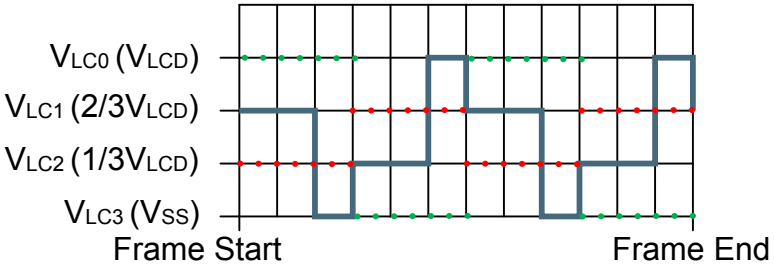


Figure 26.27. LCD 1/3 Bias and Triplex Multiplexing - LCD.COM2

1/3 bias and triplex multiplexing - LCD.SEG0

Figure 26.28 LCD 1/3 Bias and Triplex Multiplexing - LCD.SEG0 on page 1056 illustrates how different segment waveforms can be multiplexed with the COM lines in order to turn on and off LCD pixels. As illustrated in Figure 26.30 LCD 1/3 Bias and Triplex Multiplexing - LCD.SEG0-LCD.COM0 on page 1056 through Figure 26.32 LCD 1/3 Bias and Triplex Multiplexing - LCD.SEG0-LCD.COM2 on

page 1057, this waveform will turn ON pixels connected to LCD.COM1, while pixels connected to LCD.COM0 and LCD.COM2 will be turned OFF.

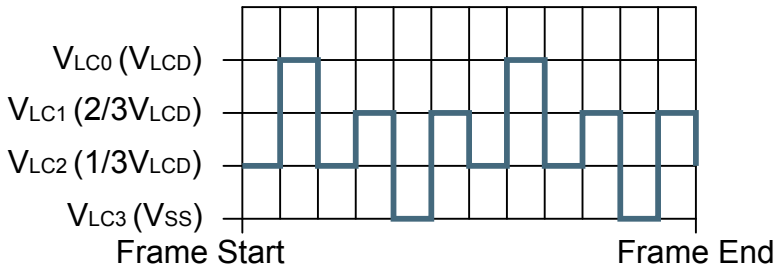


Figure 26.28. LCD 1/3 Bias and Triplex Multiplexing - LCD.SEG0

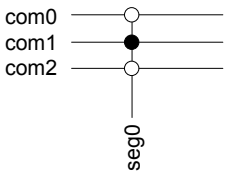


Figure 26.29. LCD 1/3 Bias and Triplex Multiplexing - LCD.SEG0 Connection

1/3 bias and triplex multiplexing - LCD.SEG0-LCD.COM0

- DC voltage = 0 (over one frame)
- V_{RMS} = 0.33 V_{LCD}
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM0 will be OFF with this waveform

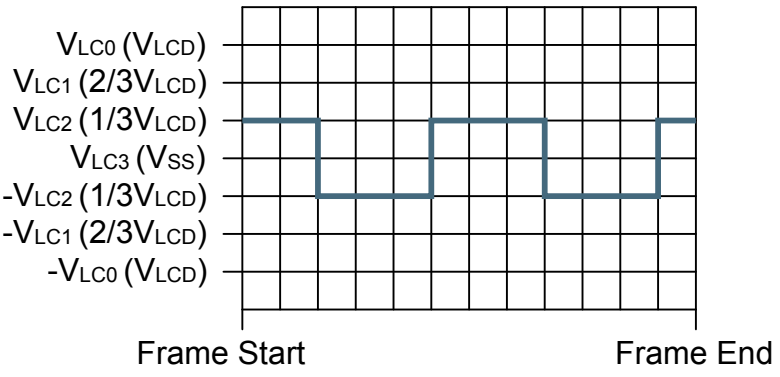


Figure 26.30. LCD 1/3 Bias and Triplex Multiplexing - LCD.SEG0-LCD.COM0

1/3 bias and triplex multiplexing - LCD.SEG0-LCD.COM1

- DC voltage = 0 (over one frame)
- V_{RMS} = 0.64 × V_{LCD}
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM1 will be ON with this waveform

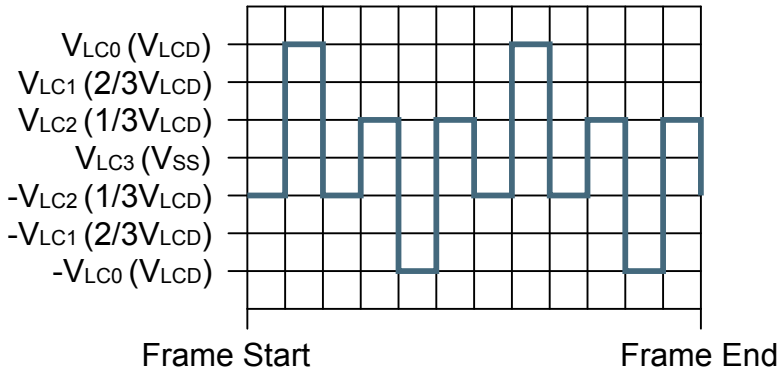


Figure 26.31. LCD 1/3 Bias and Triplex Multiplexing - LCD.SEG0-LCD.COM1

1/3 bias and triplex multiplexing - LCD.SEG0-LCD.COM2

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.33 \times V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM2 will be OFF with this waveform

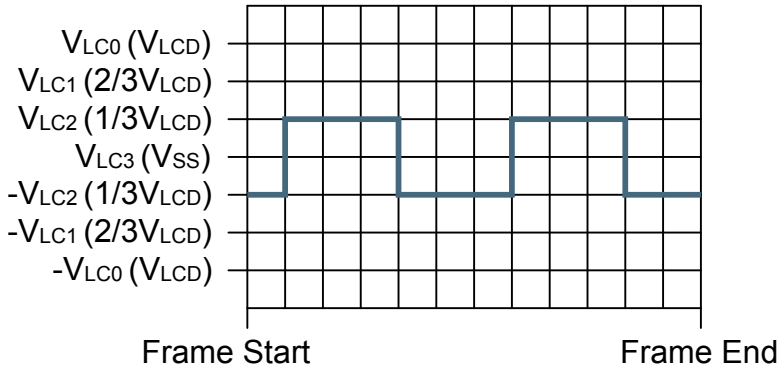


Figure 26.32. LCD 1/3 Bias and Triplex Multiplexing - LCD.SEG0-LCD.COM2

26.3.3.6 Waveforms with 1/3 Bias and Quadruplex Multiplexing

In this mode, each frame is divided into 8 periods. All COM lines can be multiplexed with all segment lines. Figure 26.33 LCD 1/3 Bias and Quadruplex Multiplexing - LCD.COM0 on page 1058 through Figure 26.36 LCD 1/3 Bias and Quadruplex Multiplexing - LCD.COM3 on page 1059 show 1/3 bias and quadruplex multiplexing (waveforms show two frames).

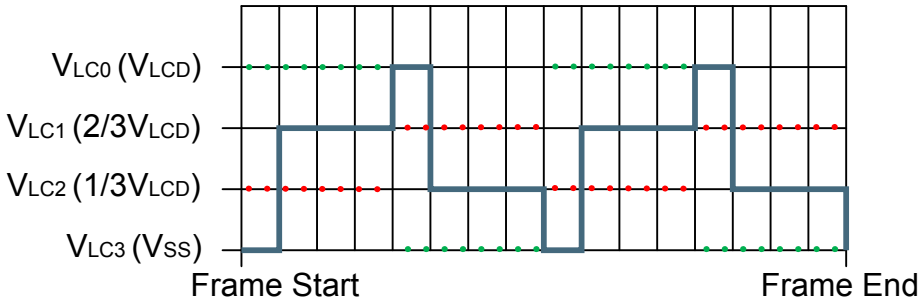


Figure 26.33. LCD 1/3 Bias and Quadruplex Multiplexing - LCD.COM0

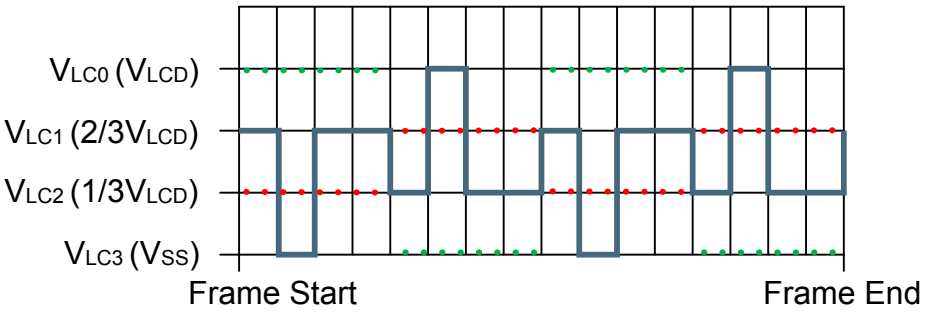


Figure 26.34. LCD 1/3 Bias and Quadruplex Multiplexing - LCD.COM1

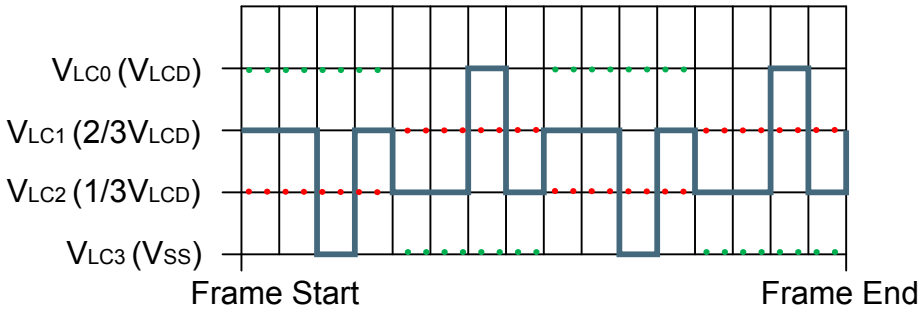


Figure 26.35. LCD 1/3 Bias and Quadruplex Multiplexing - LCD.COM2

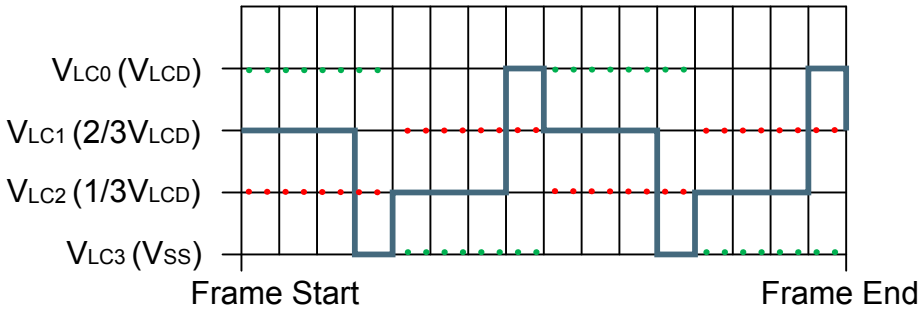


Figure 26.36. LCD 1/3 Bias and Quadruplex Multiplexing - LCD.COM3

1/3 bias and quadruplex multiplexing - LCD.SEG0

Figure 26.37 LCD 1/3 Bias and Quadruplex Multiplexing - LCD.SEG0 on page 1059 is just an example to illustrate how different segment waveforms can be multiplexed with the COM lines in order to turn on and off LCD pixels. As illustrated in Figure 26.39 LCD 1/3 Bias and Quadruplex Multiplexing - LCD.SEG0-LCD.COM0 on page 1060 through Figure 26.42 LCD 1/3 Bias and Quadruplex Multiplexing - LCD.SEG0-LCD.COM3 on page 1061, this wave form will turn ON pixels connected to LCD.COM0 and LCD.COM2, while pixels connected to LCD.COM1 and LCD.COM3 will be turned OFF.

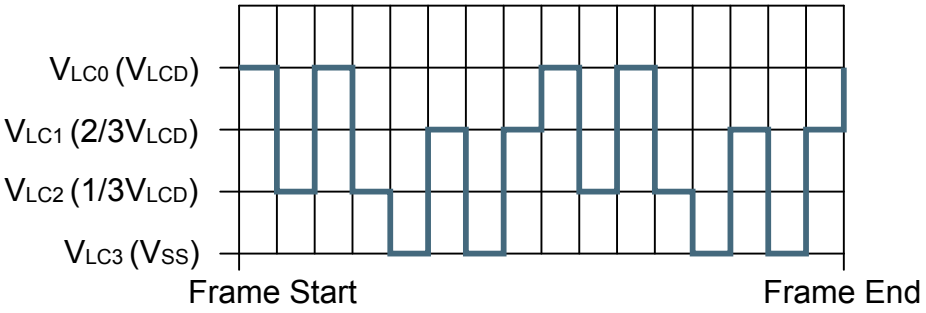


Figure 26.37. LCD 1/3 Bias and Quadruplex Multiplexing - LCD.SEG0

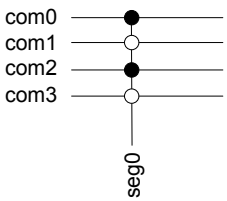


Figure 26.38. LCD 1/3 Bias and Quadruplex Multiplexing - LCD.SEG0 Connection

1/3 bias and quadruplex multiplexing - LCD.SEG0-LCD.COM0

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.58 \times V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM0 will be ON with this waveform.

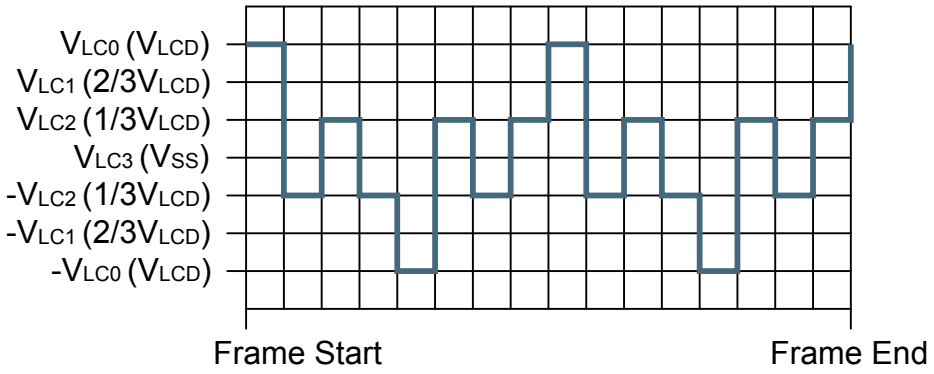


Figure 26.39. LCD 1/3 Bias and Quadruplex Multiplexing - LCD.SEG0-LCD.COM0

1/3 bias and quadruplex multiplexing - LCD.SEG0-LCD.COM1

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.33 \times V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM1 will be OFF with this waveform.

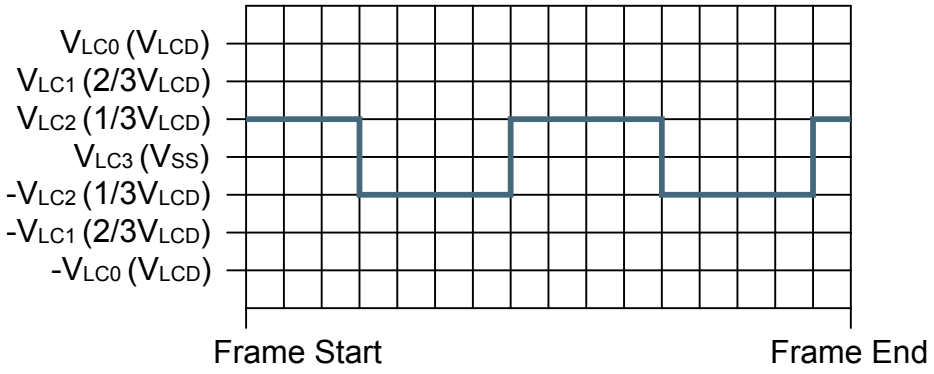


Figure 26.40. LCD 1/3 Bias and Quadruplex Multiplexing - LCD.SEG0-LCD.COM1

1/3 bias and quadruplex multiplexing - LCD.SEG0-LCD.COM2

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.58 \times V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM2 will be ON with this waveform.

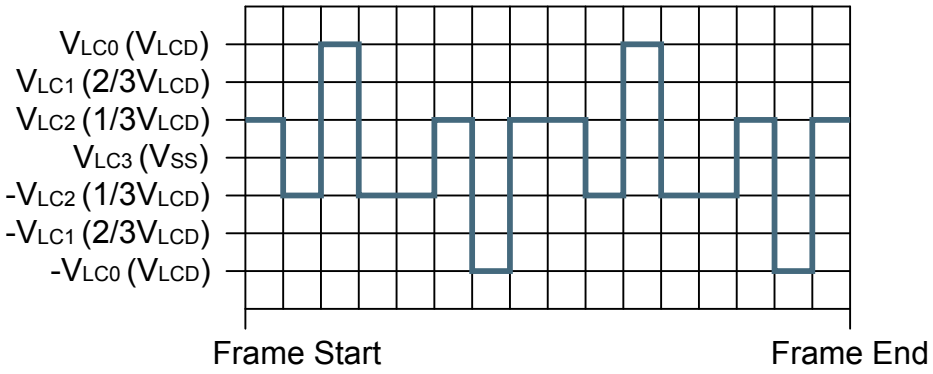


Figure 26.41. LCD 1/3 Bias and Quadruplex Multiplexing - LCD.SEG0-LCD.COM2

1/3 bias and quadruplex multiplexing - LCD.SEG0-LCD.COM2

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.33 \times V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM3 will be OFF with this waveform.

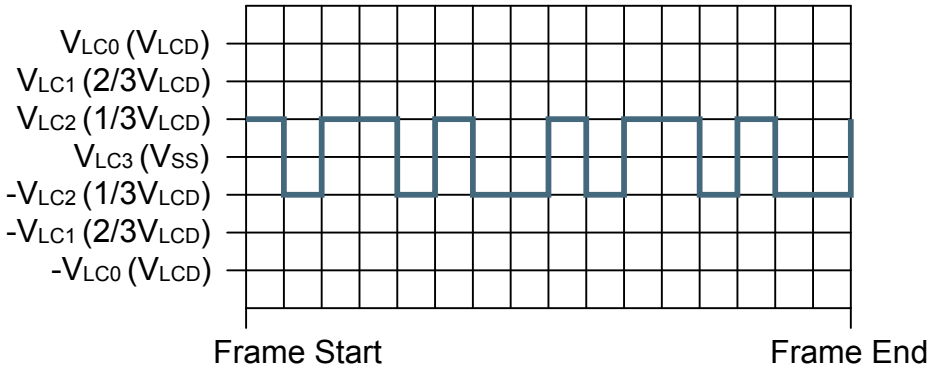


Figure 26.42. LCD 1/3 Bias and Quadruplex Multiplexing - LCD.SEG0-LCD.COM3

26.3.4 Type A Waveforms with Charge Redistribution

Charge redistribution can be used to save power by briefly connecting common lines and segment lines to a common connection between each phase of the waveform. Instead of driving some lines from high to low and other lines from low to high, the lines are briefly sharing charge to get to a mid voltage level before each waveform drives to its new voltage level.

The data sequence is monitored for each segment line, and charge redistribution is not used on specific segments at times that the data does not change. If a specific segment line at that specific point went from a previous data value of 0x1 to a new data value of 0x1, there is no reason to charge redistribute that segment line from a high voltage to a mid value voltage and back to high voltage again. Type A waveforms will always use charge redistribution on the segment lines while type B waveforms will look at the data sequence to turn off charge redistribution when the data does not change.

26.3.4.1 Type A Half Bias Triplex Waveform with Charge Redistribution

With charge redistribution between 1 to 4 LCDCLK cycles are selected with CHGRDST in LCD_DISPCTRL to short LCD pads to an intermediate voltage temporarily before each LCD pad transitions to its next voltage level. Due to the power savings charge redistribution is on by default. To switch off charge redistribution use CHGRDST in LCD_DISPCTRL. The logic looks at both the data per segment and the phase change to determine when to apply charge redistribution.

This example assumes a 32.768 kHz clock prescaled by 2 and a frame rate divider (LCD_FRAMERATE.FRDIV) of 90 which gives 30 frames per second. The charge redistribution is 1% of each phase. The normal power waveform is shown with segment 0 data of {1,1,0}.

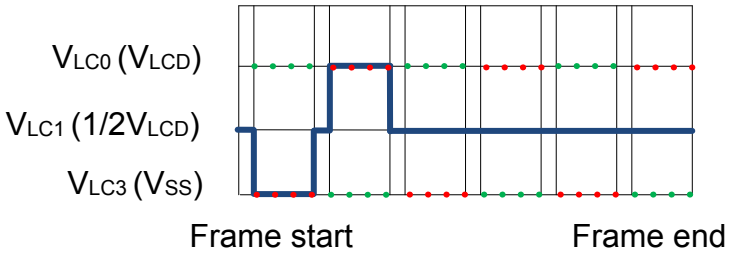


Figure 26.43. LCD Charge Redist - 1/2 Bias and Triplex Multiplexing - LCD.COM0

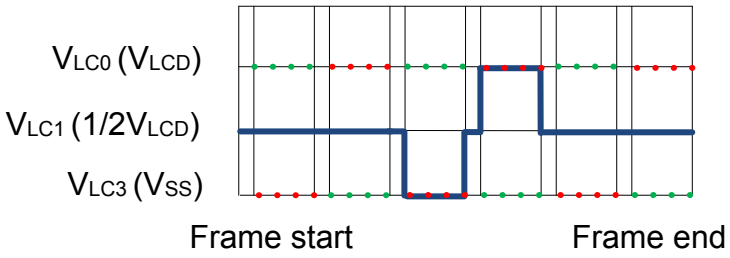


Figure 26.44. LCD Charge Redist - 1/2 Bias and Triplex Multiplexing - LCD.COM1

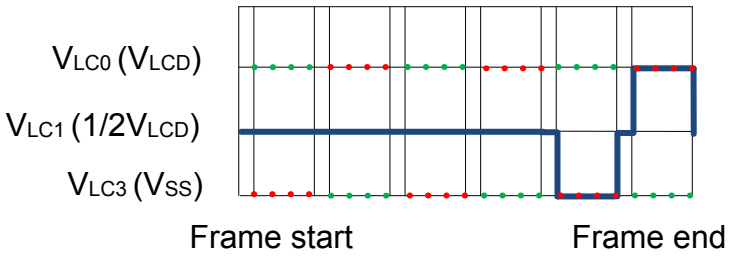


Figure 26.45. LCD Charge Redist - 1/2 Bias and Triplex Multiplexing - LCD.COM2

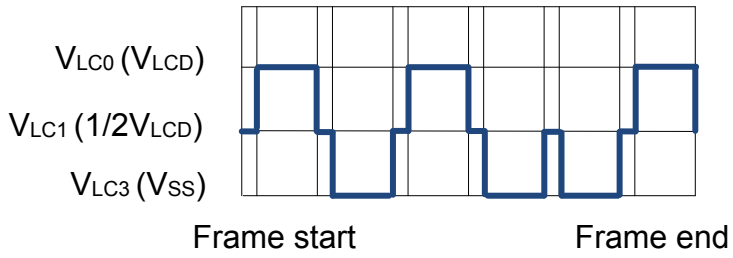


Figure 26.46. LCD Charge Redist - 1/2 Bias and Triplex Multiplexing - LCD.SEG0

Charge Redist with 1/2 bias and triplex multiplexing - LCD.SEG0-LCD.COM0

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.70 \times V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM0 will be ON with this waveform

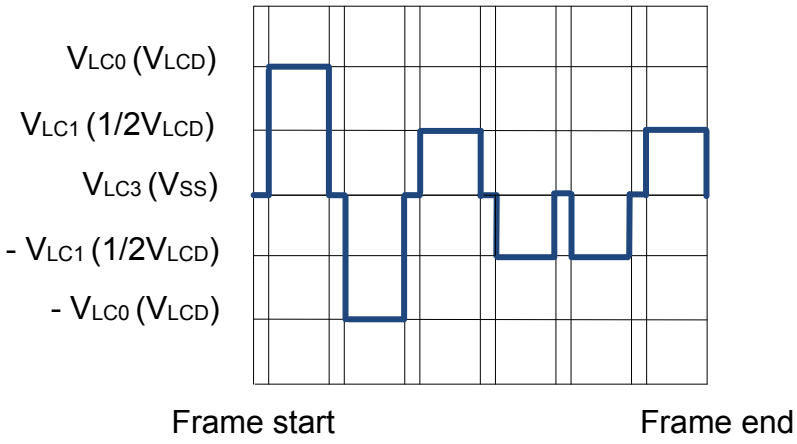


Figure 26.47. LCD Charge Redist - 1/2 Bias and Triplex Multiplexing - LCD.SEG0-LCD.COM0

Charge Redist with 1/3 bias and triplex multiplexing - LCD.SEG0-LCD.COM1

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.70 \times V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM1 will be ON with this waveform

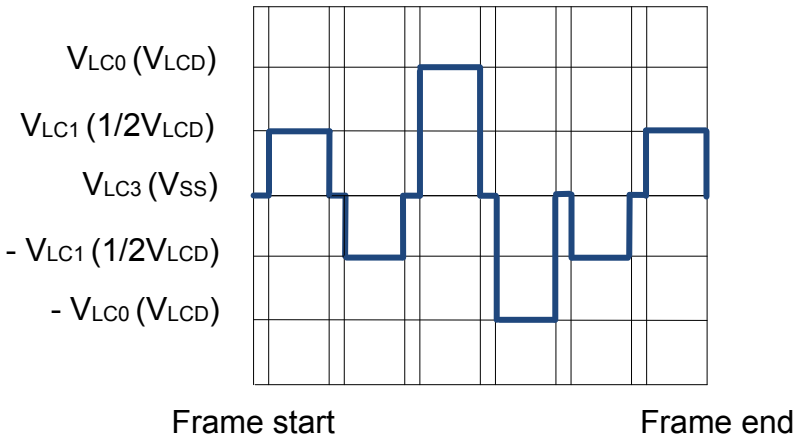


Figure 26.48. LCD Charge Redist - 1/2 Bias and Triplex Multiplexing - LCD.SEG0-LCD.COM1

Charge Redist with 1/3 bias and triplex multiplexing - LCD.SEG0-LCD.COM2

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.40 \times V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM2 will be OFF with this waveform

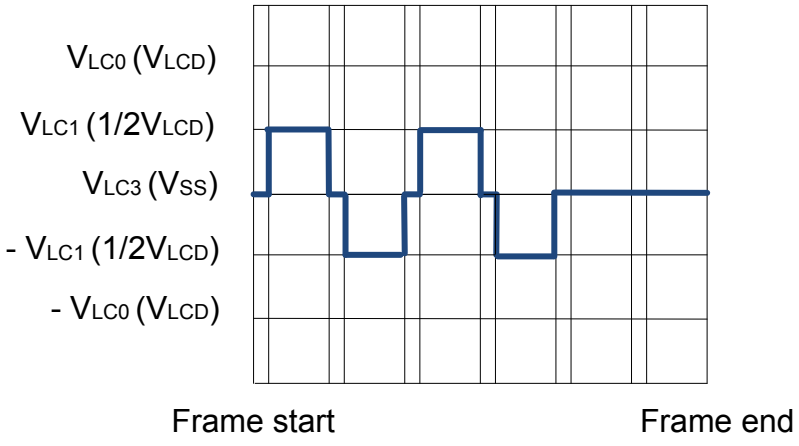


Figure 26.49. LCD Charge Redist - 1/2 Bias and Triplex Multiplexing - LCD.SEG0-LCD.COM2

26.3.4.2 Type B Half Bias Quadruplex Waveforms with Charge Redistribution

For type B waveforms there is no charge redistribution on the segment lines when the data does not change. The length of time spent in charge redistribution is exaggerated in the waveform to make it more visiable. Typically much less than 5% of the waveform is for charge redistribution.

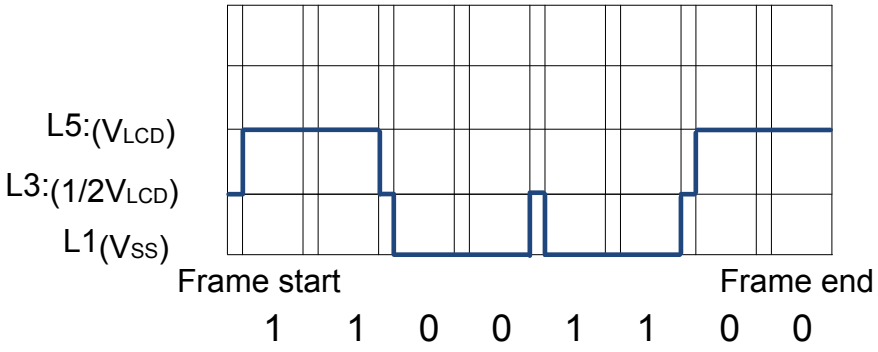


Figure 26.50. LCD Charge Redist - 1/2 Bias and Quadruplex Multiplexing - LCD.SEG0

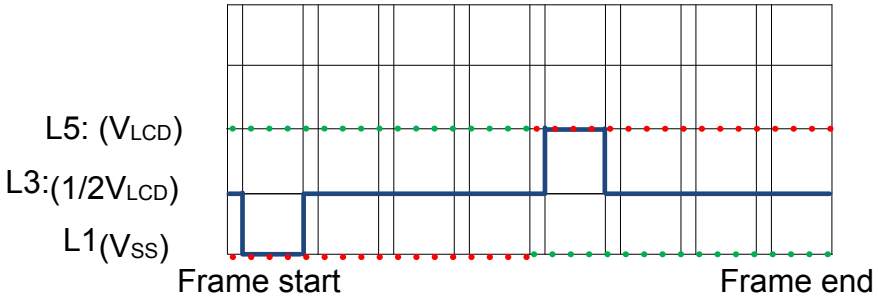


Figure 26.51. LCD Charge Redist - 1/2 Bias and Quadruplex Multiplexing - LCD.COM0

Charge Redist with 1/2 bias and quadruplex multiplexing - LCD.SEG0-LCD.COM0

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.76 \times V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM0 will be ON with this waveform

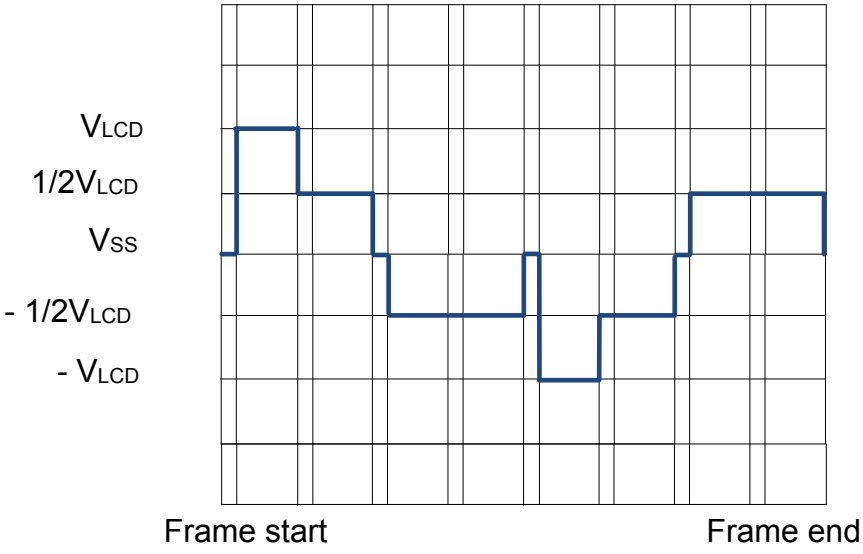


Figure 26.52. LCD Charge Redist - 1/2 Bias and Quadruplex Multiplexing - LCD.SEG0-LCD.COM0

26.3.4.3 Type B Third Bias Quadruplex Waveforms with Charge Redistribution

For type B waveforms there is no charge redistribution on the segment lines when the data does not change. The length of time spent in charge redistribution is exaggerated in the waveform to make it more visiable. Typically much less than 5% of the waveform is for charge redistribution.

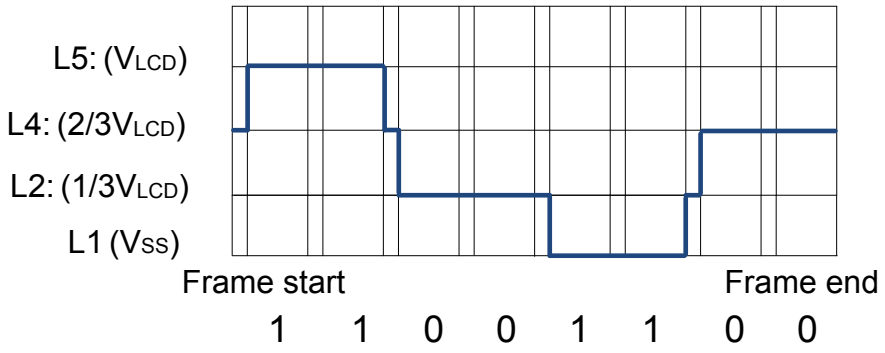


Figure 26.53. LCD Charge Redist - 1/3 Bias and Quadruplex Multiplexing - LCD.SEG0

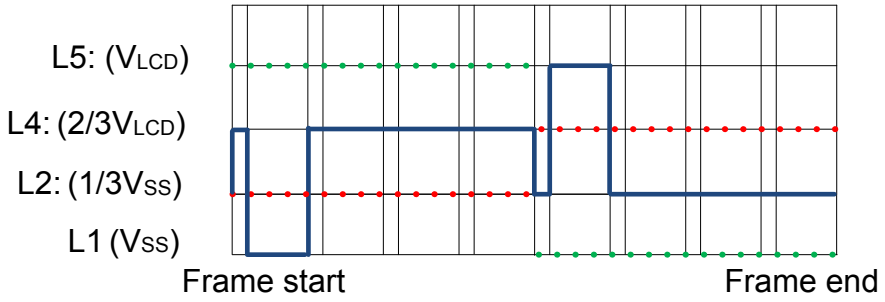


Figure 26.54. LCD Charge Redist - 1/3 Bias and Quadruplex Multiplexing - LCD.COM0

Charge Redist with 1/3 bias and quadruplex multiplexing - LCD.SEG0-LCD.COM0

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.58 \times V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM0 will be ON with this waveform

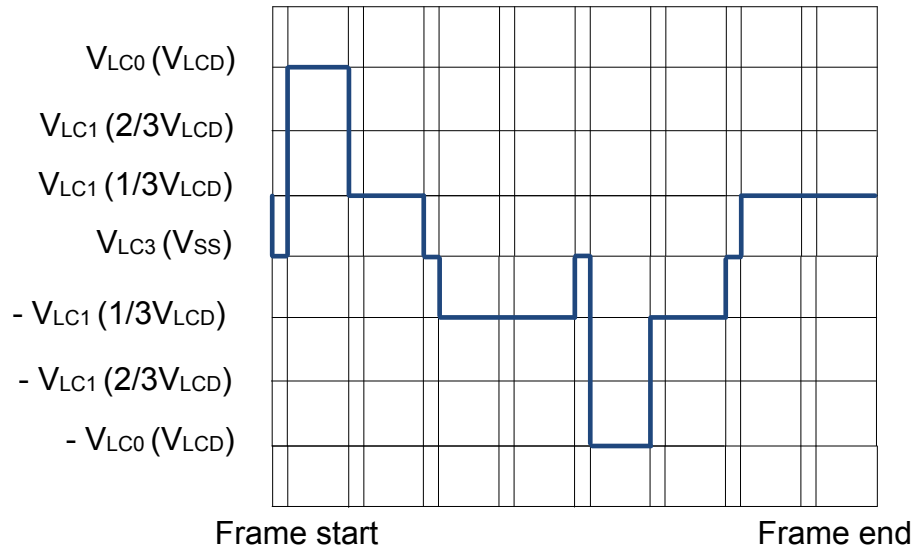


Figure 26.55. LCD Charge Redist - 1/3 Bias and Quadruplex Multiplexing - LCD.SEG0-LCD.COM0

26.3.4.4 Type B Quarter Bias Quadruplex Waveforms with Charge Redistribution

For type B waveforms there is no charge redistribution on the segment lines when the data does not change. The length of time spent in charge redistribution is exaggerated in the waveform to make it more visiable. Typically much less than 5% of the waveform is for charge redistribution.

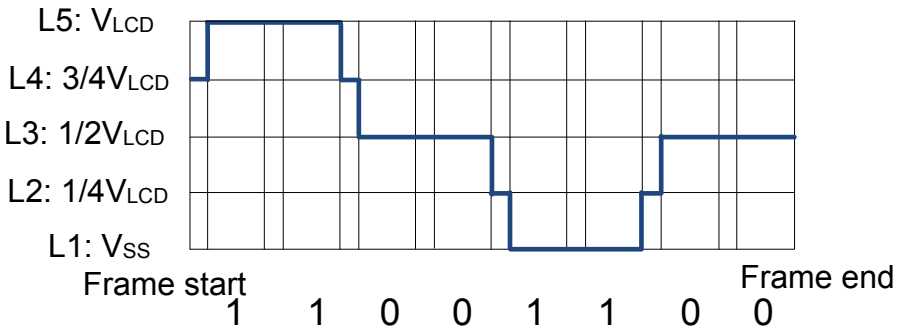


Figure 26.56. LCD Charge Redist - 1/4 Bias and Quadruplex Multiplexing - LCD.SEG0

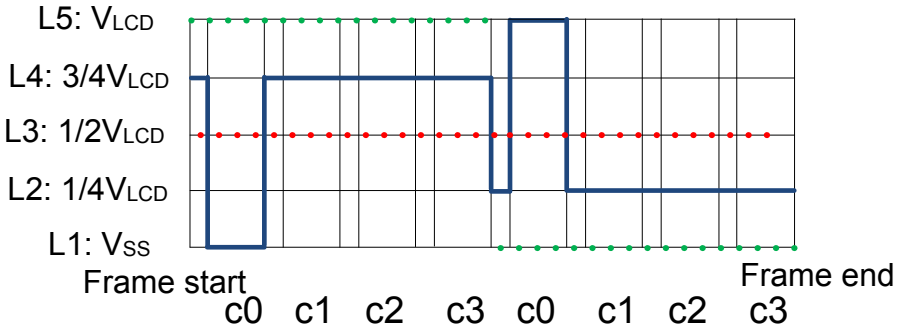


Figure 26.57. LCD Charge Redist - 1/4 Bias and Quadruplex Multiplexing - LCD.COM0

Charge Redist with ¼ bias and quadruplex multiplexing - LCD.SEG0-LCD.COM0

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.54 \times V_{LCD}$
- The LCD display pixel that is connected to LCD.SEG0 and LCD.COM0 will be ON with this waveform

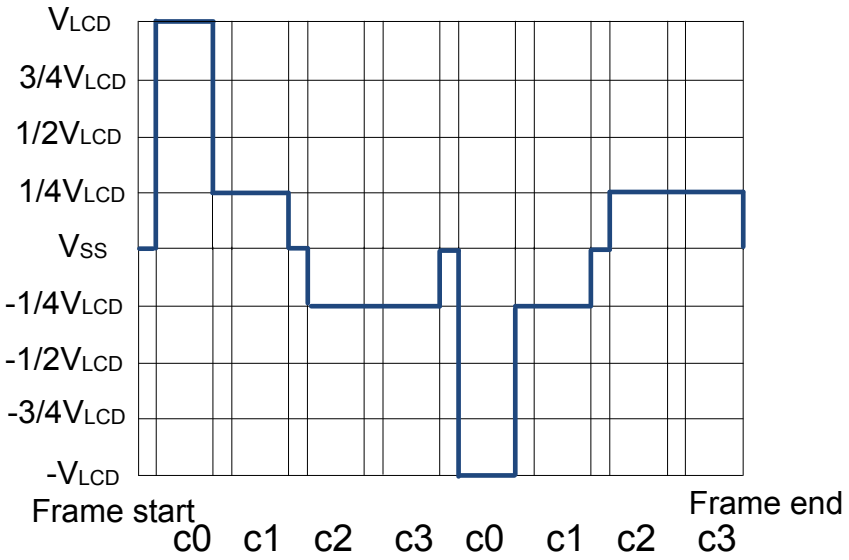


Figure 26.58. LCD Charge Redist - 1/4 Bias and Quadruplex Multiplexing - LCD.SEG0-LCD.COM0

26.3.5 LCD Contrast

Different LCD panels have different characteristics and also temperature may affect the characteristics of the LCD panels. To compensate for such variations, the LCD driver has a programmable contrast that adjusts the V_{LCD}. The contrast is set by LCD_BIASCTRL.VLCD, and can be adjusted relative to V_{DD} (V_{LCD}).

Table 26.5. LCD Contrast

Mode	Equation for vo44
step down and charge pump	2.25V + 50 mV * VLCD[4:0]

Note: Reset value is maximum contrast. vo34, vo24, and vo14 are equally spaced such as vo44*3/4, vo44*2/4, and vo44*1/4.

26.3.6 Voltage Levels and Mode Selection

The number of LCD bias levels is controlled by LCD_DISPCTRL.BIAS.

Voltage scaling of vo34, vo24, and vo14 are generated off of vo44 such as 3/4, 1/2, 1/4, and Vss for a BIAS setting of ONEFOURTH. For a BIAS setting of ONETHIRD, the scaled settings would be 2/3, 1/3, and Vss. For a BIAS setting of ONEHALF, the scaled settings would be 1/2 and Vss. For a BIAS setting of STATIC, the voltage are only vo44 and Vss.

Two modes are available for setting LCD voltage bias levels: step down mode or charge pump mode.

For the step down mode an external capacitor is regulated using an LCD comparator to maintain a vo44 voltage that is not greater than supply voltage.

For the charge pump mode, a voltage of up to twice the supply voltage is generated internally and maintained on an external capacitor. A second internal capacitor is used to pump up the external capacitor which maintains the vo44 voltage.

Pins assigned to LCD common or segment lines are capable of driving above the IOVDD supply to achieve the desired voltages.

V_{LCD} Selection

By default, the LCD driver runs in step down mode which is set with LCD_BIASCTRL.MODE or charge pump mode may be selected. Charge pump mode will boost V_{LCD} to a maximum value of approximately 1.95 * V_{DD}. Using V_{DD} = 1.8 V and setting VLCD register to its maximum value would charge up to about 3.5V. The charge pump is guaranteed to charge to the range of 2.4 V – 3.6 V by configuring LCD_BIASCTRL.VLCD to the appropriate code. Due to process variations the VLCD codes allow for a range slightly bigger than 2.4V - 3.6V. Any code that would have charged to a voltage greater than 3.6 V will be reduced to 3.6 V. Note that the charge pump circuit is not designed to operate with the selected charge pump voltage, V_{LCD}, smaller than V_{DD}.

The LCD_CP pin must be connected through a 4.7 uF capacitor to VSS.

26.3.7 Data Update

The LCD Driver logic that controls the output waveforms is clocked on the prescaled LCDCLK. The LCD data and Control Registers are clocked on the PCLK. To avoid metastability and unpredictable behavior, the data in the Segment Data (SEGDn) registers must be synchronized to the LCD driver logic. Also, it is important that data is updated at the beginning of an LCD frame since the segment waveform depends on the segment data and a change in the middle of a frame may lead to a DC-component in that frame. The LCD driver has dedicated functionality to synchronize data transfer to the LCD frames. The synchronization logic is applied to all data that need to be updated at the beginning of the LCD frames:

- LCD_SEGDn
- LCD_AREGA
- LCD_AREGB
- LCD_BACTRL
- LCD_UPDATECTRL

The different methods to update data are controlled by LCD_CTRL.UDCTRL.

Table 26.6. LCD Update Data Control (UDCTRL) Bits

UDCTRL	Mode	Description
00	REGULAR	The data transfer is controlled by SW and data synchronization is initiated by writing data to the buffers. Data is transferred as soon as possible, possibly creating a frame with a DC component on the LCD.
01	FCEVENT	The data transfer is done at the next event triggered by the Frame Counter (FC). See 26.3.7 Data Update for details on how to configure the Frame Counter. Optionally, the Frame Counter can also generate an interrupt at every event.
10	FRAMESTART	The data transfer is done at frame-start.
11	DISPLAYEVENT	The data transfer is done at a display event.

26.3.8 Direct Segment Control (DSC)

It is possible to gain direct control over the bias levels for each SEG/COM line by setting DSC in LCD_CTRL, overwriting the BIAS settings in LCD_DISPCTRL. The SEG lines bias levels can be set in SEGD0-SEGD3, while the COM line bias levels can be set in SEGD4. To represent the different bias levels, 4-bits per SEG lines are needed. For example, SEG0's bias levels can be set using SEGD0[3:0], and SEG1 can be controlled through SEGD1[3:0] etc. Bias level encoding is shown in Table 26.7 DSC BIAS Encoding on page 1071, and segment/common locations are shown in Table 26.8 DSC Segment and Common mapping on page 1071.

Table 26.7. DSC BIAS Encoding

SEGD	Bias setting
0000	tristate
0001	VSS
0010	1/3 or 1/4 V_{LCD}
0011	1/2 V_{LCD}
0100	2/3 or 3/4 V_{LCD}
0101	V_{LCD}

Table 26.8. DSC Segment and Common mapping

Register	L[31:28]	L[27:24]	L[23:20]	L[19:16]	L[15:12]	L[11:8]	L[7:4]	L[3:0]
SEGD0		seg24 / com4	seg20	seg16	seg12	seg8	seg4	seg0
SEGD1		seg25 / com5	seg21	seg17	seg13	seg9	seg5	seg1
SEGD2		seg26 / com6	seg22	seg18	seg14	seg10	seg6	seg2
SEGD3		seg27 / com7	seg23	seg19	seg15	seg11	seg7	seg3
AREGA							com2	com0
AREGB							com3	com1

26.3.9 Frame Counter (FC)

The Frame Counter is synchronized to the LCD frame start and will generate an event after a programmable number of frames. An FC event can trigger:

- LCD ready interrupt
- Blink (controlling the blink frequency)
- Next state in the Animation State Machine
- Data update if LCD_CTRL.UDCTRL = 01
- DMA transfer if LCD_BIASCTRL.DMAMODE = 01

The Frame Counter is a down counter. It is enabled by writing FCEN in LCD_BACTRL. Optionally, the Frame Counter can be prescaled so that the Frame Counter is decremented at:

- Every frame
- Every second frame
- Every fourth frame
- Every eighth frame

This is controlled by the FCPRESC in LCD_BACFG, see Table 26.9 FCPRESC on page 1072

Table 26.9. FCPRESC

FCPRESC	Mode	Description	General equation
00	Div1	$CLK_{FRAME}/1$	$CLK_{FC} = CLK_{FRAME}/2^{FCPRESC}$
01	Div2	$CLK_{FRAME}/2$	
10	Div4	$CLK_{FRAME}/4$	
11	Div8	$CLK_{FRAME}/8$	

The top value for the Frame Counter is set by FCTOP in LCD_BACTRL. Every time the frame counter reaches zero, it is reloaded with the top value, and at the same time an event, which can cause an interrupt, data update, blink, or an animation state transition is triggered.

$$CLK_{EVENT} = CLK_{FC}/(1 + FCTOP[5:0]) \text{ Hz}$$

Figure 26.59. LCD Event Frequency Equation

Figure 26.59 LCD Event Frequency Equation on page 1072 shows how to set-up the LCD event frequency. As an example, if the frame rate is 64 Hz, in order to have an LCD event frequency of 0.5 Hz, the following parameters should be set:

- Write FCPRESC to 3 => CLK_{FC} = 8 Hz (0.125 seconds)
- Write FCTOP to 15 => CLK_{EVENT} = 0.5 Hz (2 seconds)

If higher resolution is required, configure a lower prescaler value and increase FCPRESC in LCD_BACFG accordingly (e.g. FCPRESC = 2, FCTOP = 31).

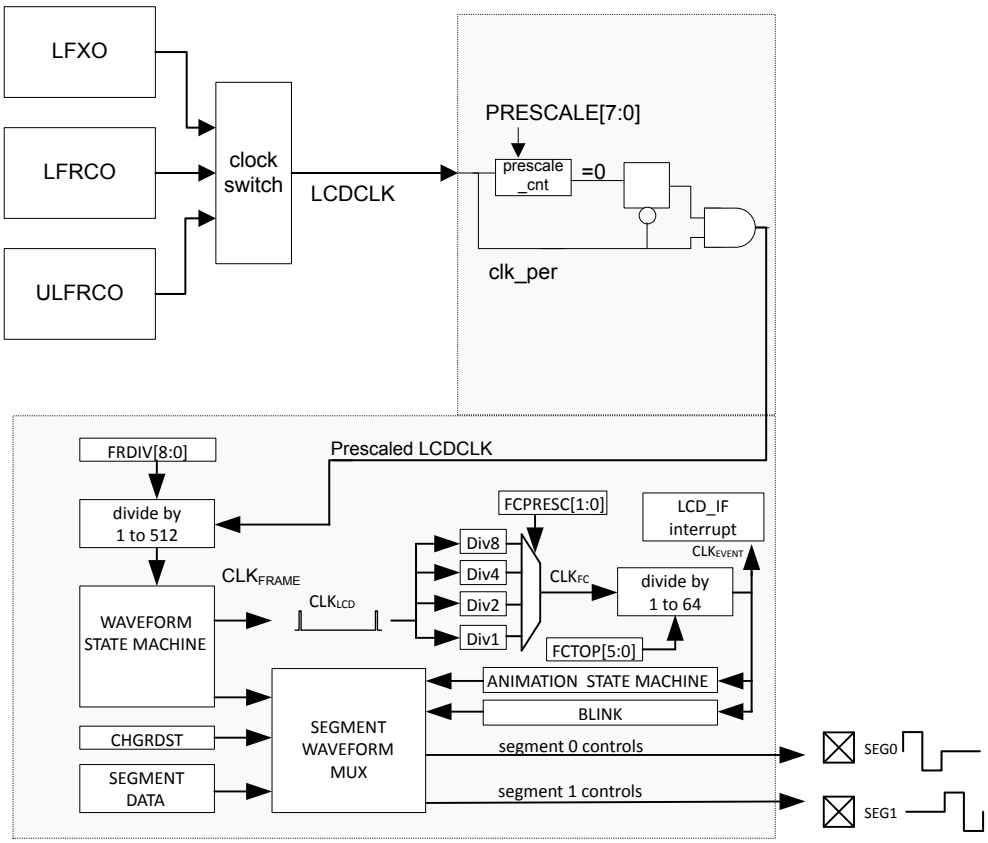


Figure 26.60. LCD Clock System in LCD Driver

26.3.10 Display Counter

The Display Counter is counting a number of frame counter events and can be used to interrupt or cause a DMA transfer at a slower rate than the frame counter. A Display counter event can trigger:

- Display counter interrupt
- Data update if LCD_CTRL.UDCTRL = 11
- DMA transfer if LCD_BIASCTRL.DMAMODE = 10

26.3.11 LCD Interrupt

The LCD interrupt can be used to synchronize data update. The FC interrupt flag is set at every LCD Frame Counter Event, which must be set-up separately. The interrupt is enabled by setting LCD_IEN.FC bit. The frame counter is enabled with LCD_BACTRL.FCEN.

There is also a display counter interrupt (LCD_IF.DISPLAY) that can be used to indicate when when the display count reaches zero. The interrupt is enabled with LCD_IEN.DISPLAY, and the display counter is enabled with LCD_BACTRL.DISPLAYCNTEN.

26.3.12 LCD DMA Transfer

The display counter can be used to have the LDMA transfer new segment data to the LCD when the display counter reaches 0x0 by setting LCD_BIASCTRL.DMAMODE to DMADISPLAY.

The LDMA transfer can be done with the frame counter by setting LCD_BIASCTRL.DMAMODE to DMAFC.

Note: An LDMA transfer typically is used to update segment data. After the LDMA writes new information to the LCD registers, the LCD peripheral must perform a hardware synchronization. See 26.3.15 LCD Synchronization. Subsequent LDMA requests will not be trggered by the LCD peripheral until hardware synchronization has completed.

26.3.13 Blink, Blank, and Animation Features

26.3.13.1 Blink

The LCD driver can be configured to blink, alternating all enabled segments between on and off. The blink frequency is given by the CLK_{EVENT} frequency, see 26.3.9 Frame Counter (FC). See 26.3.7 Data Update for details regarding synchronization of the blink feature. The FC must be on for blink to work.

26.3.13.2 Blank

Setting BLANK in LCD_BACTRL will output the “OFF” waveform on all enabled segments, effectively blanking the entire display. Writing the BLANK bit to zero disables the blanking and segment data will be output as normal.

26.3.13.3 Animation State Machine

The Animation State Machine makes it possible to enable different animations without updating the data registers, allowing specialized patterns running on the LCD panel while the microcontroller remains in Low Energy Mode and thus saving power consumption. The animation feature is available on segment 0 to 7 multiplexed with LCD.COM0. The animation is implemented as two programmable 8 bits registers that are shifted left or right every other Animation state for a total of 16 states. The animation feature is available on 8 segments multiplexed with LCD.COM0. The 8 segments can be either segments 0 to 7 or 8 to 15, depending on ALOC in LCD_BACTRL.

The shift operations applied to the shift registers are controlled by AREGASC and AREGBSC in LCD_BACTRL as shown in Table 26.10 LCD Animation Shift Register on page 1075. Note also that the FC must be on for animation to work, as it is the FC event that drives the animation state machine.

The BACFG.ASTATETOP can also be used to set the number of segments to values other than 8. Setting astatetop = 7 will give 8 segments, setting astate = 1 will give 2 segments. If the astatetop is set to 3, that will give 4 segments based on AREGA[3:0]. Setting AREGA[3:0] = 0001 and left shifting without using AREGB would give {0001, 0010, 0100, 1000, 0001, ...}.

Table 26.10. LCD Animation Shift Register

AREGnSC, n = A or B	Mode	Description
00	NOSHIFT	No Shift operation
01	SHIFTLEFT	Animation register is shifted left (LCD_AREGA is shifted every odd state, LCD_AREGB is shifted every even state)
10	SHIFTRIGHT	Animation register is shifted right (LCD_AREGA is shifted every odd state, LCD_AREGB is shifted every even state)
11	Reserved	Reserved

The two registers are either OR'ed or AND'ed to achieve the displayed animation pattern. This allows some segments to always be on while other segments appear to be moving. This is controlled by ALOGSEL in LCD_BACTRL as shown in Table 26.11 LCD Animation Pattern on page 1075. In addition, the regular segment data SEGD0[7:0] is OR'ed with the animation pattern to generate the resulting output.

Table 26.11. LCD Animation Pattern

ALOGSEL	Mode	Description
0	AND	LCD_AREGA and LCD_AREGB are AND'ed together
1	OR	LCD_AREGA and LCD_AREGB are OR'ed together

Each state is displayed one CLK_{EVENT} period, see 26.3.13.3 Animation State Machine. By reading ASTATE in LCD_STATUS, software can identify which state that is currently active in the state sequence. Note that the shifting operation is performed on internal registers that are not accessible in SW (when reading LCD_AREGA and LCD_AREGB, the data that was original written will also be read back). The SW must utilize the knowledge about the current state (ASTATE) to calculate what is currently output. ASTATE is cleared when LCD_AREGA or LCD_AREGB are updated with new values. See Table 26.12 LCD Animation Example on page 1075 for an example.

Table 26.12. LCD Animation Example

ASTATE	LCD_AREGA	LCD_AREGB	Resulting Data
0	11000000	11000000	11000000
1	01100000	11000000	11100000
2	01100000	01100000	01100000
3	00110000	01100000	01110000
4	00110000	00110000	00110000
5	00011000	00110000	00111000

ASTATE	LCD_AREGA	LCD_AREGB	Resulting Data
6	00011000	00011000	00011000
7	00001100	00011000	00011100
8	00001100	00001100	00001100
9	00000110	00001100	00001110
10	00000110	00000110	00000110
11	00000011	00000110	00000111
12	00000011	00000011	00000011
13	10000001	00000011	10000011
14	10000001	10000001	10000001
15	11000000	10000001	11000001

In the table, AREGASC = 10, AREGBSC = 10, ALOGSEL = 1 and the resulting data is to be displayed on segment lines 7-0 multiplexed with LCD.COM0.

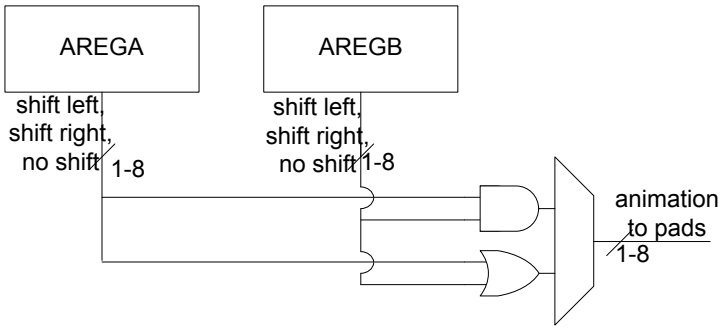


Figure 26.61. LCD Block Diagram of the Animation Circuit

Animation Example

- Write data into the animation registers LCD_AREGA, LCD_AREGB
- Enable the correct shift direction (if any)
- Decide which logical function to perform on the registers
  - ALOGSEL = 0: Data_out = LCD_AREGA & LCD_AREGB
  - ALOGSEL = 1:Data_out = LCD_AREGA | LCD_AREGB
- Configure the right animation period (CLK_{EVENT})
- Enable the animation pattern and frame counter (AEN = 1, FCEN = 1)

For updating data in the LCD while it is running an animation, and the new animation data depends on the pattern visible on the LCD, see the following example.

- Enable the LCD interrupt (the interrupt will be triggered simultaneously as the Animation State machine changes state)
- In the interrupt handler, read back the current state (ASTATE)
- Knowing the current state of the Animation State Machine makes it possible to calculate what data that is currently output
- Modify data as required (Data will be updated at the next Frame Counter Event). It is important that new data is written before the next Frame Counter Event.

26.3.14 LCD in Low Energy Modes

As long as the LCDCLK is running, the LCD controller continues to output LCD waveforms according to the data that is currently synchronized to the LCD Driver logic. In addition, the following features are still active if enabled:

- Animation State Machine
- Blink
- LCD Event Interrupt

26.3.15 LCD Synchronization

Writes to the registers (BACTRL, AREGA, AREGB, and SEGDn) require a hardware synchronization sequence. This sequence can be started with either a LCD_CMD.LOAD pulse or with a LCD_UPDATECTRL.LOADADDR matching the most recent register address. Once the sequence starts, any writes to the (BACTRL, AREGA, AREGB, or SEGDn) registers will be blocked and cause a bus fault. To prevent a bus fault, first check that LCD_STATUS.LOADBUSY is not set (0x1). Then write as many of the registers (BACTRL, AREGA, AREGB, and SEGDn) as needed. Then start the sequence. Do not write to these registers again until LCD_STATUS.LOADBUSY has cleared.

Once the sequence is started during EM0, the energy mode can switch to EM2 if desired. The rest of the synchronization does not require EM0. The data synchronizes on a UDCTRL event boundary, so it may take a very long time before synchronization has completed. If a register change is desired before the synchronization completes, use the LCD_CMD.CLEAR pulse to clear out the synchronization and stop the write from going through to the LCDCLK domain. Check the LCD_STATUS.LOADBUSY to see when the synchronization has cleared out, or use the LCD_IF.SYNCBUSYDONE interrupt to indicate that the previous synchronization is cleared.

26.4 LCD Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	LCD_IPVERSION	R	IPVERSION
0x004	LCD_EN	RW ENABLE	Enable
0x008	LCD_SWRST	RW SWRST	Software Reset
0x00C	LCD_CTRL	RW CONFIG	Control Register
0x010	LCD_CMD	W	Command Register
0x014	LCD_DISPCTRL	RW CONFIG	Display Control Register
0x018	LCD_BACFG	RW CONFIG	Blink and Animation Config Register
0x01C	LCD_BACTRL	RW	Blink and Animation Control Register
0x020	LCD_STATUS	RH	Status Register
0x024	LCD_AREGA	RW	Animation Register a
0x028	LCD_AREGB	RW	Animation Register B
0x02C	LCD_IF	RWH INTFLAG	Interrupt Enable Register
0x030	LCD_IEN	RW	Interrupt Enable
0x034	LCD_BIASCTRL	RW	Analog BIAS Control
0x038	LCD_DISPCTRLX	RW CONFIG	Display Control Extended
0x040	LCD_SEGD0	RW	Segment Data Register 0
0x048	LCD_SEGD1	RW	Segment Data Register 1
0x050	LCD_SEGD2	RW	Segment Data Register 2
0x058	LCD_SEGD3	RW	Segment Data Register 3
0x060	LCD_SEGD4	RW	Segment Data Register 4
0x068	LCD_SEGD5	RW	Segment Data Register 5
0x070	LCD_SEGD6	RW	Segment Data Register 6
0x078	LCD_SEGD7	RW	Segment Data Register 7
0x0C0	LCD_UPDATECTRL	RW	Update Control
0x0F0	LCD_FRAMERATE	RW CONFIG	Frame Rate
0x1000	LCD_IPVERSION_SET	R	IPVERSION
0x1004	LCD_EN_SET	RW ENABLE	Enable
0x1008	LCD_SWRST_SET	RW SWRST	Software Reset
0x100C	LCD_CTRL_SET	RW CONFIG	Control Register
0x1010	LCD_CMD_SET	W	Command Register
0x1014	LCD_DISPCTRL_SET	RW CONFIG	Display Control Register
0x1018	LCD_BACFG_SET	RW CONFIG	Blink and Animation Config Register
0x101C	LCD_BACTRL_SET	RW	Blink and Animation Control Register
0x1020	LCD_STATUS_SET	RH	Status Register
0x1024	LCD_AREGA_SET	RW	Animation Register a

Offset	Name	Type	Description
0x1028	LCD_AREGB_SET	RW	Animation Register B
0x102C	LCD_IF_SET	RWH INTFLAG	Interrupt Enable Register
0x1030	LCD_IEN_SET	RW	Interrupt Enable
0x1034	LCD_BIASCTRL_SET	RW	Analog BIAS Control
0x1038	LCD_DISPCTRLX_SET	RW CONFIG	Display Control Extended
0x1040	LCD_SEGD0_SET	RW	Segment Data Register 0
0x1048	LCD_SEGD1_SET	RW	Segment Data Register 1
0x1050	LCD_SEGD2_SET	RW	Segment Data Register 2
0x1058	LCD_SEGD3_SET	RW	Segment Data Register 3
0x1060	LCD_SEGD4_SET	RW	Segment Data Register 4
0x1068	LCD_SEGD5_SET	RW	Segment Data Register 5
0x1070	LCD_SEGD6_SET	RW	Segment Data Register 6
0x1078	LCD_SEGD7_SET	RW	Segment Data Register 7
0x10C0	LCD_UPDATECTRL_SET	RW	Update Control
0x10F0	LCD_FRAMERATE_SET	RW CONFIG	Frame Rate
0x2000	LCD_IPVERSION_CLR	R	IPVERSION
0x2004	LCD_EN_CLR	RW ENABLE	Enable
0x2008	LCD_SWRST_CLR	RW SWRST	Software Reset
0x200C	LCD_CTRL_CLR	RW CONFIG	Control Register
0x2010	LCD_CMD_CLR	W	Command Register
0x2014	LCD_DISPCTRL_CLR	RW CONFIG	Display Control Register
0x2018	LCD_BACFG_CLR	RW CONFIG	Blink and Animation Config Register
0x201C	LCD_BACTRL_CLR	RW	Blink and Animation Control Register
0x2020	LCD_STATUS_CLR	RH	Status Register
0x2024	LCD_AREGA_CLR	RW	Animation Register a
0x2028	LCD_AREGB_CLR	RW	Animation Register B
0x202C	LCD_IF_CLR	RWH INTFLAG	Interrupt Enable Register
0x2030	LCD_IEN_CLR	RW	Interrupt Enable
0x2034	LCD_BIASCTRL_CLR	RW	Analog BIAS Control
0x2038	LCD_DISPCTRLX_CLR	RW CONFIG	Display Control Extended
0x2040	LCD_SEGD0_CLR	RW	Segment Data Register 0
0x2048	LCD_SEGD1_CLR	RW	Segment Data Register 1
0x2050	LCD_SEGD2_CLR	RW	Segment Data Register 2
0x2058	LCD_SEGD3_CLR	RW	Segment Data Register 3
0x2060	LCD_SEGD4_CLR	RW	Segment Data Register 4
0x2068	LCD_SEGD5_CLR	RW	Segment Data Register 5
0x2070	LCD_SEGD6_CLR	RW	Segment Data Register 6

Offset	Name	Type	Description
0x2078	LCD_SEGD7_CLR	RW	Segment Data Register 7
0x20C0	LCD_UPDATECTRL_CLR	RW	Update Control
0x20F0	LCD_FRAMERATE_CLR	RW CONFIG	Frame Rate
0x3000	LCD_IPVERSION_TGL	R	IPVERSION
0x3004	LCD_EN_TGL	RW ENABLE	Enable
0x3008	LCD_SWRST_TGL	RW SWRST	Software Reset
0x300C	LCD_CTRL_TGL	RW CONFIG	Control Register
0x3010	LCD_CMD_TGL	W	Command Register
0x3014	LCD_DISPCTRL_TGL	RW CONFIG	Display Control Register
0x3018	LCD_BACFG_TGL	RW CONFIG	Blink and Animation Config Register
0x301C	LCD_BACTRL_TGL	RW	Blink and Animation Control Register
0x3020	LCD_STATUS_TGL	RH	Status Register
0x3024	LCD_AREGA_TGL	RW	Animation Register a
0x3028	LCD_AREGB_TGL	RW	Animation Register B
0x302C	LCD_IF_TGL	RWH INTFLAG	Interrupt Enable Register
0x3030	LCD_IEN_TGL	RW	Interrupt Enable
0x3034	LCD_BIASCTRL_TGL	RW	Analog BIAS Control
0x3038	LCD_DISPCTRLX_TGL	RW CONFIG	Display Control Extended
0x3040	LCD_SEGD0_TGL	RW	Segment Data Register 0
0x3048	LCD_SEGD1_TGL	RW	Segment Data Register 1
0x3050	LCD_SEGD2_TGL	RW	Segment Data Register 2
0x3058	LCD_SEGD3_TGL	RW	Segment Data Register 3
0x3060	LCD_SEGD4_TGL	RW	Segment Data Register 4
0x3068	LCD_SEGD5_TGL	RW	Segment Data Register 5
0x3070	LCD_SEGD6_TGL	RW	Segment Data Register 6
0x3078	LCD_SEGD7_TGL	RW	Segment Data Register 7
0x30C0	LCD_UPDATECTRL_TGL	RW	Update Control
0x30F0	LCD_FRAMERATE_TGL	RW CONFIG	Frame Rate

26.5 LCD Register Description

26.5.1 LCD_IPVERSION - IPVERSION

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x2																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x2	R	**IPVERSION**
	IPVERSION			

26.5.2 LCD_EN - Enable

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status**
	Disablement status			
0	EN	0x0	RW	**Enable**
	Enable			

Value	Mode	Description
0	DISABLE	Disable
1	ENABLE	Enable

26.5.3 LCD_SWRST - Software Reset

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	W
Name																															RESETTING	SWRST

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESETTING Status of Reset	0x0	R	**Software reset busy status**
0	SWRST Software reset command	0x0	W	**Software reset command**

26.5.4 LCD_CTRL - Control Register

Offset	Bit Position																														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0x0										0x4				0x0														0x0		
Access		RW										RW				RW														RW		
Name		PRESCALE										WARMUPDLY				DSC														UDCTRL		

Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
30:24	PRESCALE	0x0	RW	**Presclae**
	Prescales the LCDCLK from CMU			
23:21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20:18	WARMUPDLY	0x4	RW	**Warmup Delay**
	Keep LCD pad tristated until warm up delay. The default value of 275ms is assuming a 4.7uF external capacitor.			

Value	Mode	Description
0	WARMUP1	1mswarm up
1	WARMUP31	31ms warm up
2	WARMUP63	62ms warm up
3	WARMUP125	125ms warm up
4	WARMUP250	250ms warm up
5	WARMUP500	500ms warm up
6	WARMUP1000	1000ms warm up
7	WARMUP2000	2000ms warm up

17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	DSC	0x0	RW	**Direct Segment Control**
	This bit enables direct control over bias levels for each SEG/COM line.			

Value	Mode	Description
0	DISABLE	DSC disable
1	ENABLE	DSC enable

15:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:1	UDCTRL	0x0	RW	**Update Data Control**

Bit	Name	Reset	Access	Description
	These bits control how data from the SEGDn registers are transferred to the LCD driver.			

Value	Mode	Description
0	REGULAR	The data transfer is controlled by SW. Transfer is performed as soon as possible on the next CTRL.PRESCALE clock. This is primarily available for debug only since only some of the new SEGMENT data may be ready by the time of the UPDATE. This should not be used with interrupts since partially updating SEGMENT data may have indeterminant results.
1	FRAMESTART	Data is loaded continuously at every frame start
2	FCEVENT	The data transfer is done at the next Frame Counter event
3	DISPLAYEVENT	The data transfer is done at the next Display Counter event

0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*

26.5.5 LCD_CMD - Command Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															W(nB)	W(nB)
Name																															CLEAR	LOAD

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	CLEAR	0x0	W(nB)	**Clear command**
	This command clears the LOAD command so that new register values can be written instead of waiting for the next UDCTRL before LOAD will complete.			
0	LOAD	0x0	W(nB)	**Load command**
	Load command begins synchronization to transfer HV registers to CLK_PER domain. This may take a considerable amount of time to complete depending on the LCD_UPDATECTRL.UDCTRL setting.			

26.5.6 LCD_DISPCTRL - Display Control Register

Offset	Bit Position																															
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset							0x0			0x1																		0x0		0x0		
Access							RW			RW																		RW		RW		
Name							BIAS			CHGRDST																		WAVE		MUX		

Bit	Name	Reset	Access	Description
31:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25:24	BIAS	0x0	RW	**Bias Configuration**

These bits set the bias mode for the LCD Driver.

Value	Mode	Description
0	STATIC	Static
1	ONEHALF	1/2 Bias
2	ONETHIRD	1/3 Bias
3	ONEFOURTH	1/4 Bias

23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:20	CHGRDST	0x1	RW	**Charge Redistribution Cycles**

Selects number of prescaled low frequency clock cycles for charge redistribution.

Value	Mode	Description
0	DISABLE	Disable charge redistribution.
1	ONE	Use 1 prescaled low frequency clock cycle for charge redistribution.
2	TWO	Use 2 prescaled low frequency clock cycles for charge redistribution.
3	THREE	Use 3 prescaled low frequency clock cycles for charge redistribution.
4	FOUR	Use 4 prescaled low frequency clock cycles for charge redistribution.

19:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	WAVE	0x0	RW	**Waveform Selection**

This bit configures the output waveform.

Value	Mode	Description
0	TYPEB	Type B waveform

Bit	Name	Reset	Access	Description
	1	TYPEA		Type A waveform
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	MUX	0x0	RW	**Mux Configuration**

These bits set the multiplexing mode for the LCD Driver.

Value	Mode	Description
0	STATIC	Static
1	DUPLEX	Duplex
2	TRIPLEX	Triplex
3	QUADRUPLEX	Quadruplex
5	SEXTAPLEX	Sextaplex
7	OCTAPLEX	Octaplex

26.5.7 LCD_BACFG - Blink and Animation Config Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0						0x0															0x7		
Access									RW						RW															RW		
Name									FCTOP						FCPRESC															ASTATETOP		

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:18	FCTOP	0x0	RW	**Frame Counter Top**
	These bits contain the Top Value for the Frame Counter: CLK_EVENT = CLK_FC / (1 + FCTOP[5:0]).			
17:16	FCPRESC	0x0	RW	**Frame Counter Prescaler**
	These bits controls the prescaling value for the Frame Counter input clock.			

Value	Mode	Description
0	DIV1	every frame clock
1	DIV2	every 2nd frame clock
2	DIV4	every 4th frame clock
3	DIV8	every 8th frame clock

15:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	ASTATETOP	0x7	RW	**ASTATE top cnt**
	Reload astate once the astatetop is reached			

26.5.8 LCD_BACTRL - Blink and Animation Control Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x0																			0x0	0x0	0x0	0x0		0x0		0x0	0x0	0x0
Access				RW																			RW	RW	RW	RW		RW		RW	RW	RW
Name				ALOC																			DISPLAYCNTEN	FCEN	ALOGSEL	AREGBSC		AREGASC		AEN	BLANK	BLINKEN

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28	ALOC	0x0	RW	**Animation Location**
	Set the LCD segments which animation applies to			
	Value	Mode		Description
	0	SEG0TO7		Animation appears on segments 0 to 7
	1	SEG8TO15		Animation appears on segments 8 to 15
27:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	DISPLAYCNTEN	0x0	RW	**Display Counter Enable**
	Enable display counter for use with creating an interrupt or a DMA transfer to update the display			
	Value	Mode		Description
	0	DISABLE		Disable the display counter
	1	ENABLE		Enable the display counter
8	FCEN	0x0	RW	**Frame Counter Enable**
	When this bit is set, the frame counter is enabled.			
7	ALOGSEL	0x0	RW	**Animate Logic Function Select**
	When this bit is set, the animation registers are AND'ed together. When this bit is cleared, the animation registers are OR'ed together.			
	Value	Mode		Description
	0	AND		AREGA and AREGB AND'ed
	1	OR		AREGA and AREGB OR'ed
6:5	AREGBSC	0x0	RW	**Animate Register B Shift Control**
	These bits controls the shift operation that is performed on Animation register B.			
	Value	Mode		Description
	0	NOSHIFT		No Shift operation on Animation Register B

Bit	Name	Reset	Access	Description
	1	SHIFTLEFT		Animation Register B is shifted left
	2	SHIFTRIGHT		Animation Register B is shifted right
4:3	AREGASC	0x0	RW	**Animate Register A Shift Control**
	These bits controls the shift operation that is performed on Animation register A.			
	Value	Mode		Description
	0	NOSHIFT		No Shift operation on Animation Register A
	1	SHIFTLEFT		Animation Register A is shifted left
	2	SHIFTRIGHT		Animation Register A is shifted right
2	AEN	0x0	RW	**Animation Enable**
	When this bit is set, the animate function is enabled.			
1	BLANK	0x0	RW	**Blank Display**
	When this bit is set, all segment output waveforms are configured to blank the LCD display. The Segment Data Registers are not affected when writing this bit.			
	Value	Mode		Description
	0	DISABLE		Display is not "blanked"
	1	ENABLE		Display is "blanked"
0	BLINKEN	0x0	RW	**Blink Enable**
	When this bit is set, the Blink function is enabled. Every "ON" segment will alternate between on and off at every Frame Counter Event.			

26.5.9 LCD_STATUS - Status Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0			0x0						0x0		
Access																					R			R						R		
Name																					LOADBUSY			BLINK						ASTATE		

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	LOADBUSY	0x0	R	**Load Synchronization is busy**
	Load synchronization is busy. Doing any writes to HV registers will not go through and will cause a bus fault			
10:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	BLINK	0x0	R	**Blink State**
	This bits indicates the blink status. If this bit is 1, all segments are off. If this bit is 0, the segments(LCD_SEGDxn) which are set to 1 are on.			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	ASTATE	0x0	R	**Current Animation State**
	Contains the current animation state (0-15).			

26.5.10 LCD_AREGA - Animation Register a

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									AREGA							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	AREGA	0x0	RW	**Animation Register A Data**
	This register contains the A data for generating animation pattern.			

26.5.11 LCD_AREGB - Animation Register B

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									AREGB							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	AREGB	0x0	RW	**Animation Register B Data**
	This register contains the B data for generating animation pattern.			

26.5.12 LCD_IF - Interrupt Enable Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														SYNCBUSYDONE	DISPLAY	FC

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	SYNCBUSYDONE	0x0	RW	**Synchronization is Done**
	HV register load synchronization has completed			
1	DISPLAY	0x0	RW	**Display Update Event**
	Display Update Event			
0	FC	0x0	RW	**Frame Counter**
	Frame Counter qualified with CTRL.UDCTRL			

26.5.13 LCD_IEN - Interrupt Enable

Offset	Bit Position																														
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														SYNCBUSYDONE	DISPLAY	FC

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	SYNCBUSYDONE	0x0	RW	**Sync Busy Done**
	Load Synchronization of HV registers has completed			
1	DISPLAY	0x0	RW	**Display Update Event**
	Display Update Event			
0	FC	0x0	RW	**Frame Counter**
	Frame Counter			

26.5.14 LCD_BIASCTRL - Analog BIAS Control

Offset	Bit Position																															
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0					0x0				0x0		0x1F								0x0			0x0			0x0			0x0			
Access	RW					RW				RW		RW								RW			RW			RW			RW			
Name	DMAMODE					LCDGATE				VDDXSEL		VLCD								MODE			BUFBIAS			BUFDRV			RESISTOR			

Bit	Name	Reset	Access	Description
31:30	DMAMODE	0x0	RW	**DMA Mode**
	DMA mode			
	Value	Mode		Description
	0	DMADISABLE		No DMA requests are generated
	1	DMAFC		DMA request on frame counter event. This will also start a DMA transfer during EM23.
	2	DMADISPLAY		DMA request on display counter event. This will also start a DMA transfer during EM23.
29:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26	LCDGATE	0x0	RW	**LCD Gate**
	Tristate the LCD pins. The gating or un-gating occurs on Frame boundaries.			
	Value	Mode		Description
	0	UNGATE		LCD BIAS voltages driven onto pins.
	1	GATE		LCD BIAS MUX tristated at the pins.
25:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22	VDDXSEL	0x0	RW	**VDDX select**
	VDDX select			
	Value	Mode		Description
	0	DVDD		Connect charge pump to digital DVDD supply
	1	AVDD		Connect charge pump to analog AVDD supply
21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20:16	VLCD	0x1F	RW	**VLCD voltage level**
	This controls the VLCD supply voltage.			
15:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

Bit	Name	Reset	Access	Description
12	MODE	0x0	RW	**Mode Setting**
	This field determines the LCD mode of operation.			

Value	Mode	Description
0	STEPDOWN	Use step down control with VLCD less than VDDX. Use VLCD[4:0] to control VLCD level, and use SPEED to adjust VLCD drive strength.
1	CHARGEPUMP	Use the charge pump to pump VLCD above VDDX.

11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	BUFBIAS	0x0	RW	**Buffer Bias Setting**
	Operating bias current for the buffers.			
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	BUFDRV	0x0	RW	**Buffer Drive Strength**
	Buffer driver strength.			
3:0	RESISTOR	0x0	RW	**Resistor strength**
	Adjust the resistor strength for warmup drive of the external capacitor and for step down mode			

26.5.15 LCD_DISPCTRLX - Display Control Extended

Offset	Bit Position																															
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0					
Access																											RW					
Name																											DISPLAYDIV					

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:0	DISPLAYDIV	0x0	RW	**Display Divider**
	Creates an interrupt for every DISPLAYDIV+1 number of frame events. This can be used to wakeup and have the DMA load in new SEGMENT registers values			

26.5.16 LCD_SEGD0 - Segment Data Register 0

Offset	Bit Position																													
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0																											
Access					RW																											
Name					SEGD0																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:0	SEGD0	0x0	RW	**COM0 Segment Data Low**
	This register contains segment data for segment lines during COM0.			

26.5.17 LCD_SEGD1 - Segment Data Register 1

Offset	Bit Position																													
0x048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0																											
Access					RW																											
Name					SEGD1																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:0	SEGD1	0x0	RW	**COM1 Segment Data Low**
	This register contains segment data for segment lines during COM1.			

26.5.18 LCD_SEGD2 - Segment Data Register 2

Offset	Bit Position																													
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0																											
Access					RW																											
Name					SEGD2																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:0	SEGD2	0x0	RW	**COM2 Segment Data Low**
	This register contains segment data for segment lines during COM2.			

26.5.19 LCD_SEGD3 - Segment Data Register 3

Offset	Bit Position																													
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0																											
Access					RW																											
Name					SEGD3																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:0	SEGD3	0x0	RW	**COM3 Segment Data Low**
	This register contains segment data for segment lines during COM3.			

26.5.20 LCD_SEGD4 - Segment Data Register 4

Offset	Bit Position																													
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0																											
Access					RW																											
Name					SEGD4																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:0	SEGD4	0x0	RW	**COM4 Segment Data**
	This register contains segment data for segment lines during COM4.			

26.5.21 LCD_SEGD5 - Segment Data Register 5

Offset	Bit Position																													
0x068	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0																											
Access					RW																											
Name					SEGD5																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:0	SEGD5	0x0	RW	**COM5 Segment Data**
	This register contains segment data for segment lines during COM5.			

26.5.22 LCD_SEGD6 - Segment Data Register 6

Offset	Bit Position																													
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0																											
Access					RW																											
Name					SEGD6																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:0	SEGD6	0x0	RW	**COM6 Segment Data**
	This register contains segment data for segment lines during COM6.			

26.5.23 LCD_SEGD7 - Segment Data Register 7

Offset	Bit Position																													
0x078	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0																											
Access					RW																											
Name					SEGD7																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:0	SEGD7	0x0	RW	**COM7 Segment Data**
	This register contains segment data for segment lines during COM7.			

26.5.24 LCD_UPDATECTRL - Update Control

Offset	Bit Position																															
0x0C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x0								0x0								
Access																RW								RW								
Name																LOADADDR								AUTOLOAD								

Bit	Name	Reset	Access	Description
31:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16:13	LOADADDR	0x0	RW	**Load Address** Auto Load address which will start the synchronization from CLK_BUS to CLK_PER and not allow additional writes

Value	Mode	Description
0	BACTRLWR	Starts synchronizing registers from CLK_BUS to CLK_PER after a write to BACTRL. Use with UPDATECTRL.AUTOLOAD
1	AREGAWR	Starts synchronizing registers from CLK_BUS to CLK_PER after a write to AREGA. Use with UPDATECTRL.AUTOLOAD
2	AREGBWR	Starts synchronizing registers from CLK_BUS to CLK_PER after a write to AREGB. Use with UPDATECTRL.AUTOLOAD
3	SEGD0WR	Starts synchronizing registers from CLK_BUS to CLK_PER after a write to SEGD0. Use with UPDATECTRL.AUTOLOAD
4	SEGD1WR	Starts synchronizing registers from CLK_BUS to CLK_PER after a write to SEGD1. Use with UPDATECTRL.AUTOLOAD
5	SEGD2WR	Starts synchronizing registers from CLK_BUS to CLK_PER after a write to SEGD2. Use with UPDATECTRL.AUTOLOAD
6	SEGD3WR	Starts synchronizing registers from CLK_BUS to CLK_PER after a write to SEGD3. Use with UPDATECTRL.AUTOLOAD
7	SEGD4WR	Starts synchronizing registers from CLK_BUS to CLK_PER after a write to SEGD4. Use with UPDATECTRL.AUTOLOAD
8	SEGD5WR	Starts synchronizing registers from CLK_BUS to CLK_PER after a write to SEGD5. Use with UPDATECTRL.AUTOLOAD
9	SEGD6WR	Starts synchronizing registers from CLK_BUS to CLK_PER after a write to SEGD6. Use with UPDATECTRL.AUTOLOAD
10	SEGD7WR	Starts synchronizing registers from CLK_BUS to CLK_PER after a write to SEGD7. Use with UPDATECTRL.AUTOLOAD

12:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	AUTOLOAD	0x0	RW	**Auto Load** Auto Load will start the Load process after a write to the LOADADDR

Value	Mode	Description

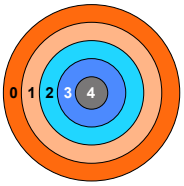
Bit	Name	Reset	Access	Description
	0	MANUAL		CLK_BUS register to CLK_PER register loads must be done manually with a write to CMD.LOAD.
	1	AUTO		CLK_BUS register to CLK_PER register loads will be started automatically after a write to the register in UPDATECTRL.LOADADDR is detected.
7:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

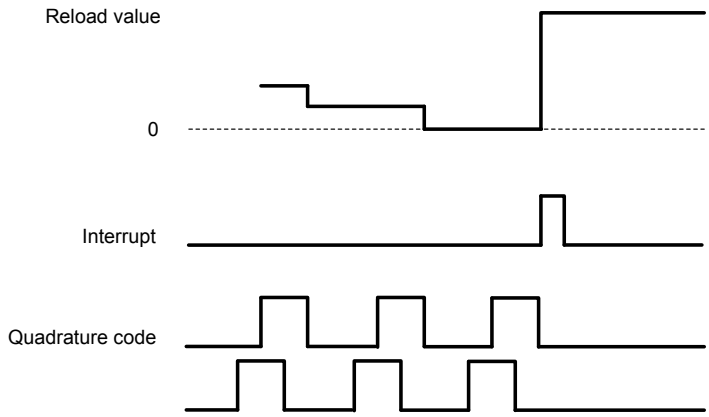
26.5.25 LCD_FRAMERATE - Frame Rate

Offset	Bit Position																														
0x0F0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0				
Access																												RW				
Name																												FRDIV				

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	FRDIV	0x0	RW	**Frame Rate Divider** Determines number of prescaled clocks per phase. Static has 2 phases, and octaplex has sixteen phases per frame.

27. PCNT - Pulse Counter





Quick Facts

What?

The Pulse Counter (PCNT) decodes incoming pulses. The module has a quadrature mode which may be used to decode the speed and direction of a mechanical shaft. PCNT can operate in EM0 down to EM3.

Why?

The PCNT generates an interrupt after a specific number of pulses (or rotations), eliminating the need for timing or I/O interrupts and CPU processing to measure pulse widths, etc.

How?

PCNT uses the EM23GRPACLK or may be externally clocked from a pin. The module incorporates a 16-bit up/down-counter to keep track of incoming pulses or rotations.

27.1 Introduction

The Pulse Counter (PCNT) can be used for counting incoming pulses on a single input or to decode quadrature encoded inputs in EM0 down to EM3. It can run from the internal EM23GRPACLK clock source while counting pulses on the PCNTn_S0IN pin. Alternatively, the PCNTn_S0IN pin or a PRS signal may be used as an external clock source that runs the PCNT counter and register access.

27.2 Features

- 16-bit counter with reload register
- Auxiliary counter for counting a single direction
- Single input oversampling up/down counter mode
- Externally clocked single input pulse up/down counter mode
- Quadrature decoder modes
  - Externally clocked quadrature decoder 1X mode
  - Oversampling quadrature decoder 1X, 2X and 4X modes
- Interrupt on counter underflow and overflow
- Interrupt when a direction change is detected (quadrature decoder mode only)
- Optional pulse width filter
- Optional input inversion/edge detect select
- Optional inputs from PRS

27.3 Functional Description

An overview of the PCNT module is shown in Figure 27.1 PCNT Overview on page 1102.

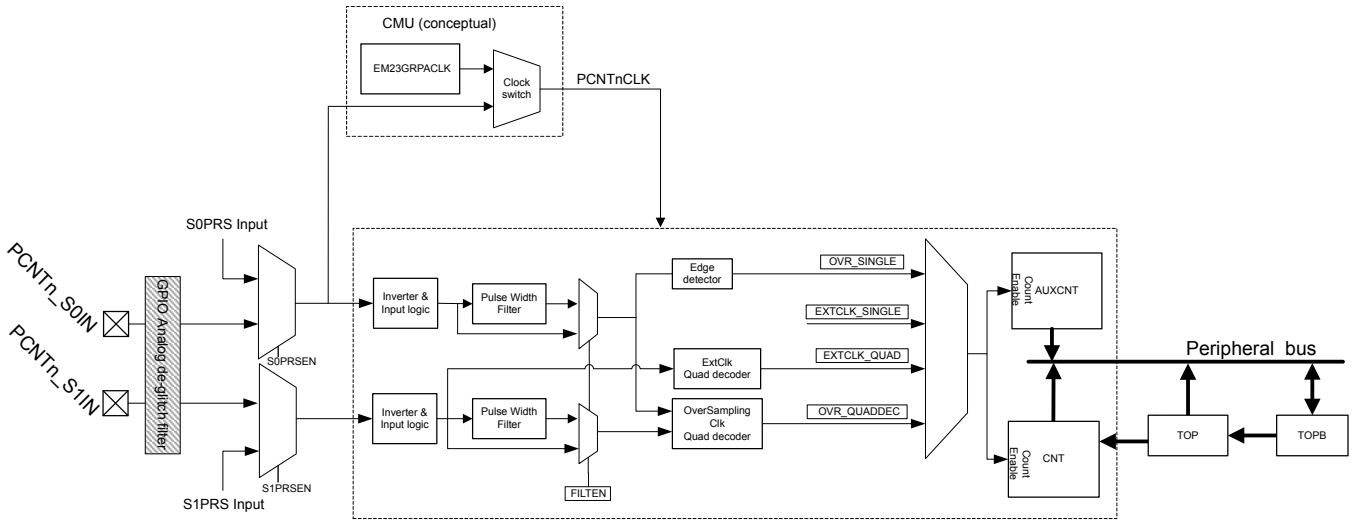


Figure 27.1. PCNT Overview

27.3.1 Pulse Counter Modes

The pulse counter can operate in single input oversampling mode (OVSSINGLE), externally clocked single input counter mode (EXTCLKSINGLE), externally clocked quadrature decoder mode (EXTCLKQUAD) and oversampling quadrature decoder modes(OVSQUAD1X, OVSQUAD2X and OVSQUAD4X). The following sections describe operation of each of these modes and how they are enabled.

27.3.1.1 Single Input Oversampling Mode

This mode is selected by writing OVSSINGLE to the MODE field in the PCNTn_CFG register. The STARTCNT bit in PCNTn_CMD is used to start the counter, and STOPCNT can be used to stop the counter. The EM23GRPACLK clock source to the pulse counter is selected by setting CLKSEL in the CMU_PCNT0CLKCTRL register to EM23GRPACLK. In this mode the maximum input toggle frequency should be 2 times slower than the frequency of the selected EM23GRPACLK clock source.

The optional pulse width filter is enabled by setting the FILTEN bit in the PCNTn_CFG register. Additionally, the PCNTn_S0IN input may be inverted, so that falling edges are counted, by setting the EDGE bit in the PCNTn_CTRL register.

If S1CDIR in the PCNTn_CTRL register is cleared, PCNTn_S0IN is the only observed input in this mode. The PCNTn_S0IN input is sampled by the PCNTnCLK and the number of detected positive or negative edges on PCNTn_S0IN appears in PCNTn_CNT. By default the counter will count up, but the counter may be configured to count down by setting the CNTDIR bit in PCNTn_CTRL.

The counting direction can also be controlled externally in this mode, by setting S1CDIR. This will make the input value on PCNTn_S1IN decide the direction counted for each PCNTn_S0IN edge. When PCNTn_S1IN is high, the count is done according to CNTDIR in PCNTn_CTRL. When PCNTn_S1IN is low, the count direction is opposite.

27.3.1.2 Externally Clocked Single Input Counter Mode

This mode is enabled by writing EXTCLKSINGLE to the MODE field in the PCNTn_CFG register. The STARTCNT bit in PCNTn_CMD is used to start the counter, and STOPCNT can be used to stop the counter. The external pin clock source is selected by setting CLKSEL in the CMU_PCNT0CLKCTRL register to PCNTS0.

Positive edges on PCNTn_S0IN are used to clock the counter. Similar to the oversampled mode, PCNTn_S1IN is used to determine the count direction if S1CDIR is set. If not, CNTDIR in PCNTn_CTRL solely defines count direction.

The digital pulse width filter is not available in this mode. The analog de-glitch filter in the GPIO pads is capable of removing some unwanted noise. However, this mode may be susceptible to spikes and unintended pulses from devices such as mechanical switches, and is therefore most suited to take input from electronic sensors etc. that generate single wire pulses.

27.3.1.3 Quadrature Decoder Modes

Two different types of quadrature decoding are supported in the pulse counter: the externally clocked (Asynchronous) quadrature decoding and the oversampling (Synchronous) quadrature decoding. The externally clocked mode supports 1X quadrature decoding whereas the oversampling mode supports 1X, 2X and 4X quadrature decoding. These modes are described in detail in 27.3.1.4 Externally Clocked Quadrature Decoder Mode and 27.3.1.5 Oversampling Quadrature Decoder Mode .

27.3.1.4 Externally Clocked Quadrature Decoder Mode

This mode is enabled by writing EXTCLKQUAD to the MODE field in PCNTn_CFG. The STARTCNT bit in PCNTn_CMD is used to start the counter, and STOPCNT can be used to stop the counter. The external pin clock source is selected by setting CLKSEL in the CMU_PCNT0CLKCTRL register to PCNTS0.

In this mode, both edges on PCNTn_S0IN pin are used to sample the PCNTn_S1IN pin, in order to decode the quadrature code. A quadrature coded signal contains information about the relative speed and direction of a rotating shaft as illustrated by Figure 27.2 PCNT Quadrature Coding on page 1104, hence the direction of the counter register PCNTn_CNT is controlled automatically.

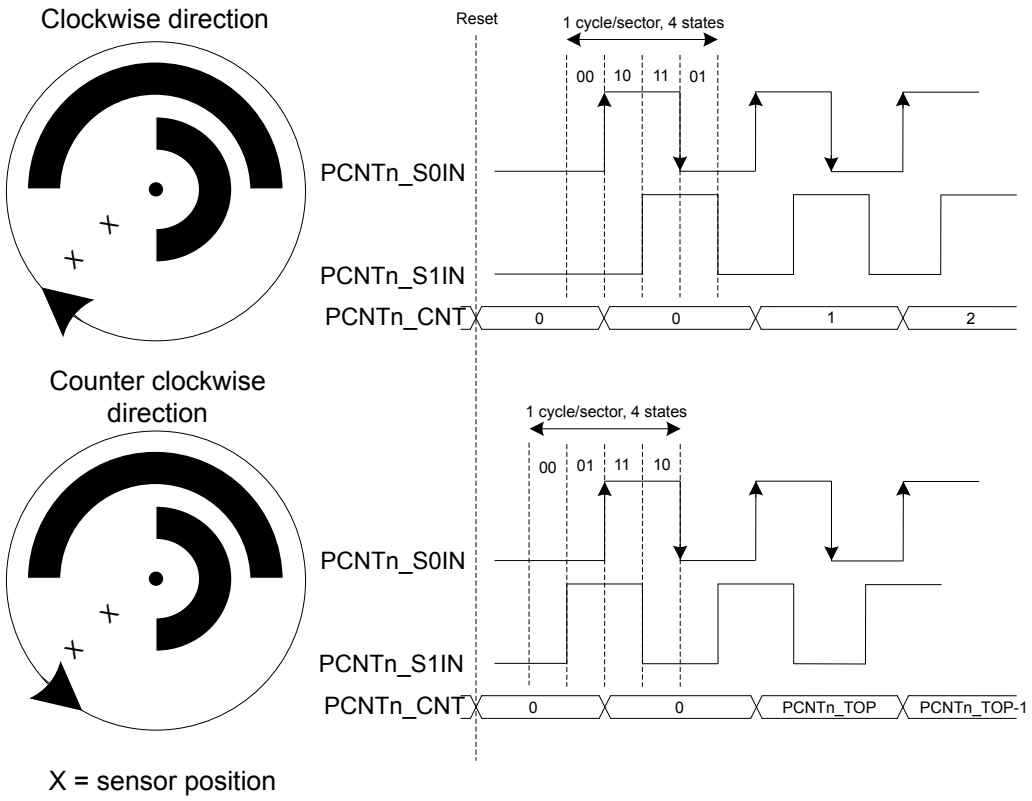


Figure 27.2. PCNT Quadrature Coding

If PCNTn_S0IN leads PCNTn_S1IN in phase, the direction is clockwise, and if it lags in phase the direction is counter-clockwise. Default behavior is illustrated by Figure 27.2 PCNT Quadrature Coding on page 1104.

The counter direction may be read from the DIR bit in the PCNTn_STATUS register. Additionally, the DIRCNG interrupt in the PCNTn_IF register is generated when a direction change is detected. When a change is detected, the DIR bit in the PCNTn_STATUS register must be read to determine the current new direction.

Note: The sector disc illustrated in the figure may be finer grained in some systems. Typically, they may generate 2-4 PCNTn_S0IN wave periods per 360° rotation.

The direction of the quadrature code and control of the counter is generated by the simple binary function outlined by Table 27.1 PCNT QUAD Mode Counter Control Function on page 1104. Note that this function also filters some invalid inputs that may occur when the shaft changes direction or temporarily toggles direction.

Table 27.1. PCNT QUAD Mode Counter Control Function

Inputs		Control/Status	
S1IN posedge	S1IN negedge	Count Enable	CNTDIR status bit
0	0	0	0

Inputs		Control/Status	
S1IN posedge	S1IN negedge	Count Enable	CNTDIR status bit
0	1	1	0
1	0	1	1
1	1	0	0
Note: PCNTn_S1IN is sampled on both edges of PCNTn_S0IN.			

27.3.1.5 Oversampling Quadrature Decoder Mode

There are three Oversampling Quadrature Decoder Modes supported: 1X , 2X and 4X. These modes are enabled by writing OVSQUAD1X, OVSQUAD2X and OVSQUAD4X, respectively, to the MODE field in PCNTn_CFG. The STARTCNT bit in PCNTn_CMD is used to start the counter, and STOPCNT can be used to stop the counter. The EM23GRPACLK clock source to the pulse counter must be selected by setting CLKSEL in the CMU_PCNT0CLKCTRL register to EM23GRPACLK.

The optional pulse width filter is enabled by setting the FILTEN bit in the PCNTn_CFG register. The filter applies to both inputs PCNTn_S0IN and PCNTn_S1IN. The filter length is configured by FILTLEN in PCNTn_OVSCTRL register.

Based on the modes selected, the decoder updates the counter on different events. In the OVSQUAD1X mode, the counter is updated on the rising edge of the PCNTn_S0IN input when counting up, and on the negedge of the PCNTn_S0IN input when counting down. In the OVSQUAD2X mode, the counter is updated on both edges of PCNTn_S0IN input. In the OVSQUAD4X mode the counter is updated on both edges of both inputs PCNTn_S0IN and PCNTn_S1IN. Table 27.2 PCNT OVSQUAD 1X, 2X and 4X Mode Counter Control Function on page 1106 outlines the increment or decrement of the counter based on the Quadrature Mode selected.

Note: The decoding behavior of OVSQUAD1X mode is slightly different compared to EXTCLKQUAD mode(also 1X mode). In the EXTCLKQUAD mode, the counter is updated only on the posedge of S0IN input. However, in the OVSQUAD1X mode, the counter is updated on the posedge of S0IN when counting up and on the negedge of S0IN when counting down.

Table 27.2. PCNT OVSQUAD 1X, 2X and 4X Mode Counter Control Function

Direction	Previous State		Next State		OVSQUAD MODE		
	S1IN	S0IN	S1IN	S0IN	1X	2X	4X
Clockwise	0	0	0	1	+1	+1	+1
	0	1	1	1			+1
	1	1	1	0		+1	+1
	1	0	0	0			+1
Counter Clockwise	1	0	1	1		-1	-1
	1	1	0	1			-1
	0	1	0	0	-1	-1	-1
	0	0	1	0			-1

Figure 27.3 PCNT State Transitions for Different Oversampling Quadrature Decoder Modes on page 1107 illustrates the different states of the quadrature input and the state transitions that updates the counter for the different modes. Each cycle of the input states results in 1 update, 2 updates and 4 updates of the counter for OVSQUAD1X, OVSQUAD2X and OVSQUAD4X modes respectively.

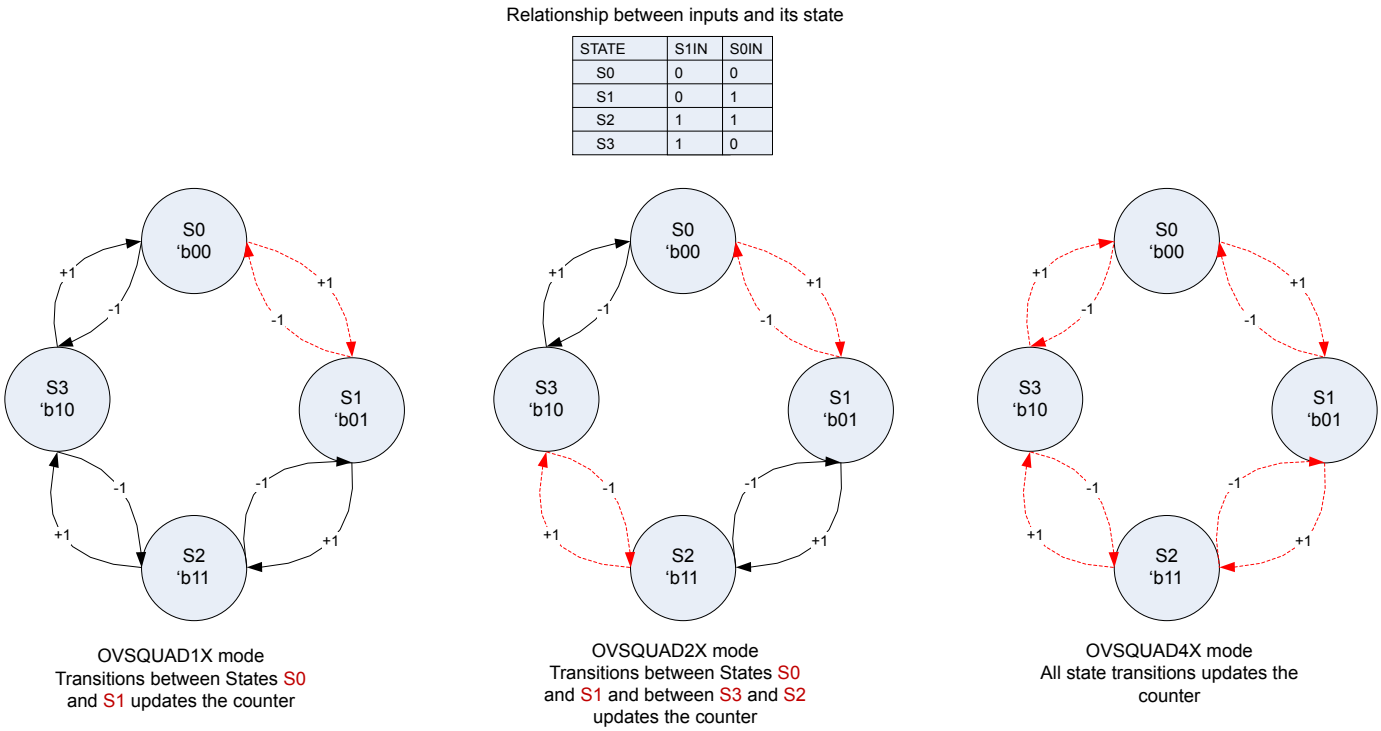


Figure 27.3. PCNT State Transitions for Different Oversampling Quadrature Decoder Modes

The counter direction can be read from the DIR bit in PCNTn_STATUS register. Additionally, the DIRCNG interrupt in the PCNTn_IF is generated when the direction change is detected. When a change is detected, the DIR bit in the PCNTn_STATUS register must be read to determine the new direction.

In the oversampling quadrature decoder modes, the maximum input toggle frequency supported is PCNTnCLK / 4. For frequencies above PCNTnCLK / 4, incorrect decoding occurs. The different decoding modes and the counter updates are further illustrated by Figure 27.4 PCNT Oversampling Quadrature Decoder 1X Mode on page 1107, Figure 27.5 PCNT Oversampling Quadrature Decoder 2X Mode on page 1108 and Figure 27.6 PCNT Oversampling Quadrature Decoder 4X Mode on page 1108.

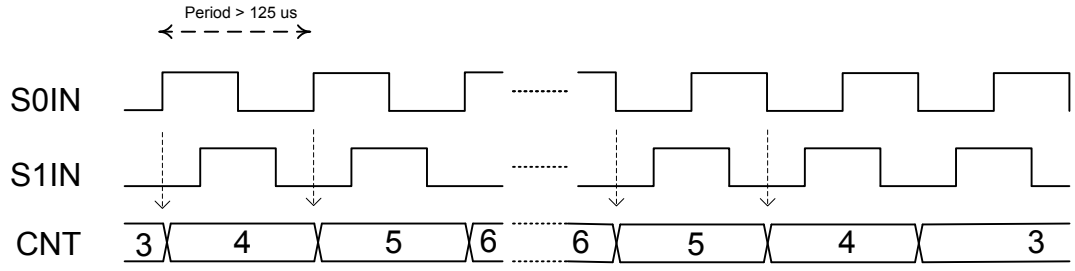


Figure 27.4. PCNT Oversampling Quadrature Decoder 1X Mode

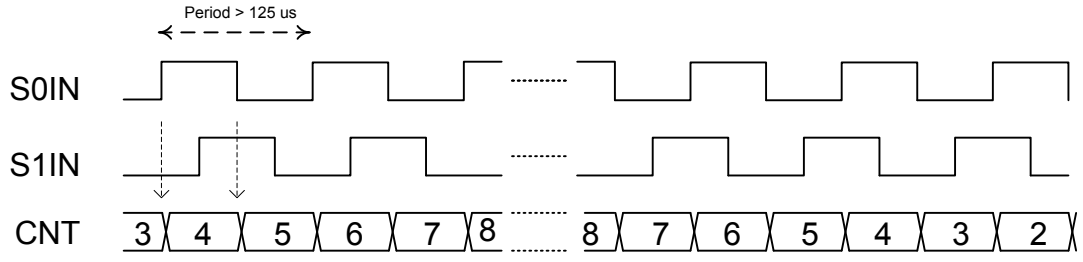


Figure 27.5. PCNT Oversampling Quadrature Decoder 2X Mode

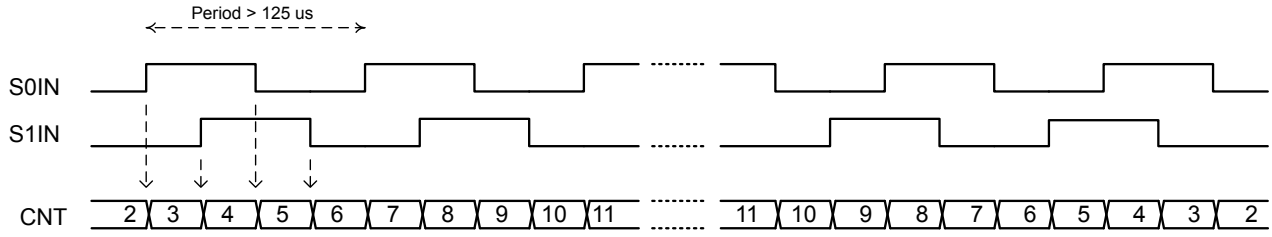


Figure 27.6. PCNT Oversampling Quadrature Decoder 4X Mode

The above modes, by default, are prone to flutter effects in the inputs PCNTn_S0IN and PCNTn_S1IN. When this occurs, the counter changes directions rapidly causing DIRCNG interrupts and unnecessarily waking the core. To prevent this, set FLUTTERRM in the PCNTn_OVSCTRL register. When enabled, flutter is removed, thus preventing unnecessary wakeup of the core. The flutter removal logic works by preventing update of the counter value if the wheel keeps changing direction as a result of flutter. The counter is only updated if the current and previous state transition of the rotation are in the same direction. These state transitions are quadrature decoder mode specific. The highlighted state transitions in Figure 27.3 PCNT State Transitions for Different Oversampling Quadrature Decoder Modes on page 1107 are the ones considered for the different quadrature decoder modes. Figure 27.7 PCNT Oversampling Quadrature Decoder with Flutter Removal on page 1108 shows how the counter is updated for the different quadrature decoder modes with flutter removal FLUTTERRM enabled in PCNTn_OVSCTRL.

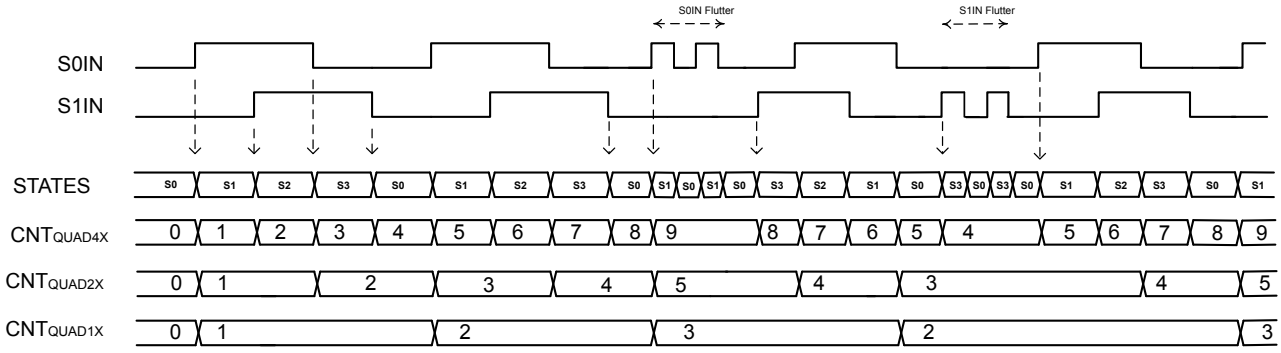


Figure 27.7. PCNT Oversampling Quadrature Decoder with Flutter Removal

27.3.2 Hysteresis

By default the pulse counter wraps to 0 when passing the configured top value, and wraps to the top value when counting down from 0. On these events, a system will likely want to wake up to store and track the overflow count. This is fine if the pulse counter is tracking a monotonic value or a value that does not change directions frequently. In the latter scenario, if the counter changes directions around the overflow/underflow point, the system will have to wake up frequently to keep track of the rotations, resulting in higher current consumption.

To solve this, the pulse counter has a way of introducing hysteresis to the counter. When HYST in PCNTn_CFG is set, the pulse counter will always wrap to TOP/2 on underflows and overflows. This takes the counter away from the area where it might overflow or underflow, removing the problem. Figure 27.8 PCNT Hysteresis behavior of Counter on page 1109 illustrates the hysteresis behavior.

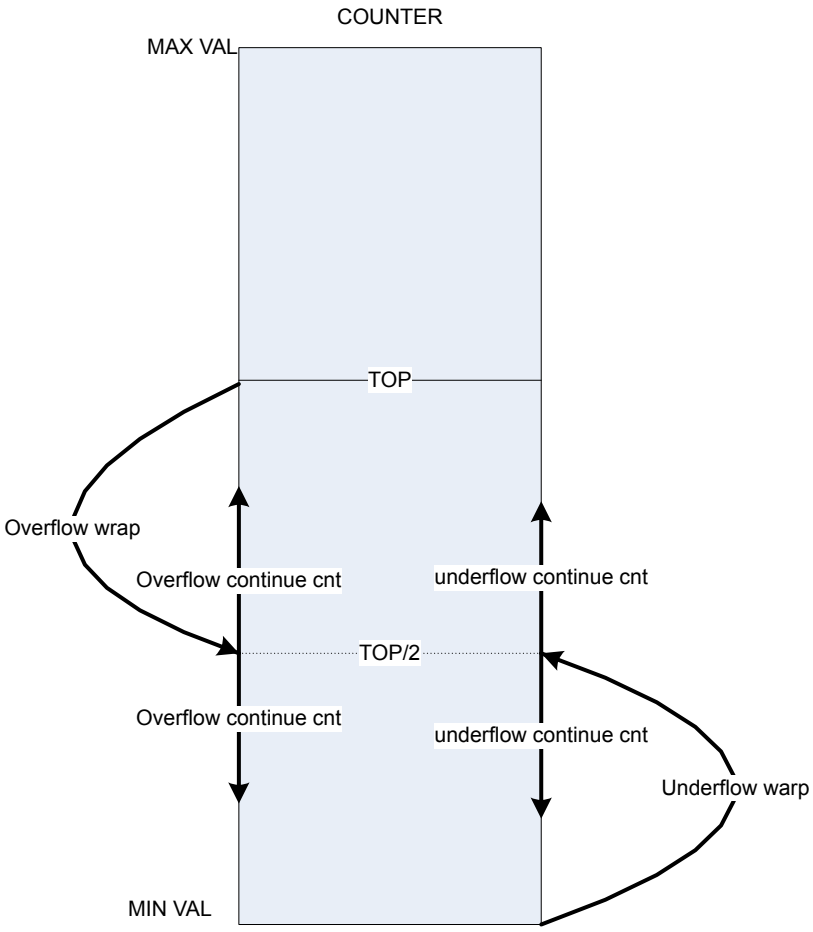


Figure 27.8. PCNT Hysteresis behavior of Counter

Given a starting value of 0 for the counter, the absolute count value when hysteresis is enabled can be calculated with the equations Figure 27.9 Absolute Position With Hysteresis and Even TOP Value on page 1109 or Figure 27.10 Absolute Position With Hysteresis and Odd TOP Value on page 1109, depending on whether the TOP value is even or odd.

$$CNT_{abs} = CNT - UF_{CNT} \times (TOP/2+1) + OF_{CNT} \times (TOP/2+1)$$

Figure 27.9. Absolute Position With Hysteresis and Even TOP Value

$$CNT_{abs} = CNT - UF_{CNT} \times (TOP/2+1) + OF_{CNT} \times (TOP/2+2)$$

Figure 27.10. Absolute Position With Hysteresis and Odd TOP Value

27.3.3 Auxiliary Counter

To be able to keep explicit track of counting in one direction in addition to the regular counter which counts both up and down, the auxiliary counter can be used. The pulse counter can, for instance, be configured to keep track of the absolute rotation of the wheel, while at the same time the auxiliary counter can keep track of how much the wheel has reversed.

The auxiliary counter is enabled by configuring AUXCNTEV in PCNTn_CTRL. It will always count up, but it can be configured whether it should count up on up-events, down-events or both, keeping track of rotation either way or general movement. The value of the auxiliary counter can be read from the PCNTn_AUXCNT register.

Overflows on the auxiliary counter happen when the auxiliary counter passes the top value of the pulse counter, configured in PCNTn_TOP. In that event, the AUXOF interrupt flag is set, and the auxiliary counter wraps to 0.

The auxiliary counter is started and stopped in a similar way to the main counter, using commands in the PCNTn_CMD register. The STARTAUXCNT bit in PCNTn_CMD is used to start the auxiliary counter, and STOPAUXCNT can be used to stop the auxiliary counter.

As the auxiliary counter, the main counter can be configured to count only on certain events. This is done through CNTEV in PCNTn_CTRL, and it is possible like for the auxiliary counter, to make the main counter count on only up and down events. The difference between the counters is that where the auxiliary counter will only count up, the main counter will count up or down depending on the direction of the count event.

27.3.4 Register Access

The counter-clock domain may be clocked externally. To update the counter-clock domain registers from software in this mode, 2-3 clock pulses on the external clock are needed to synchronize accesses to the externally clocked domain. Clock source switching is controlled from the registers in the CMU.

When the CORERST bit in the PCNTn_CMD register is set by software, the PCNT clock domain is synchronously reset and released two PCNT clock edges later. This synchronous reset restores the reset values in PCNTn_TOP, PCNTn_CNT and other core registers in the PCNT clock domain.

CNTRST works in a similar manner as CORERST, but only resets the counter, CNT. Note that the counter is also reset by CORERST.

AUXCNTRST works in a similar manner as CORERST, but only resets the auxiliary counter, PCNTn_AUXCNT. Note that the auxiliary counter is also reset by CORERST.

Note: PCNTn_CNT is a read-only register. When writing to PCNTn_TOP, make sure that the counter value, PCNTn_CNT, can not exceed the value written to PCNTn_TOP within two clock cycles.

Note: To ensure reset during SWRST, if the counter-clock domain is clocked externally, it is advisable to swith the counter-clock to the internal clock before performing the software reset. The clock pulses from an external source cannot be relied upon. By switching to an internal clock the software reset operation completion will be guaranteed.

27.3.5 Clock Sources

The pulse counter may be clocked from two possible clock sources: EM23GRPACLK or an external clock. The clock selection is configured by setting the CLKSEL field in the CMU_PCNTnCLKCTRL register. The default clock source is the EM23GRPACLK.

This PCNT module may also use PCNTn_S0IN as an external clock to clock the counter (EXTCLKSINGLE mode) and to sample PCNTn_S1IN (EXTCLKQUAD mode). Setup, hold and max frequency constraints for PCNTn_S0IN and PCNTn_S1IN for these modes are specified in the device data sheet.

Note: If changing PCNT to an external clock source, either through the PCNTn_S0IN or S0PRS input, the CMU clock switch will need at least 2 external clocks to sync over to that new clock source. Register will not load and counting will not begin until there are few external clocks.

Note: Switching external clock source to the S0PRS input can cause a clock glitch. So S0PRSEN in PCNTn_CFG should be configured before configuring the CLKSEL field in CMU_PCNTnCLKCTRL.

27.3.6 Input Filter

An optional pulse width filter is available in OVSSINGLE and OVSQUAD modes, when EM23GRPACLK is selected as a clock source for the Pulse Counter. The filter is enabled by writing 1 to the FILTEN bit in the PCNTn_CTRL register. When enabled, the high and low periods of PCNTn_S0IN and PCNTn_S1IN must be stable for a programmable number of consecutive clock cycles before the edge is passed to the edge detector. The filter length should be programmed in FILTLEN field of the PCNTn_OVSCTRL register.

The filter length is given by Figure 27.11 PCNT Input Filter Length Equation on page 1111:

Filter length = (FILTLEN + 5) EM23GRPACLK cycles

Figure 27.11. PCNT Input Filter Length Equation

The maximum filter length configured is 260 EM23GRPACLK cycles.

In EXTCLKSINGLE and EXTCLKQUAD mode, there is no digital pulse width filter available.

27.3.7 Edge Polarity

The edge polarity can be set by configuring the EDGE bit in the PCNTn_CTRL register. When this bit is cleared, the pulse counter counts positive edges of PCNTn_S0IN input. When this bit is set, the pulse counter counts negative edges in OVSSINGLE mode. Also, when the EDGE bit is set in the OVSSINGLE and EXTCLKSINGLE modes, the PCNTn_S1IN input is inverted. In OVSQUAD 1X-4X modes the EDGE bit inverts both inputs.

Note: The EDGE bit in PCNTn_CTRL has no effect in EXTCLKQUAD mode.

27.3.8 PRS and PCNTn_S0IN,PCNTn_S1IN Inputs

It is possible to receive input from a PRS signal on both PCNTn_S0IN (or PCNTn_S1IN) by setting S0PRSEN (or S1PRSEN) in PCNTn_CFG. Routing of PRS channels to the inputs is performed using the PRS_CONSUMER_PCNTn_S0IN and PRS_CONSUMER_PCNTn_S1IN registers.

In the Oversampling quadrature decoder modes, the input frequency should be less than 4 times slower than the sampling clock PCNTnCLK to ensure correct functionality.

In the Single Input Oversampling Mode, the input toggle frequency should be 2 times slower than the sampling clock PCNTnCLK to ensure correct functionality.

In the externally clocked modes (where S0IN or S0PRS is used as PCNTnCLK clock source), the input frequency should be less than 1 MHz to ensure correct functionality.

The PCNT module generates two PRS signals, the PCNTn UFOF signal and the PCNTn DIR signal. The PCNTn UFOF signal is generated when the counter overflows or underflows. The PCNTn DIR signal is a level, and indicates the current direction of count of counter CNT.

27.3.9 Interrupts

The interrupt generated by PCNT uses the PCNTn_INT interrupt vector. Software must read the PCNTn_IF register to determine which module interrupt that generated the vector invocation.

27.3.9.1 Underflow and Overflow Interrupts

The underflow interrupt flag (UF) is set when the counter counts down from 0 (i.e. when the value of the counter is 0 and a new pulse is received). The PCNTn_CNT register is loaded with the PCNTn_TOP value after this event.

The overflow interrupt flag (OF) is set when the counter counts up from the PCNTn_TOP (reload) value (i.e. if PCNTn_CNT = PCNTn_TOP and a new pulse is received). The PCNTn_CNT register is loaded with the value 0 after this event.

27.3.9.2 Direction Change Interrupt

The PCNTn_PCNT module sets the DIRCNG interrupt flag (PCNTn_IF register) for EXTCLKQUAD and OVSQUAD1X-4X modes when the direction of the quadrature code changes. The behavior of this interrupt in the EXTCLKQUAD mode is illustrated by Figure 27.12 PCNT Direction Change Interrupt (DIRCNG) Generation on page 1112.

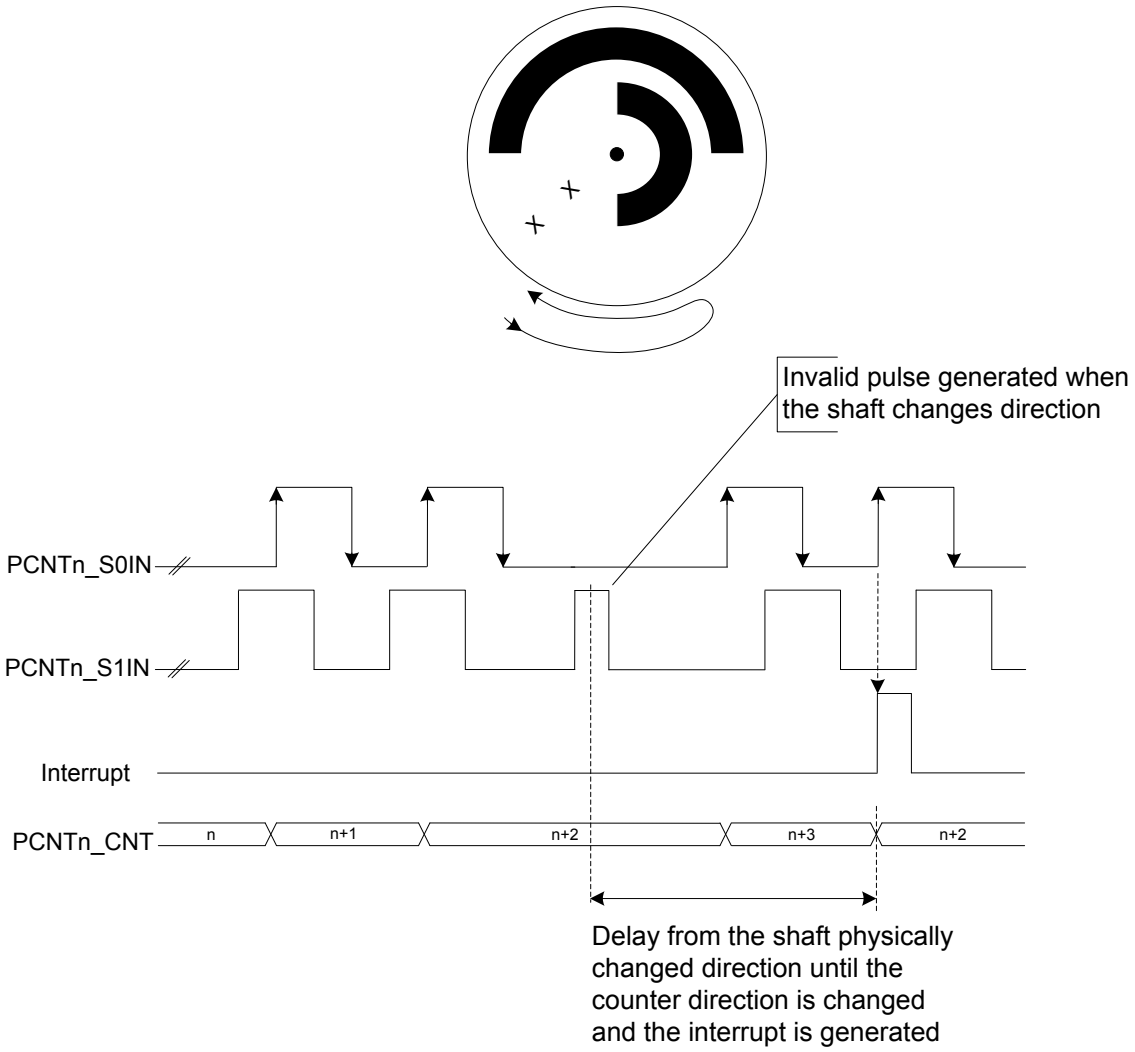


Figure 27.12. PCNT Direction Change Interrupt (DIRCNG) Generation

27.4 PCNT Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	PCNT_IPVERSION	R	IP Version ID
0x004	PCNT_EN	RW ENABLE	Module Enable Register
0x008	PCNT_SWRST	RW SWRST	Software Reset Register
0x00C	PCNT_CFG	RW CONFIG	Configuration Register
0x010	PCNT_CTRL	RW LFSYNC	Control Register
0x014	PCNT_CMD	W LFSYNC	Command Register
0x018	PCNT_STATUS	RH	Status Register
0x01C	PCNT_IF	RWH INTFLAG	Interrupt Flag Register
0x020	PCNT_IEN	RW	Interrupt Enable Register
0x024	PCNT_CNT	RH	Counter Value Register
0x028	PCNT_AUXCNT	RH	Auxiliary Counter Value Register
0x02C	PCNT_TOP	RWH LFSYNC	Top Value Register
0x030	PCNT_TOPB	RW LFSYNC	Counter Top Value Buffer Register
0x034	PCNT_OVSCTRL	RW LFSYNC	Oversampling Control Register
0x038	PCNT_SYNCBUSY	RH	Synchronization Busy Register
0x03C	PCNT_LOCK	W	Configuration Lock Register
0x1000	PCNT_IPVERSION_SET	R	IP Version ID
0x1004	PCNT_EN_SET	RW ENABLE	Module Enable Register
0x1008	PCNT_SWRST_SET	RW SWRST	Software Reset Register
0x100C	PCNT_CFG_SET	RW CONFIG	Configuration Register
0x1010	PCNT_CTRL_SET	RW LFSYNC	Control Register
0x1014	PCNT_CMD_SET	W LFSYNC	Command Register
0x1018	PCNT_STATUS_SET	RH	Status Register
0x101C	PCNT_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1020	PCNT_IEN_SET	RW	Interrupt Enable Register
0x1024	PCNT_CNT_SET	RH	Counter Value Register
0x1028	PCNT_AUXCNT_SET	RH	Auxiliary Counter Value Register
0x102C	PCNT_TOP_SET	RWH LFSYNC	Top Value Register
0x1030	PCNT_TOPB_SET	RW LFSYNC	Counter Top Value Buffer Register
0x1034	PCNT_OVSCTRL_SET	RW LFSYNC	Oversampling Control Register
0x1038	PCNT_SYNCBUSY_SET	RH	Synchronization Busy Register
0x103C	PCNT_LOCK_SET	W	Configuration Lock Register
0x2000	PCNT_IPVERSION_CLR	R	IP Version ID
0x2004	PCNT_EN_CLR	RW ENABLE	Module Enable Register
0x2008	PCNT_SWRST_CLR	RW SWRST	Software Reset Register

Offset	Name	Type	Description
0x200C	PCNT_CFG_CLR	RW CONFIG	Configuration Register
0x2010	PCNT_CTRL_CLR	RW LFSYNC	Control Register
0x2014	PCNT_CMD_CLR	W LFSYNC	Command Register
0x2018	PCNT_STATUS_CLR	RH	Status Register
0x201C	PCNT_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2020	PCNT_IEN_CLR	RW	Interrupt Enable Register
0x2024	PCNT_CNT_CLR	RH	Counter Value Register
0x2028	PCNT_AUXCNT_CLR	RH	Auxiliary Counter Value Register
0x202C	PCNT_TOP_CLR	RWH LFSYNC	Top Value Register
0x2030	PCNT_TOPB_CLR	RW LFSYNC	Counter Top Value Buffer Register
0x2034	PCNT_OVSCTRL_CLR	RW LFSYNC	Oversampling Control Register
0x2038	PCNT_SYNCBUSY_CLR	RH	Synchronization Busy Register
0x203C	PCNT_LOCK_CLR	W	Configuration Lock Register
0x3000	PCNT_IPVERSION_TGL	R	IP Version ID
0x3004	PCNT_EN_TGL	RW ENABLE	Module Enable Register
0x3008	PCNT_SWRST_TGL	RW SWRST	Software Reset Register
0x300C	PCNT_CFG_TGL	RW CONFIG	Configuration Register
0x3010	PCNT_CTRL_TGL	RW LFSYNC	Control Register
0x3014	PCNT_CMD_TGL	W LFSYNC	Command Register
0x3018	PCNT_STATUS_TGL	RH	Status Register
0x301C	PCNT_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3020	PCNT_IEN_TGL	RW	Interrupt Enable Register
0x3024	PCNT_CNT_TGL	RH	Counter Value Register
0x3028	PCNT_AUXCNT_TGL	RH	Auxiliary Counter Value Register
0x302C	PCNT_TOP_TGL	RWH LFSYNC	Top Value Register
0x3030	PCNT_TOPB_TGL	RW LFSYNC	Counter Top Value Buffer Register
0x3034	PCNT_OVSCTRL_TGL	RW LFSYNC	Oversampling Control Register
0x3038	PCNT_SYNCBUSY_TGL	RH	Synchronization Busy Register
0x303C	PCNT_LOCK_TGL	W	Configuration Lock Register

27.5 PCNT Register Description

27.5.1 PCNT_IPVERSION - IP Version ID

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP VERSION**
	Gives access to the IP VERSION of PCNT			

27.5.2 PCNT_EN - Module Enable Register

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status**
	When EN is cleared, DISABLING status is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except the INTFLAG register.			
0	EN	0x0	RW	**PCNT Module Enable**
	Enable the PCNT module. When EN is cleared(disablement), it halts module operation immediately, and initialize the core domain such that when the is re-enabled, it starts cleanly.			

27.5.3 PCNT_SWRST - Software Reset Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	W
Name																															RESETTING	SWRST

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESETTING	0x0	R	**Software reset busy status** When SWRST command is issued, resetting logic sets RESETTING status immediately, and later it is cleared when reset process finishes.
0	SWRST	0x0	W	**Software reset command** A software reset command field resets the module back to the initial condition, similar to a power on reset condition

27.5.4 PCNT_CFG - Configuration Register

Offset	Bit Position																															
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0	0x0		0x0	0x0	0x0		0x0		
Access																							RW	RW		RW	RW	RW		RW		
Name																							S1PRSEN	S0PRSEN		HYST	FILTEN	DEBUGHALT		MODE		

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	S1PRSEN	0x0	RW	**S1IN PRS Enable**
	When set, the PRS channel is selected as input to S1IN.			
8	S0PRSEN	0x0	RW	**S0IN PRS Enable**
	When set, the PRS channel is selected as input to S0IN.			
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	HYST	0x0	RW	**Enable Hysteresis**
	When hysteresis is enabled, the PCNT will always overflow and underflow to TOP/2.			
5	FILTEN	0x0	RW	**Enable Digital Pulse Width Filter**
	The filter passes all high and low periods that are at least (FILTLEN+5) clock cycles wide. This filter is only available in OVSSINGLE,OVSQUAD1X-4X modes.			
4	DEBUGHALT	0x0	RW	**Debug Mode Halt Enable**
	Set to halt the PCNT in debug mode only in OVSSINGLE and OVSQUAD modes. When in EXTCLKSINGLE or EXTCLKQUAD modes, DEBUGHALT does not halt the Pulse Counter.			

Value	Mode	Description
0	DISABLE	PCNT is running in debug mode.
1	ENABLE	PCNT is frozen in debug mode.

3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	MODE	0x0	RW	**Mode Select**
	Selects the mode of operation. The corresponding clock source must be selected from the CMU.			

Value	Mode	Description
0	OVSSINGLE	Single input EM23GRPACLK oversampling mode (available in EM0-EM3).
1	EXTCLKSINGLE	Externally clocked single input counter mode (available in EM0-EM3).
2	EXTCLKQUAD	Externally clocked quadrature decoder mode (available in EM0-EM3).

Bit	Name	Reset	Access	Description
	3	OVSQUAD1X		EM23GRPACLK oversampling quadrature decoder 1X mode (available in EM0-EM3).
	4	OVSQUAD2X		EM23GRPACLK oversampling quadrature decoder 2X mode (available in EM0-EM3).
	5	OVSQUAD4X		EM23GRPACLK oversampling quadrature decoder 4X mode (available in EM0-EM3).

27.5.5 PCNT_CTRL - Control Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0		0x0			0x0	0x0	0x0
Access																									RW		RW			RW	RW	RW
Name																									AUXCNTEV		CNTEV			EDGE	CNTDIR	S1CDIR

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:6	AUXCNTEV	0x0	RW	**Controls When the Aux Counter Counts**
	Selects whether the auxiliary counter responds to up-count events, down-count events or both			
	Value	Mode		Description
	0	BOTH		Counts up on both up-count and down-count events.
	1	UP		Counts up on up-count events.
	2	DOWN		Counts up on down-count events.
5:4	CNTEV	0x0	RW	**Controls When the Counter Counts**
	Selects whether the regular counter responds to up-count events, down-count events or both			
	Value	Mode		Description
	0	BOTH		Counts up on up-count and down on down-count events.
	1	UP		Only counts up on up-count events.
	2	DOWN		Only counts down on down-count events.
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	EDGE	0x0	RW	**Edge Select**
	Determines the polarity of the incoming edges. This bit used only in OVSSINGLE, EXTCLKSINGLE and OVSQUAD1X-4X modes.			
	Value	Mode		Description
	0	POS		Positive edges on the PCNTn_S0IN inputs are counted in OVSSINGLE mode. Does not invert PCNTn_S1IN input in OVSSINGLE and EXTCLKSINGLE modes
	1	NEG		Negative edges on the PCNTn_S0IN inputs are counted in OVSSINGLE mode. Inverts the PCNTn_S1IN input in OVSSINGLE and EXTCLKSINGLE modes
1	CNTDIR	0x0	RW	**Non-Quadrature Mode Counter Direction Co**
	The direction of the counter must be set in the OVSSINGLE and EXTCLKSINGLE modes. This bit is ignored in EXTCLKQUAD mode as the direction is automatically detected.			

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	UP		Up counter mode.
	1	DOWN		Down counter mode.
0	S1CDIR	0x0	RW	**Count Direction Determined By S1**
	Allows S1 give the direction of counting when in the OVSSINGLE or EXTCLKSINGLE modes. When S1 is high, the count direction is given by CNTDIR. When S1 is low, the count direction is the opposite given by CNTDIR			

27.5.6 PCNT_CMD - Command Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0	0x0	0x0	0x0				0x0		0x0	0x0	0x0
Access																					W	W	W	W				W(nB)		W(nB)	W(nB)	W(nB)
Name																					STOPAUXCNT	STOPCNT	STARTAUXCNT	STARTCNT				LCNTIM		AUXCNTRST	CNTRST	CORERST

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	STOPAUXCNT	0x0	W	**Stop Aux Counter**
	Write a 1 to stop the aux counter			
10	STOPCNT	0x0	W	**Stop Main Counter**
	Write a 1 to stop the main counter			
9	STARTAUXCNT	0x0	W	**Start Aux Counter**
	Write a 1 to start the aux counter			
8	STARTCNT	0x0	W	**Start Main Counter**
	Write a 1 to start the main counter			
7:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	LCNTIM	0x0	W(nB)	**Load CNT Immediately**
	Load PCNTn_TOP into PCNTn_CNT on the next counter clock cycle.			
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	AUXCNTRST	0x0	W(nB)	**AUXCNT Reset**
	The auxiliary counter, AUXCNT, is synchronously reset when this bit is set. This action clears the auxiliary counter to its reset value			
1	CNTRST	0x0	W(nB)	**CNT Reset**
	The counter, CNT, is synchronously reset when this bit is set. This action clears the counter to its reset value			
0	CORERST	0x0	W(nB)	**PCNT Clock Domain Reset**
	The PCNT clock domain logic is synchronously reset when this bit is set. This action clears the all the PCNT logic and registers to their reset value			

27.5.7 PCNT_STATUS - Status Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												R	R	R	R	R
Name																												AUXCNTRUNNING	CNTRUNNING	PCNTLOCKSTATUS	TOPBV	DIR

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	AUXCNTRUNNING	0x0	R	**Aux Counter running status**
	Indicates the current status of PCNT auxiliary counter running. The counter status will continue to indicate as Running even when it is halted during debug assertion.			
3	CNTRUNNING	0x0	R	**Main Counter running status**
	Indicates the current status of PCNT main counter running. The counter status will continue to indicate as Running even when it is halted during debug assertion.			
2	PCNTLOCKSTATUS	0x0	R	**Lock Status**
	Indicates the current status of PCNT Lock			

Value	Mode	Description
0	UNLOCKED	PCNT registers are unlocked
1	LOCKED	PCNT registers are locked

Bit	Name	Reset	Access	Description
1	TOPBV	0x0	R	**TOP Buffer Valid**
	This indicates that PCNTn_TOPB contains valid data that has not been written to PCNTn_TOP. This bit is also cleared when PCNTn_TOP is written.			
0	DIR	0x0	R	**Current Counter Direction**
	Current direction status of the counter. This bit is valid in EXTCLKQUAD mode only.			

Value	Mode	Description
0	UP	Up counter mode (clockwise in EXTCLKQUAD mode with the EDGE bit in PCNTn_CTRL set to 0).
1	DOWN	Down counter mode.

27.5.8 PCNT_IF - Interrupt Flag Register

Offset	Bit Position																														
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												OQSTERR	AUXOF	DIRCNG	OF	UF

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	OQSTERR	0x0	RW	**Oversampling Quad State Err Int Flag**
	Set in the Oversampling Quadrature Mode when incorrect state transition occurs			
3	AUXOF	0x0	RW	**Auxiliary Overflow Interrupt Read Flag**
	Set when an Auxiliary CNT overflow occurs			
2	DIRCNG	0x0	RW	**Direction Change Detect Interrupt Flag**
	Set when the count direction changes. Set in EXTCLKQUAD mode only.			
1	OF	0x0	RW	**Overflow Interrupt Read Flag**
	Set when a CNT overflow occurs			
0	UF	0x0	RW	**Underflow Interrupt Read Flag**
	Set when a CNT underflow occurs			

27.5.9 PCNT_IEN - Interrupt Enable Register

Offset	Bit Position																														
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												OQSTERR	AUXOF	DIRCNG	OF	UF

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	OQSTERR	0x0	RW	**Oversampling Quad State Err Int Flag**
	Set to enable the OQSTERRIF Interrupt			
3	AUXOF	0x0	RW	**Auxiliary Overflow Interrupt Read Flag**
	Set to enable the AUXOFIF Interrupt			
2	DIRCNG	0x0	RW	**Direction Change Detect Interrupt Flag**
	Set to enable the DIRCNGIF Interrupt			
1	OF	0x0	RW	**Overflow Interrupt Read Flag**
	Set to enable the OFIF Interrupt			
0	UF	0x0	RW	**Underflow Interrupt Read Flag**
	Set to enable the UFIF Interrupt			

27.5.10 PCNT_CNT - Counter Value Register

Offset	Bit Position																														
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	R															
Name																	CNT															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	CNT	0x0	R	**Counter Value**
	Gives read access to the counter.			

27.5.11 PCNT_AUXCNT - Auxiliary Counter Value Register

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	R															
Name																	AUXCNT															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	AUXCNT	0x0	R	**Auxiliary Counter Value**
	Gives read access to the auxiliary counter.			

27.5.12 PCNT_TOP - Top Value Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0xFF															
Access																	RW															
Name																	TOP															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	TOP	0xFF	RW	**Counter Top Value**
	When counting down, this value is reloaded into PCNTn_CNT when counting past 0. When counting up, 0 is written to the PCNTn_CNT register when counting past this value.			

27.5.13 PCNT_TOPB - Counter Top Value Buffer Register

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0xFF							
Access																									RW							
Name																									TOPB							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	TOPB	0xFF	RW	**Counter Top Buffer Register**
	These bits hold the TOP buffer value. Hardware updates the TOP value from TOPB whenever there is Underflow or Overflow event on main counter. The update to TOP wont happen for Overflow event if PCNTn_CTRL.CNTEV = BOTH			

27.5.14 PCNT_OVSCTRL - Oversampling Control Register

Offset	Bit Position																													
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																				0x0					0x0							
Access																				RW					RW							
Name																				FLUTTERRM					FILTLEN							

Bit	Name	Reset	Access	Description
31:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	FLUTTERRM	0x0	RW	**Flutter Remove**
	When set, removes flutter from Quaddecoder inputs S0IN and S1IN. Available only in OVSQUAD1X-4X modes			
11:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	FILTLEN	0x0	RW	**Configure Filter Length for Inputs S0IN**
	Used only in OVSINGLE,OVSQUAD1X-4X modes. To use this first enable FILTEN in PCNTn_CFG register. Filter length = (FILTLEN + 5) EM23GRPACLK cycles			

27.5.15 PCNT_SYNCBUSY - Synchronization Busy Register

Offset	Bit Position																													
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												R	R	R	R	R
Name																												OVSCTRL	TOPB	TOP	CMD	CTRL

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	OVSCTRL	0x0	R	**OVSCTRL Register Busy**
	Set when the value written to OVSCTRL register is being synchronized			
3	TOPB	0x0	R	**TOPB Register Busy**
	Set when the value written to TOPB register is being synchronized			
2	TOP	0x0	R	**TOP Register Busy**
	Set when the value written to TOP register is being synchronized			
1	CMD	0x0	R	**CMD Register Busy**
	Set when the value written to CMD register being synchronized			
0	CTRL	0x0	R	**CTRL Register Busy**
	Set when the value written to CTRL register being synchronized			

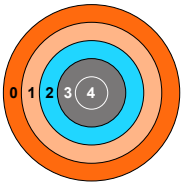
27.5.16 PCNT_LOCK - Configuration Lock Register

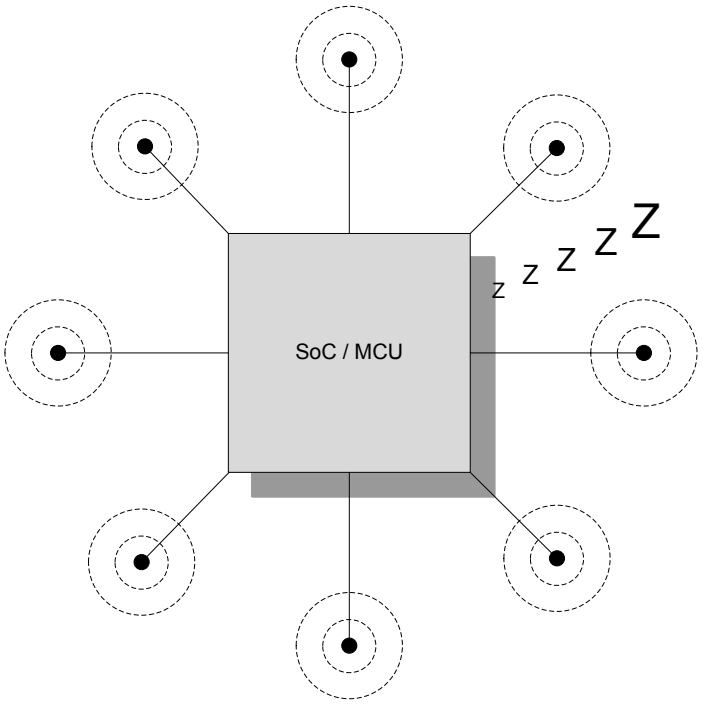
Offset	Bit Position																															
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W							
Name																									PCNTLOCKKEY							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	PCNTLOCKKEY	0x0	W	**Configuration Lock Key** Write any other value than the unlock code to lock PCNT_CFG, PCNT_EN, PCNT_SWRST, PCNT_CMD, PCNT_CTRL, PCNT_OVSCTRL, PCNT_CNT, PCNT_TOP and PCNT_TOPB registers from editing. Write the unlock code to unlock.

Value	Mode	Description
42976	UNLOCK	Write to unock PCNT lockable registers

28. LESENSE - Low Energy Sensor Interface





Quick Facts

What?

LESENSE is a low energy sensor interface capable of autonomously collecting and processing data from multiple sensors even when in EM2. Flexible configuration makes LESENSE a versatile sensor interface compatible with a wide range of sensors and measurement schemes.

Why?

Capability to autonomously monitor sensors allows the EFM32PG28 to reside in a low energy mode for long periods of time while keeping track of sensor status and sensor events.

How?

LESENSE is highly configurable and is capable of collecting data from a wide range of sensor types. Once the data is collected, the programmable state machine, LESENSE decoder, is capable of processing sensor data without CPU intervention. A large result buffer allows the chip to remain in EM2 for long periods of time while autonomously collecting data.

28.1 Introduction

LESENSE is a low energy sensor interface which utilizes on-chip peripherals to perform measurement of a configurable set of sensors. The results from sensor measurements can be processed by the LESENSE decoder, which is a configurable state machine with up to 32 states. The results can also be stored in a FIFO to be collected by CPU or DMA for further processing.

LESENSE operates from EM0 down to EM2, and can wake up the CPU on configurable events.

28.2 Features

- Up to 16 sensors
- Autonomous sensor monitoring in EM0, EM1, and EM2
- Highly configurable decoding of sensor results
- Interrupt on sensor events
- 16-deep FIFO for result storage
- Multiple evaluation modes minimize the need for software interaction
- Supports IADC and ACMP sampling and evaluation
- Support for multiple sensor types
  - LC sensors
  - General analog sensors

28.3 Functional Description

The LESENSE module is capable of controlling on-chip peripherals in order to perform monitoring of different sensors with little or no CPU intervention. LESENSE uses the analog comparators (ACMP) or IADC0 for measurement of sensor signals. LESENSE can also control VDAC0 channel 0 to generate accurate reference voltages. Figure 28.1 LESENSE Block Diagram on page 1130 shows an overview of the LESENSE module.

The LESENSE module consists of a sequencer, an evaluation block, a decoder, and a FIFO buffer.

- The sequencer handles interaction with other peripherals and controls timing of sensor measurements. It also includes a counter that can be used to count pulses on the ACMP output.
- The evaluation block is used to process the data collected by the sequencer.
- To autonomously analyze sensor results, the LESENSE decoder provides the ability to define a finite state machine with up to 32 states (64 arcs), with programmable actions upon state transitions. This allows the decoder to implement a wide range of decoding schemes, such as quadrature decoding.
- Measurement results are stored in a FIFO which enables the chip to remain in a low energy mode for longer periods of time while collecting sensor data.

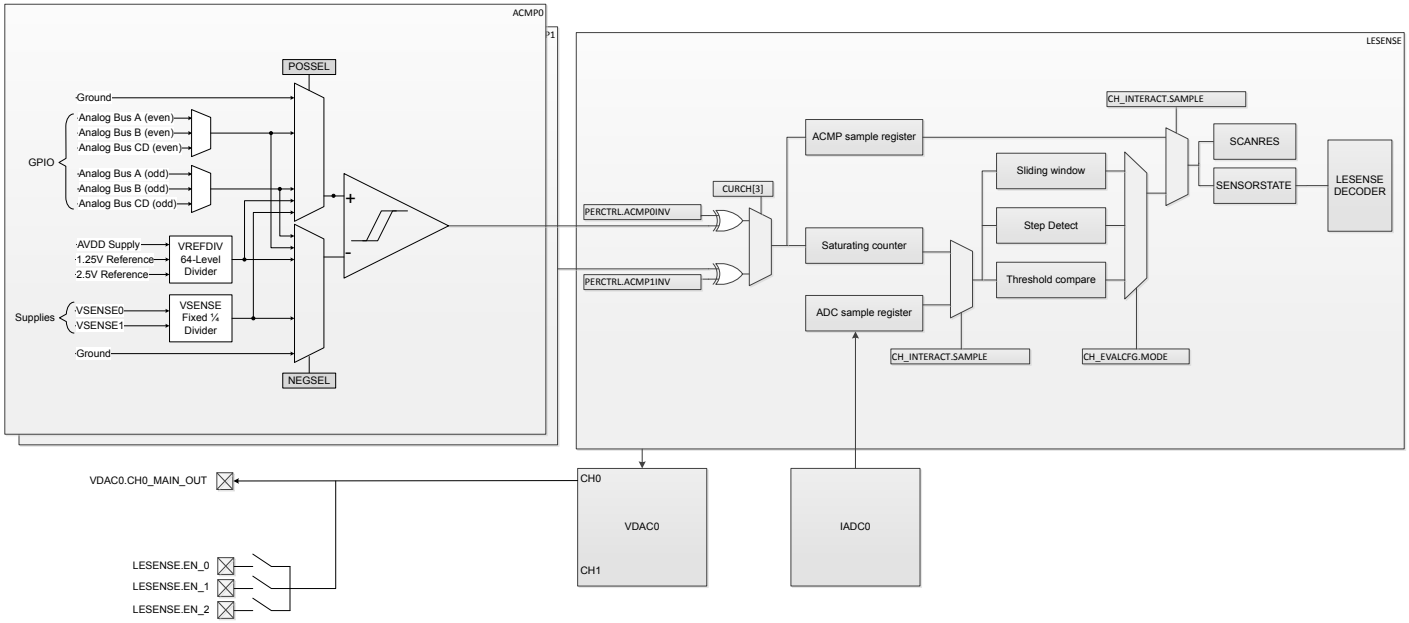


Figure 28.1. LESENSE Block Diagram

28.3.1 Interface Descriptions

ACMP interface

ACMPs are used to measure the sensors, and must be configured according to the application in order for LESENSE to work properly. Channels 0-7 trigger ACMP0, Channels 8-15 trigger ACMP1. Depending on the configuration in the ACMP0MODE and ACMP1MODE bitfields in LESENSE_PERCTRL, LESENSE can take control of the positive input mux and the voltage divider (VREFDIV) for the targeted ACMP. The remaining configuration of the analog comparators is done in the ACMP register interface.

When ACMPxMODE is set to MUX, the LESENSE module will take control of the positive input mux of the ACMP using the ACMP's external override interface. When used this way ACMP_INPUTCTRL_POSSEL should be configured to the base address of the GPIO port to be used. For example, if the desired channel is on Port A, POSSEL should be configured for EXTPA. The offset given by CHx_INTERACT.OFFSET determines which of the port I/O pins on PA to access. To connect to PB03 therefore, ACMP_INPUTCTRL_POSSEL would be set to EXTPB, and LESENSE_CHx_INTERACT.OFFSET to 0x03.

If ACMPxMODE in PERCTRL is set to MUXTHRES, LESENSE will also take control of the voltage divider (VREFDIV) in the ACMP. The threshold used are individual to each channel and is configured using the 6 LSBs of CHx_INTERACT_THRES.

In addition to all this, ACMPxINV in PERCTRL can be used to invert the result from the ACMP to the LESENSE.

IADC interface

LESENSE can be set up to trigger IADC0 conversions and use data from IADC0 to evaluate sensor status. In order to do this, the scan mode of the IADC0 has to be configured within the IADC0 module and the IADC SCAN trigger has to be setup as LESENSE. When the sample delay configured in CHx_TIMING_SAMPLEDLY has expired, LESENSE will initiate an ADC sample. The OFFSET programmed in CHx_INTERACT determines which channel specified in the IADC scanner is converted by the IADC.

VDAC interface

LESENSE is able to drive the DAC for generation of accurate reference voltages. The refresh rate of the DAC channels can be configured in DACCONVTRIG in PERCTRL. If DACCONVTRIG is set to CHANNELSTART, the LESENSE sends a conversion trigger to the DAC channels prior to each sensor measurement. If DACCONVTRIG is set to SCANSTART, the LESENSE sends a conversion trigger to the DAC prior to each scan. The conversion data used by the DAC is either taken from its own data registers in the DAC interface or from the THRES bit-field in the CHx_INTERACT register for the active LESENSE channel. DAC data used is configured in DACCH0DATA in PERCTRL.

None of the other DAC settings can be overriden from the LESENSE. For the DAC to be able to be used by the LESENSE, the DAC trigger source should be configured as LESENSE.

All LESENSE channels can send triggers to the DAC. However, only LESENSE channels 0,1,2 have the capability to switch the DAC output onto specific GPIOs. These channels have the been mapped to LESENSE.EN_0, EN_1, and EN_2 and have the possibility to control the switches that connect the VDAC channel 0 main output to these pins. This makes LESENSE able to excite sensors with output from VDAC channel 0. For LESENSE channel 0,1,2 to enable the DAC output switches in idle phase their IDLECONF should be DACOUT. For these channels to enable the switches in Excite phase the EXMODE should be DACOUT and ALTEX should be 0. Consequently, LESENSE channel 8,9,10 can enable the DAC output switches in excite phase if their EXMODE is DACOUT and ALTEX is 1.

Note:

1. The LESENSE can interact only with VDAC0.
2. The ability to connect the output of VDAC0 to one of LESENSE channels 0, 1 or 2 — including their respective pins LESENSE.EN_0, LESENSE.EN_1 and LESENSE.EN_2 — during the idle phase can be used to (a) immediately stop the oscillations from an inductive excitation phase performed on a LESENSE channel and (b) reduce the possibility for crosstalk among the three LESENSE channels when using multiple inductive LC sensor circuits.

28.3.2 Channel Configuration

LESENSE has 16 individually configurable channels. Each channel has its own set of configuration registers. Channel configuration is split into Four registers; CHx_TIMING, CHx_INTERACT, CHx_EVALCFG and CHx_EVALTHRES. Individual timing for each sensor is configured in CHx_TIMING, sensor interaction is configured in CHx_INTERACT, (static) configurations regarding evaluation of the measurements are done in CHx_EVALCFG and the (dynamic) thresholds used in the various evaluation modes are stored in the CHx_EVALTHRES. For improved readability, CHx_CONF will be used to refer to all the channel configuration registers as a group (CHx_TIMING, CHx_INTERACT, CHx_EVALCFG and CHx_EVALTHRES) in places throughout this chapter.

By default, the channel configuration registers are directly mapped to the channel number. Configuring CFG.SCANCONF makes it possible to alter this mapping.

Configuring SCANCONF to INVMAP will make channels 0-7 use the channel configuration registers for channels 8-15, and vice versa. This feature allows an application to quickly and easily switch the configuration set for the channels.

Setting SCANCONF to TOGGLE will make channel x alternate between using CH_X_CONF and CH_{X+8}_CONF. The configuration used is decided by the state of the corresponding bit in SCANRES. For instance, if channel 3 is performing a scan and bit 3 in SCANRES is set, CH_{11}_CONF will be used. Channels 8 through 15 will toggle between CH_X_CONF and CH_{X-8}_CONF. This mode provides an easy way for implementation of hysteresis on channel events as threshold values can be changed depending on sensor status.

Setting SCANCONF to DECDEF will make the state of the decoder define which scan configuration to be used. If the decoder state is at index 16 or higher, channel x will use CH_{X+8}_CONF, otherwise it will use CH_X configuration. Similarly, channels 8 through 15 will use CH_X configuration when decoder state index is less than 8 and CH_{X-8}_CONF when decoder state index is higher than 7. Allowing the decoder state to define which configuration to use enables easy implementation of hysteresis, for example, as different threshold values can be used for the same channel dependent on the state of the application. Table 28.1 LESENSE Scan Configuration Selection on page 1132 summarizes how channel configuration is selected for different setting of SCANCONF.

Table 28.1. LESENSE Scan Configuration Selection

LESENSE channel x	SCANCONF					
	DIRMAP	INVMAP	TOGGLE		DECDEF	
			SCANRES[n] = 0	SCANRES[n] = 1	DECSTATE < 16	DECSTATE >= 16
0 <= x < 8	CH_x_CONF	CH_{x+8}_CONF	CH_x_CONF	CH_{x+8}_CONF	CH_x_CONF	CH_{x+8}_CONF
8 <= x < 16	CH_x_CONF	CH_{x-8}_CONF	CH_x_CONF	CH_{x-8}_CONF	CH_x_CONF	CH_{x-8}_CONF

Channels are enabled in the CHEN register, where bit x enables channel x. During a scan, all enabled channels are measured, starting with the lowest indexed channel.

28.3.3 Scan Sequence

The LESENSE peripheral clock (LESENSECLK) is a low-speed clock derived from the EM23GRPACLK from the CMU. LESENSECLK is the default used to clock most of the peripheral timing during operation. The high-frequency LESENSE peripheral clock (LESENSEHFCLK) selected in the CMU may also be dyamically used for faster timing of individual sensor measurements. The LESENSE register interface runs on the LSPCLK coming from the CMU to enable faster register access.

All enabled channels are scanned each scan period. A new scan is started according to the conditions set by CFG.SCANMODE. If set to PERIODIC, the scan frequency is generated using a counter which is clocked by LESENSECLK. This counter has its own prescaler. This prescaling factor is configured in the PCPRESC field of TIMCTRL. A new scan sequence is started each time the counter reaches the top value, TIMCTRL.PCTOP. The scan frequency is calculated using Figure 28.2 Scan frequency on page 1133. If SCANMODE is set to ONESHOT, a single scan will be made when START in CMD is set. To start a new scan on a PRS event, set SCANMODE to PRS and configure the LESENSE consumer register in the PRS module.

$$F_{scan} = LESENSECLK/((1 + PCTOP)*2^{PCPRESC})$$

Figure 28.2. Scan frequency

It is possible to interleave additional sensor measurements in between the periodic scans. Issuing a start command when LESENSE is idle will immediately start a new scan, without disrupting the frequency of the periodic scans. If the period counter overflows during the interleaved scan, the periodically scheduled scan will start immediately after the interleaved scan completes.

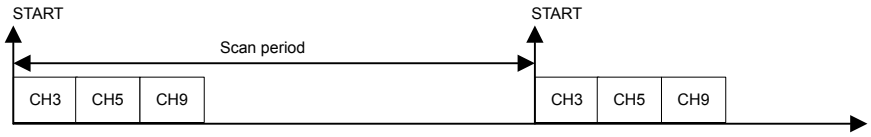


Figure 28.3. Scan Sequence

28.3.4 Sensor Timing

LESENSE includes a timer prescaler for each of the clock sources: A low frequency timer running on LESENSECLK, and a high frequency timer running on LESENSEHFCLK. The low frequency clock is prescaled by configuring LFPRESC in TIMCTRL, and the high frequency clock prescaling factor is configured in AUXPRESC in TIMCTRL. The prescaled versions of these clocks are used for channel sensor timing. A block diagram of the key controls for LESENSE clocking and timing is shown in Figure 28.4 LESENSE Clocking and Timing Control on page 1134.

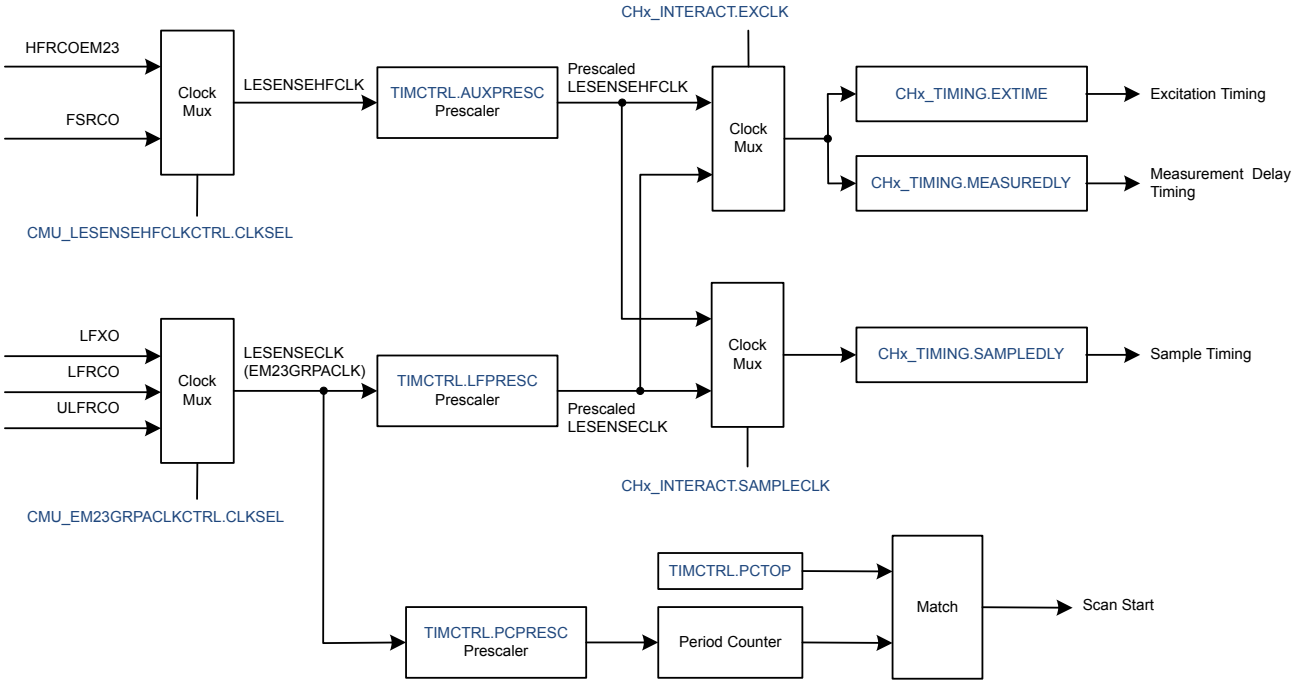


Figure 28.4. LESENSE Clocking and Timing Control

For each channel in the scan sequence, the LESENSE interface has three phases: the idle phase, the excite phase, and the measure phase. The durations of each phase are configured in the CHx_TIMING registers.

Timing of the excite and measure phases can be either a number of $LESENSEHFCLK_{prescaled}$ cycles or a number of $LESENSECLK_{prescaled}$ cycles, depending on which one is selected by the EXCLK field in CHx_INTERACT. Similarly, the sample phase is timed from the clock selected by the SAMPLECLK field in Chx_INTERACT.

The duration of the excite phase is programmed via EXTIME in the CHx_TIMING register. This sets how long the sensor excitation pulse will last.

The duration of the measurement phase is programmed via MEASUREDLY and SAMPLEDLY. The output of the ACMP will be ignored for MEASUREDLY EXCLK cycles after start of the sensor measurement. Sampling of the sensor will happen after SAMPLEDLY SAMPLECLK cycles. The configurable measure and sample delays enable LESENSE to easily define exact time windows for sensor measurements. A start delay can also be inserted before sensor measurement begins by configuring STARTDLY in TIMCTRL. This delay can be used to ensure that a VDAC conversion is complete and voltages have stabilized before sensor measurement begins. To reduce power consumption, the startup of HFRCOEM23/FSRCO (driving LESENSEHFCLK) can be delayed until the system enters the excite phase. This is done by configuring AUXSTARTUP in TIMCTRL to ONDEMAND. This will reduce the time the HFRCOEM23/FSRCO is enabled with the tradeoff that that the starting point for high frequency timing will also be delayed the required oscillator startup time.

Figure 28.5 Timing Diagram, LESENSEHFCLK Based Timing on page 1135 depicts a sensor sequence using $LESENSEHFCLK_{prescaled}$ based timing (EXTIME=6, MEASUREDLY=13, SAMPLEDLY=20), while Figure 28.6 Timing Diagram, LESENSECLK Based Timing on page 1135 depicts a sequence with $LESENSECLK_{prescaled}$ based timing (EXTIME=1, MEASUREDLY=2, SAMPLEDLY=3).

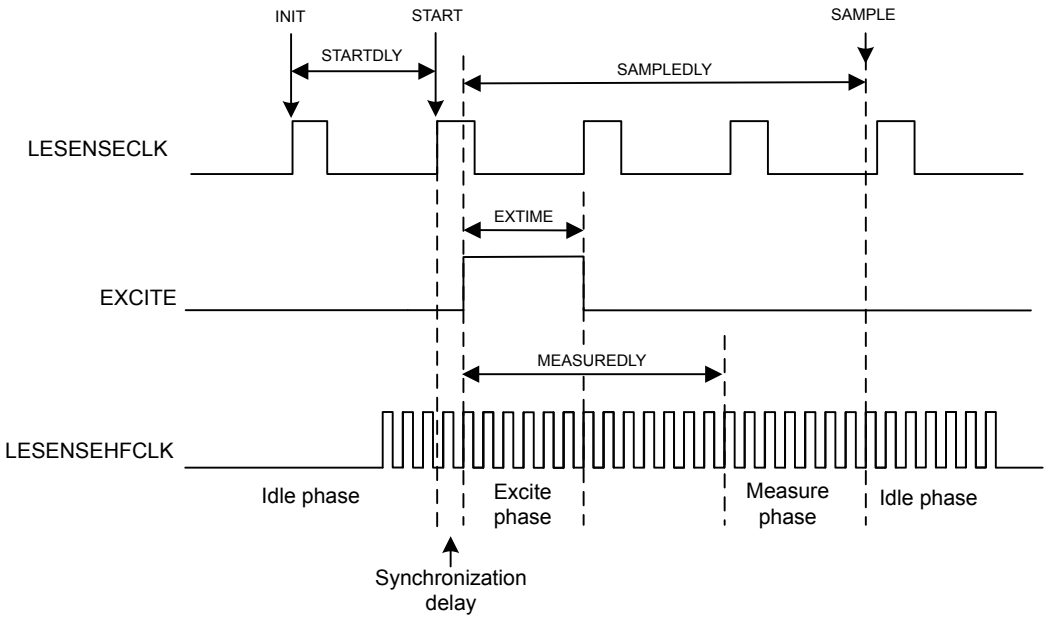


Figure 28.5. Timing Diagram, LESENSEHFCLK Based Timing

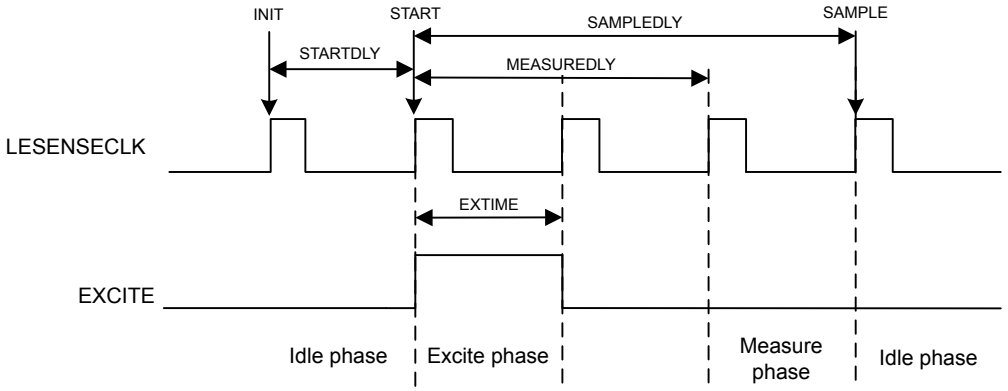


Figure 28.6. Timing Diagram, LESENSECLK Based Timing

28.3.5 Sensor Interaction

Many sensor types require some type of excitation in order to work. LESENSE can generate a variety of sensor stimuli, both on the same pin as the measurement is to be made on, and on alternative pins.

By default, excitation is performed on the LESENSE.CHxOUT signal associated with the channel number. These signals are routed to GPIO using DBUS (see 23.3.12.1 Digital Bus (DBUS)). The mode of the pin during the excitation phase is configured using EXMODE in CHx_INTERACT. The available modes during the excite phase are:

- DISABLED: The pin is disabled.
- HIGH: The pin is driven high.
- LOW: The pin is driven low.
- DACOUT: The pin is connected to the output of VDAC0 channel 0.

Note:

1. Excitation with VDAC0 is available only for LESENSE channels 0, 1 and 2 such that VDAC0.CH0_MAIN_OUT is shorted to the dedicated LESENSE.EN_0, LESENSE.EN_1, and LESENSE.EN_2 pins, respectively. These pins may differ from the GPIOs selected for those specific channels.
2. The fact that LESENSE channels 0, 1 and 2 are the only channels that can be connected to the output of the VDAC makes them especially useful with LC sensor circuits. Because each of these channels can be configured to be connected to VDAC0.CH0_MAIN_OUT during the idle phase, the low output impedance of the VDAC stops oscillations in the LC tank circuit, thereby preventing possible crosstalk between measurement channels in case of multiple LC sensors being sequentially excited.

LESENSE is able to perform sensor excitation on one channel while measuring with another. When ALTEX in CHx_INTERACT is set, the excitation will occur on the alternative excite pin associated with the given channel. The alternative pin associated with LESENSE CH_X is LESENSE $CH_{X+8 \bmod 16}$. Table 28.2 LESENSE Excitation Pin Mapping on page 1136 shows the mapping of excitation and alternate excitation for every channel.

Table 28.2. LESENSE Excitation Pin Mapping

LESENSE channel	ALTEX = 0	ALTEX = 1
0	LESENSE.CH0OUT	LESENSE.CH8OUT
1	LESENSE.CH1OUT	LESENSE.CH9OUT
2	LESENSE.CH2OUT	LESENSE.CH10OUT
3	LESENSE.CH3OUT	LESENSE.CH11OUT
4	LESENSE.CH4OUT	LESENSE.CH12OUT
5	LESENSE.CH5OUT	LESENSE.CH13OUT
6	LESENSE.CH6OUT	LESENSE.CH14OUT
7	LESENSE.CH7OUT	LESENSE.CH15OUT
8	LESENSE.CH8OUT	LESENSE.CH0OUT
9	LESENSE.CH9OUT	LESENSE.CH1OUT
10	LESENSE.CH10OUT	LESENSE.CH2OUT
11	LESENSE.CH11OUT	LESENSE.CH3OUT
12	LESENSE.CH12OUT	LESENSE.CH4OUT
13	LESENSE.CH13OUT	LESENSE.CH5OUT
14	LESENSE.CH14OUT	LESENSE.CH6OUT
15	LESENSE.CH15OUT	LESENSE.CH7OUT

Figure 28.7 Pin Sequencing on page 1137 illustrates the sequencing of the pin associated with the active channel and its alternative excite pin.

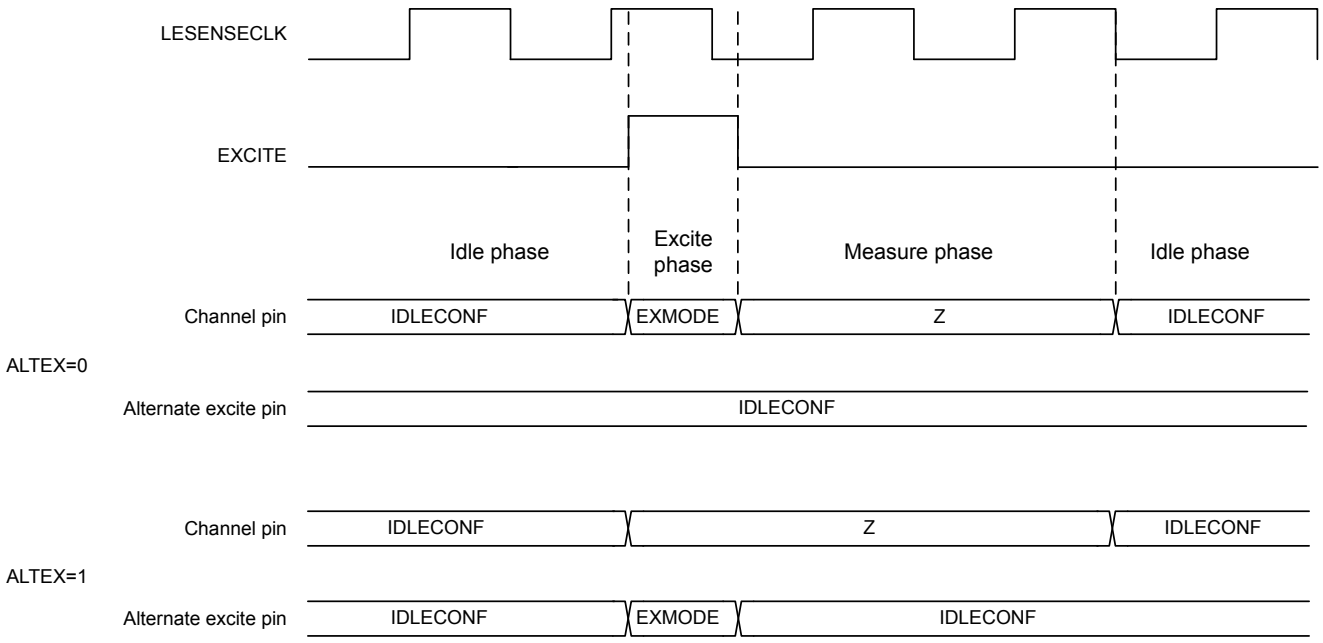


Figure 28.7. Pin Sequencing

Note: When exciting on the pin associated with the active channel, the pin will go through a tri-stated phase before returning to the idle configuration. This will not happen on pins used as alternative excitation pins.

The pin configuration for the idle phase can be configured individually for each LESENSE channel pin in the IDLECONF register. The modes available are the same as the modes available in the excitation phase. During the measure phase, the pin mode on the active channel is always disabled (analog input).

To enable LESENSE to control GPIO, the PORT:PIN have to programmed in the GPIO Route registers for each LESENSE channel being used. In addition, the given pin must be configured as push-pull.

28.3.6 Sensor Sampling

During the measurement phase, LESENSE can sample data from sensors using either the IADC0 or one of the ACMP modules (ACMP0 for LESENSE channels 0-7 and ACMP1 for LESENSE channels 8-15). This is configured in CHx_INTERACT.SAMPLE. Together with the configuration of these peripherals, the CHx_INTERACT.OFFSET field may be used to specify different pins to be sampled for each LESENSE channel (see 28.3.1 Interface Descriptions).

If ACMP is used, LESENSE can evaluate the ACMP output at a single point in time (CHx_INTERACT.SAMPLE = ACMP), or count pulses on the ACMP output (CHx_INTERACT.SAMPLE = ACMPCOUNT) for a programmable period of time.

LESENSE includes the possibility to sample both analog comparators simultaneously, effectively cutting the time spent on sensor interaction in some applications in half. Setting DUALSAMPLE in LESENSE.CFG enables this mode. In dual sample mode, channels X and X+8 are paired, meaning they will be sampled at the same time. Dualsample mode only works when CHx_INTERACT_SAMPLE is set to ACMP.

If the IADC is used, LESENSE will initiate IADC conversions and fetch the IADC data for further evaluation. If the IADC is configured for two's complement bi-polar output, CHx_INTERACT.SAMPLE must be set to ADCDIFF. In this mode, the output from the IADC and the threshold used for comparison will be interpreted as two's complement notation.

The sampled data from IADC or ACMP will be referred to as sensor data in the remainder of this manual.

28.3.7 Sensor Evaluation

When a measurement phase is completed, the sensor data is evaluated by the evaluation block. If the sensor data is taken from ACMP sample in a single point in time (CHx_INTERACT_SAMPLE = ACMP), the evaluation is limited to determening if the sensor data is 0 or 1. For the other sample modes, there are three ways to do sensor evaluation; threshold comparison, sliding window, or step detection. Evaluation mode is configured in CHx_EVALCFG.MODE.

If the evaluation of sensor data evaluates to true, the corresponding bit in the result register, SCANRES, is set. By configuring SETIF in CHx_INTERACT, interrupt flags can also be set on SCANRES events.

Figure 28.8 Scan Result and Interrupt Generation on page 1138 illustrates how the sensor data or ACMP sample is used for evaluation and interrupt generation.

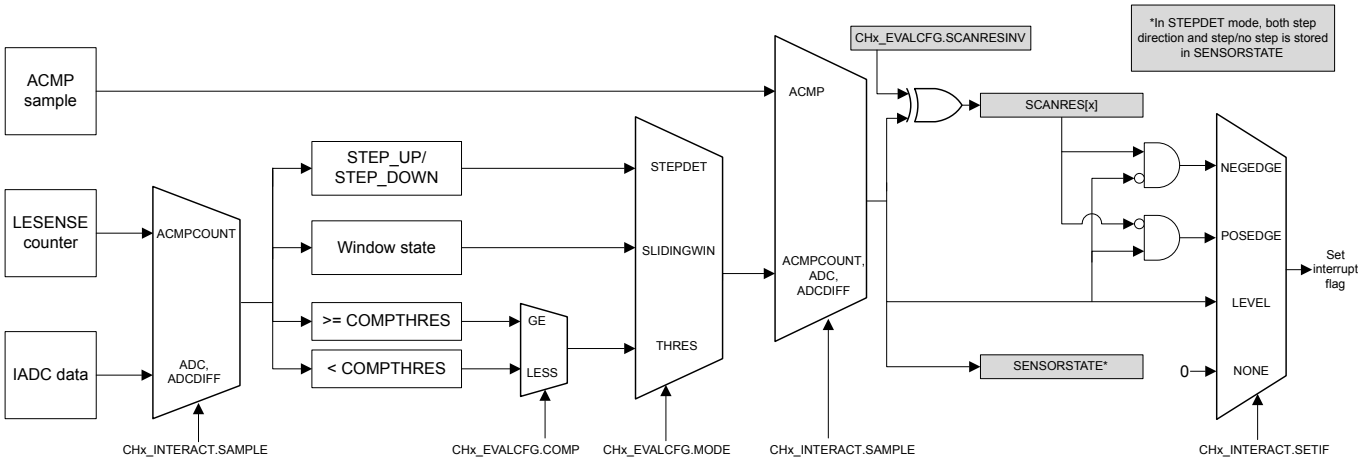


Figure 28.8. Scan Result and Interrupt Generation

The results from sensor data evaluation can be fed into the decoder through the SENSORSTATE register, by setting CHx_EVALCFG.DECODE. In DUALSAMPLE mode, results from both the sampled ACMPs will be stored in both SCANRES and SENSORSTATE (if enabled).

28.3.8 Threshold Comparison

In threshold comparison mode, the sensor data is compared to the threshold configured in CHx_EVALTHRES. There are two modes of threshold comparison; 'less than', and 'greater than or equal'. Threshold comparison mode is configured in CHx_EVALCFG.COMP.

28.3.9 Sliding Window

In sliding window mode, the sensor data is compared against the upper and lower limits of a window range. The window is defined by a base, given by CHx_EVALTHRES, and a size configured in EVALCTRL_WINSIZE. The window size is constant and the same for all LESENSE channels, while the base is specific to each channel and will be updated by LESENSE when the sensor data is outside the current window range. If the sensor data is within the window range, the sensor evaluation will remain the same as it was for the previous measurement. If the sensor data is below the window range, the measurement will evaluate as false. If the sensor data is above the window range, the measurement will evaluate as true. In both cases, the window base in CHx_EVALTHRES will be updated to reflect the new window range. Figure 28.9 Sliding Window on page 1139 shows how the sliding window evaluation mode can be used to implement a system with two self calibrating thresholds.

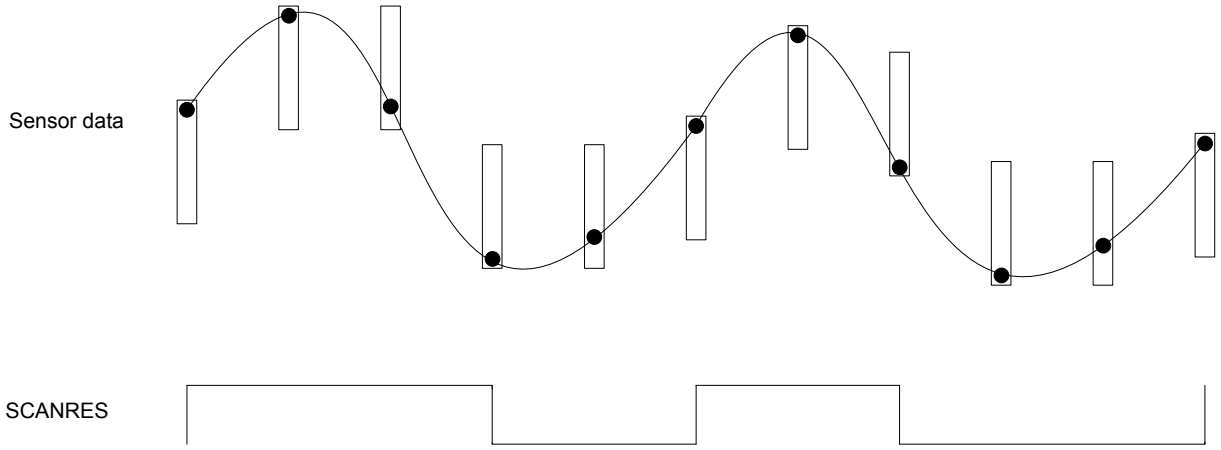


Figure 28.9. Sliding Window

28.3.10 Step Detection

Step detection is used to detect steps in the sensor data compared to sensor data from the previous measurement. The size of the step is configured in EVALCTRL_WINSIZE. In this mode, step up and step down are evaluated as described in Figure 28.10 Step detection on page 1139:

$$STEP\_UP = SENSORDATA_i >= SENSORDATA_{i-1} + EVALCTRL\_WINSIZE$$

$$STEP\_DOWN = SENSORDATA_i < SENSORDATA_{i-1} - EVALCTRL\_WINSIZE$$

Figure 28.10. Step detection

If either a step up or a step down is detected, the SCANRES bit for the active channel will be set. In addition, the STEPDIR bit for the channel will be updated to indicate if a step up or a step down was detected. STEPDIR = 1 indicates a step up. In this mode, previous sensor data is stored in CHx_EVALTHRES.

28.3.11 Decoder

Many applications require some sort of processing of the sensor readings, for instance in the case of quadrature decoding. In quadrature decoding, the sensors repeatedly pass through a set of states which correspond to the position of the sensors. This sequence, and many other decoding schemes, can be described as a finite state machine. To support this type of decoding without CPU intervention, LESENSE includes a highly configurable decoder, capable of decoding input from up to four sensors. The decoder is implemented as a programmable state machine with up to 64 arcs which can represent transitions between up to 32 states. When doing a sensor scan, the results from the sensors are placed in the decoder input register, SENSORSTATE, if DECODE in CHx_INTERACT is set, as depicted in Figure 28.11 Sensor Scan and Decode Sequence on page 1140.

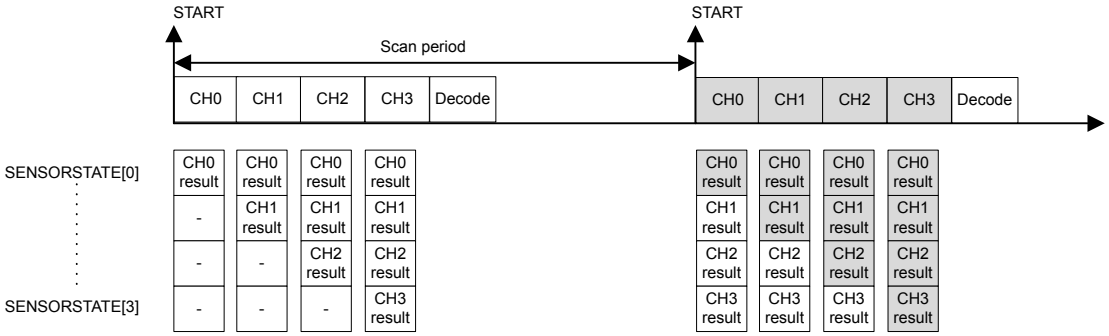


Figure 28.11. Sensor Scan and Decode Sequence

The current state is indicated by the DECSTATE register. At the end of a scan sequence, the decoder evaluates the results placed in SENSORSTATE against the defined STx_ARC registers and the current state. The decoder evaluates all the ARCs together in parallel and uses one LESENSECLK cycle to determine the next state.

The transition criteria is established by CURRSTATE, SMASK, and SCOMP in the STx_ARC registers. SMASK defines which bits of SENSORSTATE to mask (ignore). A '0' in SMASK means that the corresponding bit in SENSORSTATE is evaluated, while a '1' in SMASK means that bit will be masked. If STx_ARC.CURRSTATE == DECSTATE, and the non-masked bits in SENSORSTATE are equal to those defined in SCOMP, then the transition criteria evaluates to True, and the state machine transitions to the value specified by STxARC.NEXTSTATE.

It is possible that multiple ARCs will match the transition criteria. In this case, the lowest numbered ARC that meets the transition criteria determines the next state.

Note: When using the decoder, it is important that at least one ARC be configured for the initial state (CURRSTATE == 0), and that the ARCs be configured to not produce any dead states (a state which is the NEXTSTATE of an ARC but is not the CURRSTATE of any other ARC). Doing otherwise will result in the possibility of the decoder getting stuck where no further state transitions are possible. If this happens, a software reset of the LESENSE block would need to be performed.

Upon a state transition, LESENSE can generate a pulse on one or more of the decoder PRS producers. Which producers to generate a pulse on is configured in the PRSACT bit field. If PRSCNT in DECCTRL is set, count signals will be generated on decoder PRS producers 0 and 1 according to the PRSACT configuration. In this mode, producer 0 will pulse each time a count event occurs while producer 1 indicates the count direction (1 for up and 0 for down). The count direction will be kept at its previous state in between count events. The pulse counter on the chip may be used to keep track of events by listening to these PRS outputs. For this the PRS needs to be programmed to configure the producer of the channel(s) from the LESENSE decoder and those same channel(s) are consumed by the PCNT module.

Table 28.3. LESENSE Decoder PRS Outputs (DECCTRL.PRSCNT = 0)

PRSACT Value	Description
0	No PRS output generated
1	Pulse generated on LESENSE PRS output 0
2	Pulse generated on LESENSE PRS output 1
3	Pulse generated on LESENSE PRS output 0 and 1

PRSACT Value	Description
4	Pulse generated on LESENSE PRS output 2
5	Pulse generated on LESENSE PRS output 0 and 2
6	Pulse generated on LESENSE PRS output 1 and 2
7	Pulse generated on LESENSE PRS output 0 , 1 and 2

Table 28.4. LESENSE Decoder PRS Outputs (DECCTRL.PRSCNT = 1)

PRSACT Value	Description
0	No Count Outputs
1	Count Up
2	Count Down
4	Pulse generated on LESENSE PRS output 2
5	Count Up and Pulse generated on LESENSE PRS output 2
6	Count Down and Pulse generated on LESENSE PRS output 2

The DECODER has a PRS producer named DECCMP. This output can be used to indicate which state, or subset for states, the decoder is currently in. This PRS output is enabled by setting DECCMPEN in PRSCTRL, and configured through DECCMPMASK and DECCMPVAL in PRSCTRL. The value of this PRS output is given by Figure 28.12 DECCMP PRS output on page 1141,

PRS_DECCMP = (DECSTATE & ~DECCMPMASK) = (DECCMPVAL & ~DECCMPMASK)

Figure 28.12. DECCMP PRS output

If SETIF is set, the DECODER interrupt flag will be set when the transition occurs. If INTMAP in DECCTRL and SETIF is set, a transition from state x and x+16 will set the CHx interrupt flag in addition to the DECODER flag.

To prevent unnecessary interrupt requests or PRS outputs when the decoder toggles back and forth between two states, a hysteresis option is available. The hysteresis function is triggered if a transition from STATE A to STATE B is preceded by a transition from STATE B to STATE A, and vice-versa. Figure 28.13 Decoder Hysteresis on page 1142 illustrates how the hysteresis triggers upon state transitions.

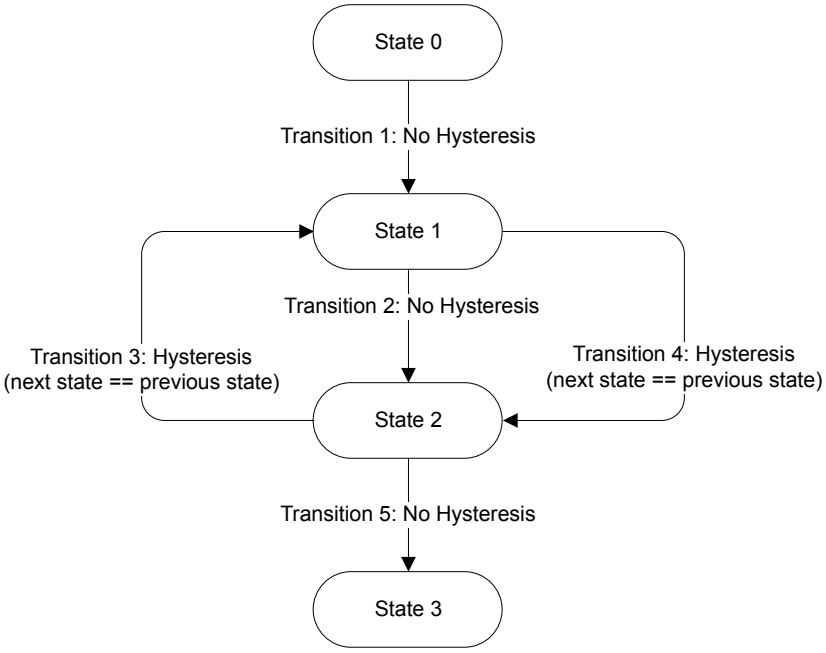


Figure 28.13. Decoder Hysteresis

- When HYSTPRSx is set, PRS signal x is suppressed when the hysteresis triggers.
- When HYSTIRQ is set, interrupt requests are suppressed when the hysteresis triggers.

28.3.12 Measurement Results

LESENSE has a FIFO for storage of up to 16 results from sensor measurements. Each time LESENSE writes data to the result buffer, the result count, RESCOUNT, is incremented. Each time a new result is read through the RESFIFO register, the result count, RESCOUNT, is decremented. The result count will not be decremented if there is no valid, unread data in the result buffer.

Note: The 20-bit wide RESFIFO.BUFDATASRC bitfield contains both the LESENSE result (least-significant 16 bits) as well as the LESENSE channel index (most-significant 4 bits)

LESENSE has a FIFO for storage of up to 16 results from sensor measurements. Each time LESENSE writes data to the result buffer, the result count, RESCOUNT, is incremented. Each time a new result is read through the RESFIFO register, the result count, RESCOUNT, is decremented. The result count will not be decremented if there is no valid, unread data in the result buffer.

LESENSE will not write additional data to a full result buffer and will flag a FIFO overflow in the STATUS register which can also generate an interrupt. Reading the FIFO involves synchronization of the pointers from the read (APB) to the write domain (LF) which will result in STATUS.READBUSY being high. The user should wait for the READBUSY to go low before disabling the LESENSE clock in the CMU. The FIFO read will be non-blocking. By writing the CLEARBUF field, the user can flush the FIFO. When the FIFO is flushing the FLUSHING bit in the STATUS register will be high. It is recommended that the user not flush the FIFO when the scan is running because a FIFO write during FLUSHING may be flushed as well. The APB bus is not stalled during a FLUSH. FLUSHING will also involve pointer synchronization so READBUSY can be high during and after FLUSHING. The FIFO will generate interrupt flags(RESWFIF) when the RESCOUNT reaches the CFG.RESFIDL.

During a scan, the state of each sensor is stored in SCANRES. If a sensor triggers, a 1 is stored in SCANRES, otherwise a 0 is stored in SCANRES. Whether or not a sensor is said to be triggered depends of the configuration for the given channel, and the sensor evaluation mode.

If SAMPLE is set to ACMP, the sensor is said to be triggered if the output from the analog comparator is 1 when sensor sampling is performed. If SAMPLE is set to ACMPCOUNT, a sensor is said to be triggered if the LESENSE counter value is greater than or equal, or less than CHx_EVALTHRES, depending on the configuration of COMP. If STRSAMPLE in CHx_EVALCFG is set, the sensor measurement (ACMP sample, IADC sample, or counter value) for the channel will be stored in the result FIFO. If STRSCANRES in CFG is set, the result vector, SCANRES, will also be stored in the result buffer. This will be stored after each scan and will be interleaved with any stored sensor measurements.

28.3.13 DMA Requests

LESENSE issues a DMA request when the RESCOUNT reaches CFG.RESFIDL, this request is cleared when the buffer level drops below the threshold. A single DMA request is also set whenever there is unread data in the buffer. CFG.DMAWU controls if the LESENSE can send a request for a wakeup from EM2 to transfer data once the RESCOUNT reaches CFG.RESFIDL.

The DMA controller shall read the RESFIFO register.

28.3.14 PRS

LESENSE is an asynchronous PRS producer and consumer. For LESENSE as a PRS consumer, other PRS producers can start a LESENSE scan. From a producer standpoint, LESENSE generates PRS signals from the programmable FSM which can be used by the PRS consumers.

28.4 Programmer's Model

The following steps show how to configure LESENSE to scan through 4 buttons 100 times per second, issuing an interrupt if one of them is pressed.

1. Assuming $PERCLK_{LESENSE}$ is 32kHz, set PCPRESC to 3 and PCTOP to 39 in PERCTRL. This will make the scan frequency to 100 Hz.
2. Enable channels 0 through 3 in CHEN and IDLECONF for these channels to DISABLED. The GPIO for these channels will need to configured in the GPIO peripheral.
3. Configure and enable the ACMP.
4. Configure the following bit fields in CHx_CONF for channel 0 through 3
   a. Set EXTIME to 0. No excitation is needed in this mode.
   b. Set SAMPLE to ACMPCOUNT and COMP to LESS. This makes LESENSE interpret a sensor as active if the frequency of the channel drops below the threshold, i.e. the button is pressed.
   c. set SAMPLEDLY to an appropriate value, each sensor will be measured for SAMPLEDLY / $PERCLK_{LESENSE}$ seconds. MEASUREDLY should be set to 0.
5. Enable Interrupts on Channels 0 through 3.
6. Configure EVALTHRES for Channels 0 to 3. An interrupt will be issued if the counter value for a sensor is below EVALTHRES for the channel after the mesurement phase.
7. Enable LESENSE by writing EN to 1.
8. Start scan sequence by writing 1 to START in CMD.
9. To store the results of the enabled channels into the LESENSE FIFO configure STRSAMPLE for these channels as 1. STRSAMPLE is in a CONFIG register so it should be configured before enabling LESENSE.
10. To configure channels to be used by the decoder set DECODE in CHx_EVALCFG.Assume that we have set DECODE for channels 0 to 3.
11. The decoder will be called to act upon the SENSORSTATE at the end of the scan. In this case it will be after channel 3 is measured and evaluated.
12. To setup the decoder FSM configure the STn_ARC registers before enabling the LESENSE.The reset value for the DECSTATE is 0 so there should be atleast one STn_ARC with the current state as 0 otherwise no matter what the SENSORSTATE is there won't be a state transition. For the same reason there should not be any dead states (a state which is a NEXTSTATE of an ARC but is not the CURRENTSTATE of any ARC) while configuring the FSM else the decoder will be stuck there and the USER will need to perform a SWRST.

28.5 LESENSE Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	LESENSE_IPVERSION	R	IPVERSION
0x004	LESENSE_EN	RW ENABLE	Enable
0x008	LESENSE_SWRST	RW SWRST	Software Reset Register
0x00C	LESENSE_CFG	RW CONFIG	Configuration
0x010	LESENSE_TIMCTRL	RW CONFIG	Timing Control
0x014	LESENSE_PERCTRL	RW CONFIG	Peripheral Control
0x018	LESENSE_DECCTRL	RW	Decoder Control
0x01C	LESENSE_EVALCTRL	RW CONFIG	LESENSE Evaluation
0x020	LESENSE_PRSCTRL	RW	PRS Control
0x024	LESENSE_CMD	W LFSYNC	Command
0x028	LESENSE_CHEN	RW	Channel Enable
0x02C	LESENSE_SCANRES	RH	Scan Result
0x030	LESENSE_STATUS	RH	Status
0x034	LESENSE_RESCOUNT	RH	Result FIFO Count
0x038	LESENSE_RESFIFO	RH	Result Fifo
0x03C	LESENSE_CURCH	RH	Current Channel Index
0x040	LESENSE_DECSTATE	RH	Current Decoder State
0x044	LESENSE_SENSORSTATE	RH	Sensor State
0x048	LESENSE_IDLECONF	RW CONFIG	IDLE Configuration
0x050	LESENSE_SYNCBUSY	RH	Synchronization
0x060	LESENSE_IF	RWH INTFLAG	Interrupt Flags
0x064	LESENSE_IEN	RW	Interrupt Enables
0x100	LESENSE_CHx_TIMING	RW CONFIG	Scan Configuration
0x104	LESENSE_CHx_INTERACT	RW CONFIG	Scan Configuration
0x108	LESENSE_CHx_EVALCFG	RW CONFIG	Scan Configuration
0x10C	LESENSE_CHx_EVALTHRES	RWH LFSYNC	Scan Confguration
0x200	LESENSE_STx_ARC	RW CONFIG	State Transition Arc
0x1000	LESENSE_IPVERSION_SET	R	IPVERSION
0x1004	LESENSE_EN_SET	RW ENABLE	Enable
0x1008	LESENSE_SWRST_SET	RW SWRST	Software Reset Register
0x100C	LESENSE_CFG_SET	RW CONFIG	Configuration
0x1010	LESENSE_TIMCTRL_SET	RW CONFIG	Timing Control
0x1014	LESENSE_PERCTRL_SET	RW CONFIG	Peripheral Control
0x1018	LESENSE_DECCTRL_SET	RW	Decoder Control
0x101C	LESENSE_EVALCTRL_SET	RW CONFIG	LESENSE Evaluation

Offset	Name	Type	Description
0x1020	LESENSE_PRSCTRL_SET	RW	PRS Control
0x1024	LESENSE_CMD_SET	W LFSYNC	Command
0x1028	LESENSE_CHEN_SET	RW	Channel Enable
0x102C	LESENSE_SCANRES_SET	RH	Scan Result
0x1030	LESENSE_STATUS_SET	RH	Status
0x1034	LESENSE_RESCOUNT_SET	RH	Result FIFO Count
0x1038	LESENSE_RESFIFO_SET	RH	Result Fifo
0x103C	LESENSE_CURCH_SET	RH	Current Channel Index
0x1040	LESENSE_DECSTATE_SET	RH	Current Decoder State
0x1044	LESENSE_SENSOR-STATE_SET	RH	Sensor State
0x1048	LESENSE_IDLECONF_SET	RW CONFIG	IDLE Configuration
0x1050	LESENSE_SYNCBUSY_SET	RH	Synchronization
0x1060	LESENSE_IF_SET	RWH INTFLAG	Interrupt Flags
0x1064	LESENSE_IEN_SET	RW	Interrupt Enables
0x1100	LESENSE_CHx_TIMING_SET	RW CONFIG	Scan Configuration
0x1104	LESENSE_CHx_INTER-ACT_SET	RW CONFIG	Scan Configuration
0x1108	LESENSE_CHx_EVALCFG_SET	RW CONFIG	Scan Configuration
0x110C	LESENSE_CHx_EVAL-THRES_SET	RWH LFSYNC	Scan Confguration
0x1200	LESENSE_STx_ARC_SET	RW CONFIG	State Transition Arc
0x2000	LESENSE_IPVERSION_CLR	R	IPVERSION
0x2004	LESENSE_EN_CLR	RW ENABLE	Enable
0x2008	LESENSE_SWRST_CLR	RW SWRST	Software Reset Register
0x200C	LESENSE_CFG_CLR	RW CONFIG	Configuration
0x2010	LESENSE_TIMCTRL_CLR	RW CONFIG	Timing Control
0x2014	LESENSE_PERCTRL_CLR	RW CONFIG	Peripheral Control
0x2018	LESENSE_DECCTRL_CLR	RW	Decoder Control
0x201C	LESENSE_EVALCTRL_CLR	RW CONFIG	LESENSE Evaluation
0x2020	LESENSE_PRSCTRL_CLR	RW	PRS Control
0x2024	LESENSE_CMD_CLR	W LFSYNC	Command
0x2028	LESENSE_CHEN_CLR	RW	Channel Enable
0x202C	LESENSE_SCANRES_CLR	RH	Scan Result
0x2030	LESENSE_STATUS_CLR	RH	Status
0x2034	LESENSE_RESCOUNT_CLR	RH	Result FIFO Count
0x2038	LESENSE_RESFIFO_CLR	RH	Result Fifo
0x203C	LESENSE_CURCH_CLR	RH	Current Channel Index

Offset	Name	Type	Description
0x2040	LESENSE_DECSTATE_CLR	RH	Current Decoder State
0x2044	LESENSE_SENSOR-STATE_CLR	RH	Sensor State
0x2048	LESENSE_IDLECONF_CLR	RW CONFIG	IDLE Configuration
0x2050	LESENSE_SYNCBUSY_CLR	RH	Synchronization
0x2060	LESENSE_IF_CLR	RWH INTFLAG	Interrupt Flags
0x2064	LESENSE_IEN_CLR	RW	Interrupt Enables
0x2100	LESENSE_CHx_TIMING_CLR	RW CONFIG	Scan Configuration
0x2104	LESENSE_CHx_INTER-ACT_CLR	RW CONFIG	Scan Configuration
0x2108	LESENSE_CHx_EVALCFG_CLR	RW CONFIG	Scan Configuration
0x210C	LESENSE_CHx_EVAL-THRES_CLR	RWH LFSYNC	Scan Confguration
0x2200	LESENSE_STx_ARC_CLR	RW CONFIG	State Transition Arc
0x3000	LESENSE_IPVERSION_TGL	R	IPVERSION
0x3004	LESENSE_EN_TGL	RW ENABLE	Enable
0x3008	LESENSE_SWRST_TGL	RW SWRST	Software Reset Register
0x300C	LESENSE_CFG_TGL	RW CONFIG	Configuration
0x3010	LESENSE_TIMCTRL_TGL	RW CONFIG	Timing Control
0x3014	LESENSE_PERCTRL_TGL	RW CONFIG	Peripheral Control
0x3018	LESENSE_DECCTRL_TGL	RW	Decoder Control
0x301C	LESENSE_EVALCTRL_TGL	RW CONFIG	LESENSE Evaluation
0x3020	LESENSE_PRSCTRL_TGL	RW	PRS Control
0x3024	LESENSE_CMD_TGL	W LFSYNC	Command
0x3028	LESENSE_CHEN_TGL	RW	Channel Enable
0x302C	LESENSE_SCANRES_TGL	RH	Scan Result
0x3030	LESENSE_STATUS_TGL	RH	Status
0x3034	LESENSE_RESCOUNT_TGL	RH	Result FIFO Count
0x3038	LESENSE_RESFIFO_TGL	RH	Result Fifo
0x303C	LESENSE_CURCH_TGL	RH	Current Channel Index
0x3040	LESENSE_DECSTATE_TGL	RH	Current Decoder State
0x3044	LESENSE_SENSOR-STATE_TGL	RH	Sensor State
0x3048	LESENSE_IDLECONF_TGL	RW CONFIG	IDLE Configuration
0x3050	LESENSE_SYNCBUSY_TGL	RH	Synchronization
0x3060	LESENSE_IF_TGL	RWH INTFLAG	Interrupt Flags
0x3064	LESENSE_IEN_TGL	RW	Interrupt Enables
0x3100	LESENSE_CHx_TIMING_TGL	RW CONFIG	Scan Configuration

Offset	Name	Type	Description
0x3104	LESENSE_CHx_INTERACT_TGL	RW CONFIG	Scan Configuration
0x3108	LESENSE_CHx_EVALCFG_TGL	RW CONFIG	Scan Configuration
0x310C	LESENSE_CHx_EVALTHRES_TGL	RWH LFSYNC	Scan Confguration
0x3200	LESENSE_STx_ARC_TGL	RW CONFIG	State Transition Arc

28.6 LESENSE Register Description

28.6.1 LESENSE_IPVERSION - IPVERSION

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IPVERSION**
	IPVERSION			

28.6.2 LESENSE_EN - Enable

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disabling**
	Module is in disabling			
0	EN	0x0	RW	**Enable**
	Global Enable of LESENSE functions			

Value	Mode	Description
0	DISABLE	Disable
1	ENABLE	Enable

28.6.3 LESENSE_SWRST - Software Reset Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	W
Name																															RESETTING	SWRST

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESETTING	0x0	R	**Software reset busy status**
	Module is undergoing SWRST			
0	SWRST	0x0	W	**Software reset command**
	Trigger SWRST			

28.6.4 LESENSE_CFG - Configuration

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0						0x0				0x0	0x0	0x0		0x0		0x0	
Access															RW						RW				RW	RW	RW		RW		RW	
Name															DEBUGRUN						RESFIDL				DMAWU	STRSCANRES	DUALSAMPLE		SCANCONF		SCANMODE	

Bit	Name	Reset	Access	Description
31:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	DEBUGRUN	0x0	RW	**Debug Mode Run Enable**
	Set to keep LESENSE running in debug mode.			
	Value	Mode		Description
	0	X0		LESENSE can not start new scans in debug mode
	1	X1		LESENSE can start new scans in debug mode
16:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:8	RESFIDL	0x0	RW	**Result FIFO level**
	Result FIFO interrupt and DMA trigger level			
7	DMAWU	0x0	RW	**DMA wake-up from EM2**
	When set, DMA can wake and collect information based on FIFO levels			
	Value	Mode		Description
	0	DISABLE		No DMA wake-up from EM2
	1	ENABLE		DMA wake-up from EM2 when FIFO count is greater or equal to RESFIDL
6	STRSCANRES	0x0	RW	**Enable storing of SCANRES**
	When set, SCANRES will be stored in the result buffer after each scan			
5	DUALSAMPLE	0x0	RW	**Enable dual sample mode**
	When set, both ACMPs will be sampled simultaneously.			
4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:2	SCANCONF	0x0	RW	**Select scan configuration**
	These bits control which CHx_CONF registers to be used.			
	Value	Mode		Description

Bit	Name	Reset	Access	Description
	0	DIRMAP		The channel configuration register registers used are directly mapped to the channel number.
	1	INVMAP		The channel configuration registers used are CH[X+8]_CONF for channels 0-7 and CH[X-8]_CONF for channels 8-15.
	2	TOGGLE		The channel configuration registers used toggle between CH[X]_CONF and CH[X+8]_CONF when channel x triggers
	3	DECDEF		The decoder state defines the CONF registers to be used.
1:0	SCANMODE	0x0	RW	**Configure scan mode**
	These bits control how the scan frequency is decided			
	Value	Mode		Description
	0	PERIODIC		A new scan is started each time the period counter overflows
	1	ONESHOT		A single scan is performed when START in CMD is set
	2	PRS		Pulse on PRS channel

28.6.5 LESENSE_TIMCTRL - Timing Control

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x0					0x0				0x0									0x0				0x0					0x0	
Access				RW					RW				RW									RW				RW					RW	
Name				AUXSTARTUP					STARTDLY				PCTOP									PCPRESC				LFPRESC					AUXPRESC	

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28	AUXSTARTUP	0x0	RW	**AUX startup config**

Value	Mode	Description
0	PREDEMAND	Request oscillator .5 LESENSECLK cycle before sensing starts
1	ONDEMAND	Request oscillator at sensing time

27:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:22	STARTDLY	0x0	RW	**Start delay configuration**
	Delay sensor interaction STARTDELAY LESENSECLK cycles for each channel			
21:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:12	PCTOP	0x0	RW	**Period counter top value**
	Top value for the period counter			
11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10:8	PCPRESC	0x0	RW	**Period counter prescaling**
	Prescaling factor for period counter used for scan interval			

Value	Mode	Description
0	DIV1	The period counter clock frequency is LESENSECLK/1
1	DIV2	The period counter clock frequency is LESENSECLK/2
2	DIV4	The period counter clock frequency is LESENSECLK/4
3	DIV8	The period counter clock frequency is LESENSECLK/8
4	DIV16	The period counter clock frequency is LESENSECLK/16
5	DIV32	The period counter clock frequency is LESENSECLK/32
6	DIV64	The period counter clock frequency is LESENSECLK/64
7	DIV128	The period counter clock frequency is LESENSECLK/128

Bit	Name	Reset	Access	Description
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	LFPRESC	0x0	RW	**Prescaling factor for low frequency time**

Prescaling factor for low frequency timer used for Sensor timing

Value	Mode	Description
0	DIV1	Low frequency timer is clocked with LESENSECLK/1
1	DIV2	Low frequency timer is clocked with LESENSECLK/2
2	DIV4	Low frequency timer is clocked with LESENSECLK/4
3	DIV8	Low frequency timer is clocked with LESENSECLK/8
4	DIV16	Low frequency timer is clocked with LESENSECLK/16
5	DIV32	Low frequency timer is clocked with LESENSECLK/32
6	DIV64	Low frequency timer is clocked with LESENSECLK/64
7	DIV128	Low frequency timer is clocked with LESENSECLK/128

3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	AUXPRESC	0x0	RW	**Prescaling factor for high frequency tim**

Prescaling factor for high frequency timer used for Senosr timing

Value	Mode	Description
0	DIV1	High frequency timer is clocked at LESENSEHFCLK/1
1	DIV2	High frequency timer is clocked at LESENSEHFCLK/2
2	DIV4	High frequency timer is clocked at LESENSEHFCLK/4
3	DIV8	High frequency timer is clocked at LESENSEHFCLK/8

28.6.6 LESENSE_PERCTRL - Peripheral Control

Offset	Bit Position																														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset							0x0	0x0		0x0		0x0												0x0		0x0				0x0		
Access							RW	RW		RW		RW												RW		RW				RW		
Name							ACMP1INV	ACMP0INV		ACMP1MODE		ACMP0MODE												DACCONVTRIG		DACSTARTUP				DACCH0DATA		

Bit	Name	Reset	Access	Description
31:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25	ACMP1INV	0x0	RW	**Invert analog comparator 1 output**
	Bit to invert the ACMP1 output before using it within LESENSE			
24	ACMP0INV	0x0	RW	**Invert analog comparator 0 output**
	Bit to invert the the ACMP0 output before using it within LESENSE			
23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22	ACMP1MODE	0x0	RW	**ACMP1 mode**
	Configure how LESENSE controls ACMP1			

Value	Mode	Description
0	MUX	LESENSE controls the POSSEL of ACMP1
1	MUXTHRES	LESENSE POSSEL and reference divider of ACMP1

Bit	Name	Reset	Access	Description
21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20	ACMP0MODE	0x0	RW	**ACMP0 mode**
	Configure how LESENSE controls ACMP0			

Value	Mode	Description
0	MUX	LESENSE controls POSSEL of ACMP0
1	MUXTHRES	LESENSE controls POSSEL and reference divider of ACMP0

Bit	Name	Reset	Access	Description
19:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	DACCONVTRIG	0x0	RW	**DAC conversion trigger configuration**
	Decides if the DAC is triggerd before every channel measurement or once per scan			

Value	Mode	Description
0	CHANNELSTART	DAC is enabled before every LESENSE channle measurement.
1	SCANSTART	DAC is only enabled once per scan.

Bit	Name	Reset	Access	Description
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	DACSTARTUP	0x0	RW	**DAC startup configuration**

Decides if the DAC is triggered 0.5 or 1 LESENSECLK cycle before START

Value	Mode	Description
0	FULLCYCLE	DAC is started a full LESENSECLK before sensor interaction starts.
1	HALFCYCLE	DAC is started half a LESENSECLK cycle before sensor interaction starts.

5:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	DACCH0DATA	0x0	RW	**DAC CH0 data selection.**

Setting this bit will override the data the DAC uses for conversion

Value	Mode	Description
0	DACDATA	DAC data is defined by CH0DATA in the DAC interface.
1	THRES	DAC data is defined by THRES in CHx_INTERACT.

1:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

28.6.7 LESENSE_DECCTRL - Decoder Control

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0	0x0	0x0	0x0	0x0	0x0		0x0
Access																									RW	RW	RW	RW	RW	RW		RW
Name																									PRSCNT	HYSTIRQ	HYSTPRS2	HYSTPRS1	HYSTPRS0	INTMAP		DECDIS

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	PRSCNT	0x0	RW	**Enable count mode on decoder PRS channel**
	When set, decoder PRS0 and PRS1 will be used to produce output which can be used by a PCNT to count up or down.			
6	HYSTIRQ	0x0	RW	**Enable decoder hysteresis on interrupt r**
	When set, hysteresis is enabled in the decoder, suppressing interrupt requests.			
5	HYSTPRS2	0x0	RW	**Enable decoder hysteresis on PRS2 output**
	When set, hysteresis is enabled in the decoder, suppressing changes on PRS channel 2			
4	HYSTPRS1	0x0	RW	**Enable decoder hysteresis on PRS1 output**
	When set, hysteresis is enabled in the decoder, suppressing changes on PRS channel 1			
3	HYSTPRS0	0x0	RW	**Enable decoder hysteresis on PRS0 output**
	When set, hysteresis is enabled in the decoder, suppressing changes on PRS channel 0			
2	INTMAP	0x0	RW	**Enable decoder to channel interrupt map**
	When set, a transition from state x in the decoder will set interrupt flag CH[x mod 16]			
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	DECDIS	0x0	RW	**Disable the decoder**
	When set, the decoder is disabled. When disabled the decoder will keep its current state			

28.6.8 LESENSE_EVALCTRL - LESENSE Evaluation

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									WINSIZE							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	WINSIZE	0x0	RW	**Sliding window and step detection size** In sliding window mode, this bit-field configures the window size. In step detection mode, this bit-field is used to configure the threshold for step detection

28.6.9 LESENSE_PRSCTRL - PRS Control

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x0						0x0								0x0		
Access																RW						RW								RW		
Name																DECCMPEN						DECCMPMASK								DECCMPVAL		

Bit	Name	Reset	Access	Description
31:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	DECCMPEN	0x0	RW	**Enable PRS output DECCMP** Enables decoder state compare match PRS output
15:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12:8	DECCMPMASK	0x0	RW	**Decoder state compare value mask** Masks COMPVAL and DECSTATE for comparison
7:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4:0	DECCMPVAL	0x0	RW	**Decoder state compare value** Triggers prs output when equal to DECSTATE

28.6.10 LESENSE_CMD - Command

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													W(nB)	W(nB)	W(nB)	W(nB)
Name																													CLEARBUF	DECODE	STOP	START

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	CLEARBUF	0x0	W(nB)	**Clear result buffer**
2	DECODE	0x0	W(nB)	**Start decoder**
1	STOP	0x0	W(nB)	**Stop scanning of sensors** If issued during a scan, the command will take effect after scan completion.
0	START	0x0	W(nB)	**Start scanning of sensors.**

28.6.11 LESENSE_CHEN - Channel Enable

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	RW															
Name																	CHEN															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	CHEN	0x0	RW	**Enable scan channel** Set bit X to enable channel X

28.6.12 LESENSE_SCANRES - Scan Result

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																0x0															
Access	R																R															
Name	STEPDIR																SCANRES															

Bit	Name	Reset	Access	Description
31:16	STEPDIR	0x0	R	**Direction of previous step detection**
	In step detection mode, bit X will be set if a step up was detected on channel X; write to initialize when LESENSE is disabled.			
15:0	SCANRES	0x0	R	**Scan results**
	Bit X will be set depending on channel X evaluation;			

28.6.13 LESENSE_STATUS - Status

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																										0x0	0x0	0x0	0x0		0x0	0x0
Access																										R	R	R	R		R	R
Name																										FLUSHING	READBUSY	RUNNING	SCANACTIVE		RESFIFOFULL	RESFIFOV

Bit	Name	Reset	Access	Description
31:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	FLUSHING	0x0	R	**FIFO Flushing**
5	READBUSY	0x0	R	**FIFO Read Busy**
4	RUNNING	0x0	R	**LESENSE periodic counter running**
	LESENSE is running in periodic mode.			
3	SCANACTIVE	0x0	R	**LESENSE scan active**
	LESENSE is currently interfacing sensors.			
2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESFIFOFULL	0x0	R	**Result fifo full**
	Set when the result fifo is full			
0	RESFIFOV	0x0	R	**Result fifo valid**
	Set when data is available in the result fifo. Cleared when the buffer is empty.			

28.6.14 LESENSE_RESCOUNT - Result FIFO Count

Offset	Bit Position																													
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0				
Access																												R				
Name																												COUNT				

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4:0	COUNT	0x0	R	**Result Fifo Count**
	Number of valid data in the Result FIFO			

28.6.15 LESENSE_RESFIFO - Result Fifo

Offset	Bit Position																													
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																			
Access													R																			
Name													BUFDATASRC																			

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:0	BUFDATASRC	0x0	R	**Result data and source**
	This bitfield contains the sensor result (least-significant 16 bits) and the channel index (most-significant 4 bits)			

28.6.16 LESENSE_CURCH - Current Channel Index

Offset	Bit Position																															
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													R		
Name																													CURCH		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	CURCH	0x0	R	**Shows the index of the current channel**

28.6.17 LESENSE_DECSTATE - Current Decoder State

Offset	Bit Position																															
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0		
Access																														R		
Name																														DECSTATE		

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4:0	DECSTATE	0x0	R	**Shows the current decoder state**

28.6.18 LESENSE_SENSORSTATE - Sensor State

Offset	Bit Position																															
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													R			
Name																													SENSORSTATE			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	SENSORSTATE	0x0	R	**Sensor State** Sensor State used as input to the DECODER state machine

28.6.19 LESENSE_IDLECONF - IDLE Configuration

Offset	Bit Position																											
0x048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0	
Access	RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW		RW	
Name	CHIDLE15		CHIDLE14		CHIDLE13		CHIDLE12		CHIDLE11		CHIDLE10		CHIDLE9		CHIDLE8		CHIDLE7		CHIDLE6		CHIDLE5		CHIDLE4		CHIDLE3		CHIDLE2		CHIDLE1		CHIDLE0	

Bit	Name	Reset	Access	Description
31:30	CHIDLE15	0x0	RW	**Channel IDLE configuration**
	Channel 15 idle phase configuration			
	Value	Mode		Description
	0	DISABLE		Channel output is disabled in idle phase
	1	HIGH		Channel output is high in idle phase
	2	LOW		Channel output is low in idle phase
	3	DAC		Channel output is connected to DAC output in idle phase (CH 0,1,2 only)
29:28	CHIDLE14	0x0	RW	**Channel IDLE configuration**
	Channel 14 idle phase configuration			
	Value	Mode		Description
	0	DISABLE		Channel output is disabled in idle phase
	1	HIGH		Channel output is high in idle phase
	2	LOW		Channel output is low in idle phase
	3	DAC		Channel output is connected to DAC output in idle phase (CH 0,1,2 only)
27:26	CHIDLE13	0x0	RW	**Channel IDLE configuration**
	Channel 13 idle phase configuration			
	Value	Mode		Description
	0	DISABLE		Channel output is disabled in idle phase
	1	HIGH		Channel output is high in idle phase
	2	LOW		Channel output is low in idle phase
	3	DAC		Channel output is connected to DAC output in idle phase (CH 0,1,2 only)
25:24	CHIDLE12	0x0	RW	**Channel IDLE configuration**
	Channel 12 idle phase configuration			
	Value	Mode		Description

Bit	Name	Reset	Access	Description
	0	DISABLE		Channel output is disabled in idle phase
	1	HIGH		Channel output is high in idle phase
	2	LOW		Channel output is low in idle phase
	3	DAC		Channel output is connected to DAC output in idle phase (CH 0,1,2 only)
23:22	CHIDLE11	0x0	RW	**Channel IDLE configuration**
	Channel 11 idle phase configuration			
	Value	Mode		Description
	0	DISABLE		Channel output is disabled in idle phase
	1	HIGH		Channel output is high in idle phase
	2	LOW		Channel output is low in idle phase
	3	DAC		Channel output is connected to DAC output in idle phase (CH 0,1,2 only)
21:20	CHIDLE10	0x0	RW	**Channel IDLE configuration**
	Channel 10 idle phase configuration			
	Value	Mode		Description
	0	DISABLE		Channel output is disabled in idle phase
	1	HIGH		Channel output is high in idle phase
	2	LOW		Channel output is low in idle phase
	3	DAC		Channel output is connected to DAC output in idle phase (CH 0,1,2 only)
19:18	CHIDLE9	0x0	RW	**Channel IDLE configuration**
	Channel 9 idle phase configuration			
	Value	Mode		Description
	0	DISABLE		Channel output is disabled in idle phase
	1	HIGH		Channel output is high in idle phase
	2	LOW		Channel output is low in idle phase
	3	DAC		Channel output is connected to DAC output in idle phase (CH 0,1,2 only)
17:16	CHIDLE8	0x0	RW	**Channel IDLE configuration**
	Channel 8 idle phase configuration			
	Value	Mode		Description
	0	DISABLE		Channel output is disabled in idle phase
	1	HIGH		Channel output is high in idle phase
	2	LOW		Channel output is low in idle phase
	3	DAC		Channel output is connected to DAC output in idle phase (CH 0,1,2 only)

Bit	Name	Reset	Access	Description
15:14	CHIDLE7	0x0	RW	**Channel IDLE configuration**
	Channel 7 idle phase configuration			
	Value	Mode		Description
	0	DISABLE		Channel output is disabled in idle phase
	1	HIGH		Channel output is high in idle phase
	2	LOW		Channel output is low in idle phase
	3	DAC		Channel output is connected to DAC output in idle phase (CH 0,1,2 only)
13:12	CHIDLE6	0x0	RW	**Channel IDLE configuration**
	Channel 6 idle phase configuration			
	Value	Mode		Description
	0	DISABLE		Channel output is disabled in idle phase
	1	HIGH		Channel output is high in idle phase
	2	LOW		Channel output is low in idle phase
	3	DAC		Channel output is connected to DAC output in idle phase (CH 0,1,2 only)
11:10	CHIDLE5	0x0	RW	**Channel IDLE configuration**
	Channel 5 idle phase configuration			
	Value	Mode		Description
	0	DISABLE		Channel output is disabled in idle phase
	1	HIGH		Channel output is high in idle phase
	2	LOW		Channel output is low in idle phase
	3	DAC		Channel output is connected to DAC output in idle phase (CH 0,1,2 only)
9:8	CHIDLE4	0x0	RW	**Channel IDLE configuration**
	Channel 4 idle phase configuration			
	Value	Mode		Description
	0	DISABLE		Channel output is disabled in idle phase
	1	HIGH		Channel output is high in idle phase
	2	LOW		Channel output is low in idle phase
	3	DAC		Channel output is connected to DAC output in idle phase (CH 0,1,2 only)
7:6	CHIDLE3	0x0	RW	**Channel IDLE configuration**
	Channel 3 idle phase configuration			
	Value	Mode		Description
	0	DISABLE		Channel output is disabled in idle phase

Bit	Name	Reset	Access	Description
	1	HIGH		Channel output is high in idle phase
	2	LOW		Channel output is low in idle phase
	3	DAC		Channel output is connected to DAC output in idle phase (CH 0,1,2 only)
5:4	CHIDLE2	0x0	RW	**Channel IDLE configuration**
	Channel 2 idle phase configuration			
	Value	Mode		Description
	0	DISABLE		Channel output is disabled in idle phase
	1	HIGH		Channel output is high in idle phase
	2	LOW		Channel output is low in idle phase
	3	DAC		Channel output is connected to DAC output in idle phase (CH 0,1,2 only)
3:2	CHIDLE1	0x0	RW	**Channel IDLE configuration**
	Channel 1 idle phase configuration			
	Value	Mode		Description
	0	DISABLE		Channel output is disabled in idle phase
	1	HIGH		Channel output is high in idle phase
	2	LOW		Channel output is low in idle phase
	3	DAC		Channel output is connected to DAC output in idle phase (CH 0,1,2 only)
1:0	CHIDLE0	0x0	RW	**Channel IDLE configuration**
	Channel 0 idle phase configuration			
	Value	Mode		Description
	0	DISABLE		Channel output is disabled in idle phase
	1	HIGH		Channel output is high in idle phase
	2	LOW		Channel output is low in idle phase
	3	DAC		Channel output is connected to DAC output in idle phase (CH 0,1,2 only)

28.6.20 LESENSE_SYNCBUSY - Synchronization

Offset	Bit Position																														
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																R
Name																																CMD

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	CMD	0x0	R	**Command**
	Command Register Busy			

28.6.21 LESENSE_IF - Interrupt Flags

Offset	Bit Position																															
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access											RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name											RESUF	CNTOF	RESOF	RESWL	DEC	SCANDONE	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	RESUF	0x0	RW	**Result Underflow**
20	CNTOF	0x0	RW	**Counter Overflow**
	LESENSE Ripple Counter Overflows			
19	RESOF	0x0	RW	**Result Overflow**
	Result FIFO Overflow			
18	RESWL	0x0	RW	**Result Watermark Level**
	Set when Result FIFO count reaches RESFIDL			
17	DEC	0x0	RW	**Decoder**
	Set when decoder triggers interrupt			
16	SCANDONE	0x0	RW	**Scan Done**
	Set when scan sequence is completed			
15	CH15	0x0	RW	**Channel**
	Set when Channel 15 triggers			
14	CH14	0x0	RW	**Channel**
	Set when Channel 14 triggers			
13	CH13	0x0	RW	**Channel**
	Set when Channel 13 triggers			
12	CH12	0x0	RW	**Channel**
	Set when Channel 12 triggers			
11	CH11	0x0	RW	**Channel**
	Set when Channel 11 triggers			
10	CH10	0x0	RW	**Channel**
	Set when Channel 10 triggers			
9	CH9	0x0	RW	**Channel**
	Set when Channel 9 triggers			

Bit	Name	Reset	Access	Description
8	CH8	0x0	RW	**Channel**
	Set when Channel 8 triggers			
7	CH7	0x0	RW	**Channel**
	Set when Channel 7 triggers			
6	CH6	0x0	RW	**Channel**
	Set when Channel 6 triggers			
5	CH5	0x0	RW	**Channel**
	Set when Channel 5 triggers			
4	CH4	0x0	RW	**Channel**
	Set when Channel 4 triggers			
3	CH3	0x0	RW	**Channel**
	Set when Channel 3 triggers			
2	CH2	0x0	RW	**Channel**
	Set when Channel 2 triggers			
1	CH1	0x0	RW	**Channel**
	Set when Channel 1 triggers			
0	CH0	0x0	RW	**Channel**
	Set when Channel 0 triggers			

28.6.22 LESENSE_IEN - Interrupt Enables

Offset	Bit Position																															
0x064	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access											RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name											RESUF	CNTOF	RESOF	RESWL	DEC	SCANDONE	CH15	CH14	CH13	CH12	CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	RESUF	0x0	RW	**Result Underflow**
20	CNTOF	0x0	RW	**Counter Overflow**
19	RESOF	0x0	RW	**Result Overflow**
18	RESWL	0x0	RW	**Result Watermark Level**
17	DEC	0x0	RW	**Decoder**
16	SCANDONE	0x0	RW	**Scan Complete**
15	CH15	0x0	RW	**Channel**
	Enable Channel 15 interrupt			
14	CH14	0x0	RW	**Channel**
	Enable Channel 14 interrupt			
13	CH13	0x0	RW	**Channel**
	Enable Channel 13 interrupt			
12	CH12	0x0	RW	**Channel**
	Enable Channel 12 interrupt			
11	CH11	0x0	RW	**Channel**
	Enable Channel 11 interrupt			
10	CH10	0x0	RW	**Channel**
	Enable Channel 10 interrupt			
9	CH9	0x0	RW	**Channel**
	Enable Channel 9 interrupt			
8	CH8	0x0	RW	**Channel**

Bit	Name	Reset	Access	Description
	Enable Channel 8 interrupt			
7	CH7	0x0	RW	**Channel**
	Enable Channel 7 interrupt			
6	CH6	0x0	RW	**Channel**
	Enable Channel 6 interrupt			
5	CH5	0x0	RW	**Channel**
	Enable Channel 5 interrupt			
4	CH4	0x0	RW	**Channel**
	Enable Channel 4 interrupt			
3	CH3	0x0	RW	**Channel**
	Enable Channel 3 interrupt			
2	CH2	0x0	RW	**Channel**
	Enable Channel 2 interrupt			
1	CH1	0x0	RW	**Channel**
	Enable Channel 1 interrupt			
0	CH0	0x0	RW	**Channel**
	Enable Channel 0 interrupt			

28.6.23 LESENSE_CHx_TIMING - Scan Configuration

Offset	Bit Position																													
0x100	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0										0x0								0x0					
Access									RW										RW								RW					
Name									MEASUREDLY										SAMPLEDLY								EXTIME					

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:14	MEASUREDLY	0x0	RW	**Set measure delay**
	Configure measure delay. Sensor measuring is delayed for MEASUREDLY EXCLK cycles.			
13:6	SAMPLEDLY	0x0	RW	**Set sample delay**
	Configure sample delay. Sampling will occur after SAMPLEDLY SAMPLECLK cycles.			
5:0	EXTIME	0x0	RW	**Set excitation time**
	Configure excitation time. Excitation will last EXTIME EXCLK cycles.			

28.6.24 LESENSE_CHx_INTERACT - Scan Configuration

Offset	Bit Position																															
0x104	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset			0x0		0x0				0x0			0x0	0x0	0x0	0x0						0x0											
Access			RW		RW				RW			RW	RW	RW	RW						RW											
Name			SAMPLE		OFFSET				SETIF			EXCLK	SAMPLECLK	ALTEX	EXMODE						THRES											

Bit	Name	Reset	Access	Description
31:30	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
29:28	SAMPLE	0x0	RW	**Sample mode Selection**

Value	Mode	Description
0	ACMPCOUNT	
1	ACMP	
2	ADC	
3	ADCDIFF	

27:24	OFFSET	0x0	RW	**OFFSET for IADC/ACMP interaction**
23:21	SETIF	0x0	RW	**Enable interrupt generation**

Select interrupt generation mode for CHx interrupt flag.

Value	Mode	Description
0	NONE	No interrupt is generated
1	LEVEL	Set interrupt flag if the sensor triggers.
2	POSEDGE	Set interrupt flag on positive edge of the sensor state
3	NEGEDGE	Set interrupt flag on negative edge of the sensor state
4	BOTHEDGES	Set interrupt flag on both edges of the sensor state

20	EXCLK	0x0	RW	**Select clock used for excitation timing**

Value	Mode	Description
0	LFACLK	Prescaled low-frequency LESENSECLK will be used for timing
1	AUXHFRCO	Prescaled high-frequency LESENSEHFCLK will be used for timing

19	SAMPLECLK	0x0	RW	**Select clock used for timing of sample d**

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	LFACLK		Prescaled low-frequency LESENSECLK will be used for timing
	1	AUXHFRCO		Prescaled high-frequency LESENSEHFCLK will be used for timing
18	ALTEX	0x0	RW	**Use alternative excite pin**
	If set, alternative excite pin will be used for excitation			
17:16	EXMODE	0x0	RW	**Set GPIO mode**
	GPIO mode for the excitation phase of the scan sequence. Note that DACOUT is only available on channels 0, 1, 2			
	Value	Mode		Description
	0	DISABLE		Disabled
	1	HIGH		Push Pull, GPIO is driven high
	2	LOW		Push Pull, GPIO is driven low
	3	DACOUT		DAC output
15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:0	THRES	0x0	RW	**ACMP threshold or DAC data**
	Set threshold used for ACMP, or data used in DAC conversion			

28.6.25 LESENSE_CHx_EVALCFG - Scan Configuration

Offset	Bit Position																													
0x108	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0			0x0	0x0		0x0	0x0		
Access																							RW			RW	RW		RW	RW		
Name																							MODE			SCANRESINV	STRSAMPLE		COMP	DECODE		

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	MODE	0x0	RW	**Configure evaluation mode**
	Select which evaluation mode to be used on the measurement result			
	Value	Mode		Description
	0	THRES		Threshold comparison is used to evaluate sensor result
	1	SLIDINGWIN		Sliding window is used to evaluate sensor result
	2	STEPDET		Step detection is used to evaluate sensor result
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	SCANRESINV	0x0	RW	**Enable inversion of result**
	If set, the result is inverted and stored in SCANRES .			
5:4	STRSAMPLE	0x0	RW	**Enable storing of sensor sample in resul**
	If set, the sensor sample value will be stored and available in the result buffer			
	Value	Mode		Description
	0	DISABLE		Nothing will be stored in the result buffer.
	1	DATA		The sensor sample data will be stored in the result buffer.
	2	DATASRC		The data source, i.e. the channel, will be stored alongside the sensor sample data.
3	COMP	0x0	RW	**Select mode for threshold comparison**
	Set compare mode			
	Value	Mode		Description
	0	LESS		Comparison evaluates to 1 if sensor data is less than CTRTHRESHOLD, or if the ACMP output is 0
	1	GE		Comparison evaluates to 1 if sensor data is greater than, or equal to CTRTHRESHOLD, or if the ACMP output is 1
2	DECODE	0x0	RW	**Send result to decoder**

Bit	Name	Reset	Access	Description
	If set, the result from this channel will be shifted into the decoder register.			
1:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

28.6.26 LESENSE_CHx_EVALTHRES - Scan Confguration

Offset	Bit Position																															
0x10C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									EVALTHRES							

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	EVALTHRES	0x0	RW	**Threshold**
	This is the THRESHOLD used for evaluating the sensor results in the different evaluation modes.			

28.6.27 LESENSE_STx_ARC - State Transition Arc

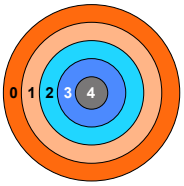
Offset	Bit Position																														
0x200	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0	0x0					0x0			0x0					0x0				0x0			
Access											RW	RW					RW			RW					RW				RW			
Name											SETIF	NEXTSTATE					PRSACT			CURSTATE					SMASK				SCOMP			

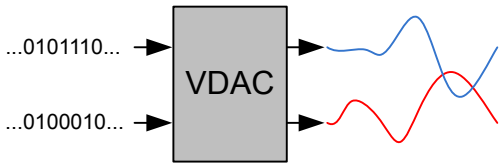
Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	SETIF	0x0	RW	**Set interrupt flag**
	Set interrupt flag when sensor state equals COMP			
20:16	NEXTSTATE	0x0	RW	**Next state index**
	Index of next state to be entered if the sensor state equals COMP			
15:13	PRSACT	0x0	RW	**Configure transition action in normal mode**
	Configure which action to perform when sensor state equals COMP. Note different actions when DECCTRL.PRSCNT = 0 or 1.			

Value	Mode	Description
0	NONE	No PRS output generated (if PRSCOUNT == 0), or do not count (if PRSCOUNT == 1).
1	PRS0	Pulse generated on LESENSE PRS output 0 (if PRSCOUNT == 0).
1	UP	Count Up (if PRSCOUNT == 1).
2	PRS1	Pulse generated on LESENSE PRS output 1 (if PRSCOUNT == 0).
2	DOWN	Count Down (if PRSCOUNT == 1).
3	PRS01	Pulse generated on LESENSE PRS output 0 and 1 (if PRSCOUNT == 0).
4	PRS2	Pulse generated on LESENSE PRS output 2. (PRSCOUNT == 0 OR 1).
5	PRS02	Pulse generated on LESENSE PRS output 0 and 2 (if PRSCOUNT == 0).
5	UPANDPRS2	Count Up and Pulse generated on LESENSE PRS output 2 (if PRSCOUNT == 1).
6	PRS12	Pulse generated on LESENSE PRS output 1 and 2 (if PRSCOUNT == 0).
6	DOWNANDPRS2	Count Down and Pulse generated on LESENSE PRS output 2 (if PRSCOUNT == 1).
7	PRS012	Pulse generated on LESENSE PRS output 0, 1 and 2 (if PRSCOUNT == 0).

Bit	Name	Reset	Access	Description
12:8	CURSTATE	0x0	RW	**Current State**
	Current State Index			
7:4	SMASK	0x0	RW	**Sensor mask**
	Set bit X to exclude sensor X from evaluation.			
3:0	SCOMP	0x0	RW	**Sensor compare value**
	State transition is triggered when sensor state equals COMP			

29. VDAC - Digital to Analog Converter





Quick Facts

What?

The VDAC is designed for low energy consumption, but can also provide very good performance. It can convert digital values to analog signals at up to 500 ksamples/second with 12-bit accuracy.

Why?

The VDAC can be used to generate accurate analog signals for sound, sensors and other applications, using only a limited amount of energy.

How?

The VDAC can generate high-resolution analog signals while the MCU is operating at low frequencies and with low total power consumption. Using the LDMA, the VDAC can be used to generate waveforms without any CPU intervention. The VDAC is available down to Energy Mode 3.

29.1 Introduction

The Voltage Digital to Analog Converter (VDAC) converts a digital value to an analog output voltage. It is useful in a number of different applications such as sensor excitation or low/medium frequency sound output. The VDAC has two rail-to-rail output channels that may act as independent DACs or be combined into a differential pair. The output buffers support high or low power operation as well as a high capacitance mode for driving loads up to 50 pF. Both channels have a 4-word FIFO for efficient throughput and minimum CPU intervention. Flexible conversion trigger sources allow for accurate AC and DC waveform timing, and a special sample-off mode can be used in conjunction with periodic output refresh to reduce energy consumption or act as a temporary excitation source.

29.2 Features

Each VDAC instance in a device supports the following features:

- Up to 500 ksps operation
- Two output channels
  - Can be combined into one differential output
- Integrated 7-bit clock prescaler with division factors ranging from 1 to 128
- Selectable voltage reference
  - Internal 2.5 V (effective full-scale)
  - Internal 1.25 V (effective full-scale)
  - AVDD supply
  - External VREFP pin
- Outputs available for internal and external use
  - Main low-impedance outputs to dedicated GPIO
  - Auxiliary outputs routable through ABUS to any ABUS-capable GPIO (higher impedance)
  - Internal routing of auxiliary outputs to other analog blocks
- Data conversion modes selectable per channel
  - Continuous Mode for high speed conversion or constant DC output
  - Sample-off Mode for per-sample conversion followed by channel disable
- Independent FIFO per channel
  - 4 word (12bit) depth for each channel
  - Programmable data valid level
  - Supports software flush
- Conversion trigger sources
  - Data write (software)
  - PRS input (synchronous and asynchronous)
  - Internal timer with power-of-2 selection from 2-64 prescaled clock cycles
  - LESENSE
- Refresh trigger sources
  - Refresh timer with power-of-2 selection from 2-256 low-frequency clock cycles
  - PRS input (synchronous or asynchronous)
- PRS Communication
  - Separate line for each channel
  - Sync and async PRS output pulse on finished conversion
  - PRS Level Output till channel is warmed
  - Async PRS Output Pulse on Refresh Timer Overflow and Internal Timer Overflow
- LDMA request on FIFO data valid level
  - Independent requests for each DAC channel
- Sine generation mode with differential support

29.3 Functional Description

An overview of the VDAC module is shown Figure 29.1 VDAC Block Overview Diagram on page 1180

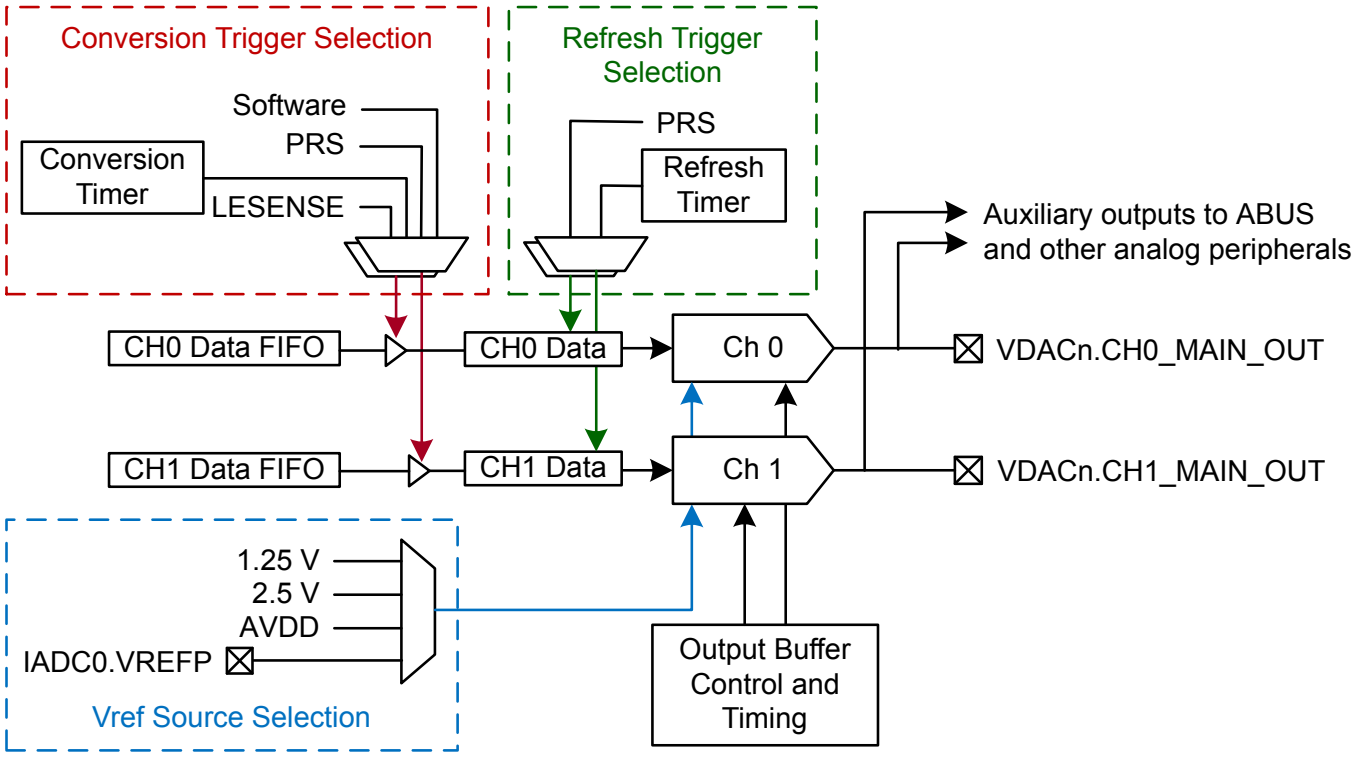


Figure 29.1. VDAC Block Overview Diagram

29.3.1 Power Supply

The VDAC module power (V_{VDAC}) is derived from the AVDD supply pin.

29.3.2 I/O Pin Considerations

The maximum usable analog signal that can be seen on external VDAC outputs depends on both the AVDD and IOVDD supplies. Specifically, the VDAC output will be limited to the lower of the two supply voltages on AVDD and IOVDD.

29.3.3 Enabling and Disabling a Channel

The VDAC module is enabled by writing 1 to EN in VDAC_EN.

A VDAC channel is enabled by writing 1 to the CHxEN and disabled by writing 1 to CHxDIS using in the CMD register. The channel status (enabled or disabled) can be read using the CHxENS bits in the STATUS register. The CHxENS bit will go high after a synchronization delay following a write to CHxEN. When disabling a channel the CHxENS bit will stay high until the VDAC channel is completely disabled.

Software should configure the VDAC before enabling a channel. The following registers are used to configure all the available features of the VDAC:

- VDAC_CFG
- VDAC_CH0CFG
- VDAC_CH1CFG
- VDAC_OUTTIMERCFG

A VDAC channel will not begin driving its output before it is enabled *and* has received a conversion trigger or refresh trigger. After a channel is enabled, it will listen for either conversion trigger sources specified in TRIGMODE in VDAC_CHxCFG or refresh trigger sources specified in REFRESHSOURCE in VDAC_CHxCFG. If TRIGMODE is set to SW and the CHxF FIFO is not empty, a conversion will start immediately when the channel is enabled. When disabling a channel, any pending triggers are flushed.

When disabling the VDAC module, user code must poll the status bit VDAC_EN.DISABLING to ensure that the module is cleanly reset and back to its initial condition.

29.3.4 Clock Selection

The VDAC logic accepts three clock sources from the CMU: LSPCLK, VDACn_CLK, and VDACn_REFRESH_CLK. The APB register interface and FIFO write logic are clocked from the LSPCLK. The rest of the VDAC state machine is clocked mainly by a prescaled version of VDACn_CLK. VDACn_DAC from the CMU can be up to 80 MHz. The PRESC bit field in the CFG register should be set to scale VDACn_CLK to no more than 1 MHz. The VDACn_REFRESH_CLK is a low-frequency clock source which only clocks the dedicated refresh timer.

The clock request for VDACn_CLK to the CMU from VDAC is on-demand by default. This means the VDAC core clock is gated off most of the time except:

- New Conversion or Refresh Trigger
- Sine Generation Active Window
- VDAC_CHxCFG.TRIGMODE = SYNCPRS
- VDAC_CHxCFG.REFRESHSOURCE = SYNCPRS
- VDAC_CHxCFG.TRIGMODE = SW and the VDAC CHx FIFO is not empty
- VDAC_CHxCFG.TRIGMODE = INTTIMER

In some cases it is necessary or preferred to have the VDAC clock active all the time. To turn off on-demand clocking, the VDAC_CFG.ONDEMANDCLK bit can be set to '1', which always requests VDACn_CLK from the CMU. On-demand clocking should be disabled if EM23GRPACLK is selected for VDACn_CLK.

If the VDAC will only perform conversions in EM0 and EM1, any clock source for the VDAC may be used.

If the VDAC is to be operated in EM2 or EM3, VDACn_CLK must be configured to use either HFRCOEM23, EM23GRPACLK or FSRCO instead of the EM01GRPACLK clock. FSRCO is the fast start oscillator which starts quickly but is not as accurate as the other oscillators. HFRCOEM23 is generally recommended for EM2/EM3 operation. When using FSRCO or HFRCOEM23, the clock source can be selected to be "on demand" so as not to waste current when the DAC is not doing a conversion. On demand clocking is configured by setting VDAC_CFG.ONDEMANDCLK to 0.

EM23GRPACLK is recommended only when VDAC is expected to do very slow sample conversions/refresh. VDAC_CFG.ONDEMANDCLK should be set to 1 if EM23GRPACLK is selected as the VDAC clock source.

Note: When HFRCOEM23 is selected as a clock source to VDAC and clocking is on-demand (CFG.ONDEMANDCLK = 0), HFRCOEM23 on-demand clocking must be enabled. This allows the power domain which powers HFRCOEM23 to be ON during EM2 and the clock request to HFRCOEM23 can be honored.

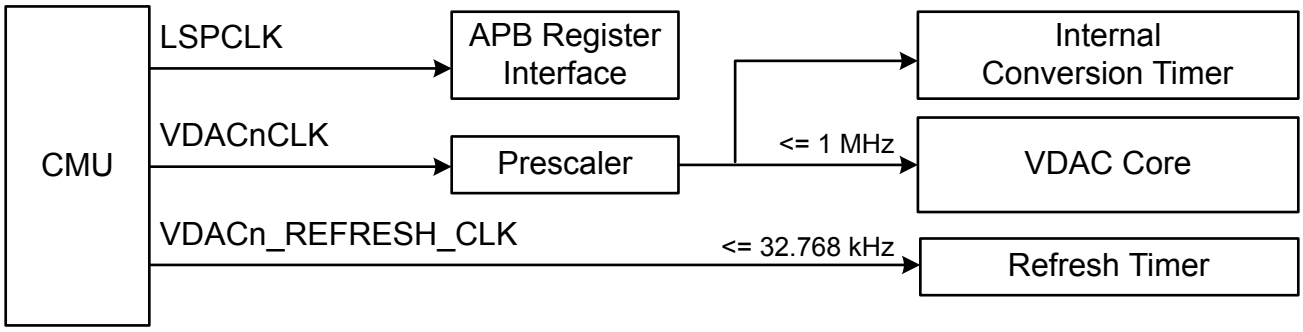


Figure 29.2. VDAC Block Clocks Diagram

29.3.4.1 Internal Clock Prescaler

The VDAC has an internal clock prescaler, which can divide the VDACn_CLK input clock by any factor between 1 and 128, by setting the PRESC field in the CFG register. The resulting prescaled clock is 50% duty cycle and is used by the converter core. The frequency is given by Figure 29.3 VDAC Clock Prescaling on page 1183 :

$$f_{CLK\_DAC\_PRESC} = f_{VDACn\_CLK} / (PRESC + 1)$$

Figure 29.3. VDAC Clock Prescaling

where f_{VDACn_CLK} is the input clock frequency coming from the CMU. The $f_{CLK\_DAC\_PRESC}$ must be programmed to be at most 1 MHz.

The prescaler runs continuously when the VDAC is enabled or when VDAC_SWRST.SWRST is set. When running with a prescaler setting higher than 0, there can be an unpredictable delay from the time a conversion was triggered to the time the actual conversion takes place. This is because the conversions are controlled by the prescaled time base, and the conversion can arrive at any time during a prescaled clock (CLK_DAC_PRESC) period. A second reason for unpredictable delay between a trigger and the associated conversion is that the activity on one channel can impact whether the VDAC reference is warm or not - when two channels are used independently with warmup, it can impact whether a warmup is required on a trigger.

The uncertainty related to the clock prescaler can be addressed by using CH0PRESCRST. If the CH0PRESCRST bit in VDAC_CFG is set, the prescaler will be reset every time a conversion is triggered on channel 0. This leads to a predictable latency between channel 0 triggers and conversions (assuming the warmup sequence is deterministic as well). If channel x is used in continuous mode, the warmup sequence will only apply once when the channel is enabled and software can use the VDAC_STATUS.CHxWARM bit to determine if the VDAC has warmed up.

29.3.4.2 VDACn_REFRESH_CLK

VDAC also has another clock coming in from the CMU, VDACn_REFRESH_CLK. This clock is asynchronous to the VDACn_CLK and is used to clock the internal refresh timer. VDACn_REFRESH_CLK is connected directly to the EM23GRPACLK source in the CMU and hence it is a low-frequency clock sourse with a maximum frequency of ~32 kHz. The refresh timer inside VDAC is a slow running, low power timer. When this refresh timer is used as a refresh trigger source and the timer overflows, it will trigger a VDACn_CLK on-demand request from the CMU.

29.3.5 Conversions

The VDAC consists of two channels (channel 0 and 1) with separate 4 deep FIFOs with 12-bits data elements (VDAC_CHxF.DATA). These can be used to produce two independent single ended outputs or the channel 0 register can be used to drive both outputs in a differential mode. The VDAC supports two conversion modes: **continuous** and **sample-off.**

Continuous Mode

In continuous mode the VDAC buffers will remain on, and channels will drive their outputs continuously till the channel is disabled with the data in the VDAC_CHxF.DATA register. A channel is configured in continuous mode by programming the CONVMODE bitfield in VDAC_CHxCFG to CONTINUOUS. This mode will maintain the output voltage from a conversion indefinitely, until a new output is triggered or until the channel is disabled.

In continuous mode the CHxOUTHOLDTIME field in VDAC_OUTTIMERCFG should be programmed to zero to achieve the maximum update rate. Both these settings need to be configured before VDAC module is enabled.

Sample-off Mode

In sample-off mode the VDAC will only drive the output for a limited time per conversion. A channel is configured in sample-off mode by programming the CONVMODE bitfield in VDAC_CHxCFG to SAMPLEOFF. The CHxOUTHOLDTIME field in the VDAC_OUTTIMERCFG register determines how long the output will be driven after a conversion or refresh trigger occurs. The VDAC will drive the output for CHxOUTHOLDTIME number of CLK_DAC_PRESC cycles before tri-stating the output again (and therefore if CHxOUTHOLDTIME is set to zero, the output will never be driven when using sample-off mode). Both these settings need to be configured before VDAC module is enabled.

29.3.6 Conversion Trigger

Conversions can only be performed while a channel is enabled and the CHxF FIFO is not empty, see 29.3.3 Enabling and Disabling a Channel.

- If CHxCFG.TRIGMODE is programmed to SW, a conversion can be started automatically when the CHxF.DATA is not empty. Writing to the FIFO will trigger a conversion on the specified channel.
- If CHxCFG.TRIGMODE is programmed to SYNCPRS or ASYNCPRS, a conversion can be started by an incoming pulse on the selected PRS channel. VDAC expects a PRS pulse coming from both synchronous and asynchronous PRS producers. Depending on the TRIGMODE and channel enable, the VDAC will process the PRS pulse from the respective producer.
- If CHxCFG.TRIGMODE is programmed to INTERNALTIMER, the internal timer is uses to start conversions, and a new conversion will start on any overflow of the internal timer. See 29.3.13 Internal Timers.
- For Channel 0, If CH0CFG.TRIGMODE is programmed to LESENSE a conversion will start when the LESENSE block sends a request. This setting needs to be selected whenever the channel 0 is under LESENSE control. Note that LESENSE can only trigger a conversion to Channel 0 and not Channel 1.

29.3.7 Refresh Trigger

The refresh mechanism can be used to periodically refresh the output with the most recent conversion result. A refresh trigger can only happen while a channel is enabled, see 29.3.3 Enabling and Disabling a Channel.

- If CHxCFG.REFRESHSOURCE is programmed to SYNCPRS or ASYNCPRS, a refresh can be started by an incoming pulse on the selected PRS channel. The VDAC refresh mechanism requires a PRS pulse for either configuration of CHxCFG.REFRESHSOURCE. Depending on the REFRESHSOURCE and channel enable, VDAC processes the PRS pulse from the respective producer.
- If CHxCFG.REFRESHSOURCE is programmed to REFRESHTIMER, a conversion will start on an overflow of the refresh timer. See 29.3.13 Internal Timers.

Note: The refresh mechanism never pops a new conversion from the FIFO. It just re-triggers the conversion based on the last data converted by the conversion trigger to generate the same voltage output at the load. A periodic refresh of the output voltage into a suitable RC filter (to reduce ripple) may be a way to establish a low-energy DC bias in some applications.

Conversion Trigger and Refresh Trigger co-existence

- Conversion triggers are given preference over refresh triggers. During a conversion trigger sample conversion, if a refresh trigger occurs it is ignored.
- Refresh triggers are served only when the conversion trigger sample conversion is completely finished.
- During a refresh trigger sample conversion, if a conversion trigger occurs, it is held by the VDAC and served once the refresh trigger conversion is finished. The VDAC never ignores conversion triggers.

29.3.8 PRS Communication

PRS triggers can be used to set a constant sample frequency, for instance by using a TIMER, or to synchronize conversion events with other hardware. In order to get a jitter-free sample rate from a PRS source, set CHxCFG.TRIGMODE to SYNCPRS and set the CFG.CH0PRESCRST bit to ensure the prescaler is reset on trigger. Note that because the prescaler is shared between channels, this is only possible for channel 0.

For CHxCFG.TRIGMODE / CHxCFG.REFRESHSOURCE of ASYNCPRS, the sample frequency cannot be guaranteed to be jitter-free with respect to the PRS pulses. The CH0PRESCRST bit in VDAC_CFG can still be set to reset the clock prescaler on every PRS trigger for better predictability. Note, this can be set only on channel 0.

The Input PRS frequency should never be higher than 0.5 MHz (the fastest possible sample rate). In addition, the input PRS frequency should not be higher than f_{VDACn_CLK} / 4 (in synchronous mode). If the PRS frequency is set too high, some PRS pulses may be dropped and the output can jitter.

VDAC also sends out the following list of output signals as a PRS producer:

- CHxDONEASYNC - CHx Conversion Done Asynchronous PRS Pulse Output
- CHxWARM - CHx Output Valid Async PRS Level Output
- CHxDONESYNC - CHx Conversion Done Sync PRS Pulse Output
- REFRESHTIMEROF - Refresh Timer Overflow Async PRS Pulse Output
- INTERNALTIMEROF - Internal Timer Overflow Async PRS Pulse Output

29.3.9 Reference Selection

The VDAC supports four voltage reference options:

- Internal 1.25 V Reference (effective full scale)
- Internal 2.5 V Reference (effective full scale)
- AVDD
- External IADC0.VREFP Pin

The voltage reference is selected by programming the REFRSEL field in VDAC_CFG, and is shared for both VDAC channels. The selected voltage reference sets the full scale output voltage of the VDAC. In the case of the internal 1.25 V and 2.5 V reference options, the VDAC uses a lower internal voltage as the reference, and scaling techniques to produce an effective full scale voltage of 1.25 V or 2.5 V respectively. The 2.5 V internal reference can still be used over the entire supply voltage range of the device without any dropout. However, the output will be limited by the supply rail(s) and VDAC will not be able to drive above the supply.

29.3.10 Power Modes

The VDAC supports independently-selectable high power and low power modes per channel. Each has a maximum load capacitance of 50 pF. The power mode for each channel is configured by setting VDAC_CHxCFG.POWERMODE. By default, the VDAC starts in HIGHPOWER mode.

VDAC also supports an additional high capacitance load mode in conjunction with the HIGHPOWER power mode. High capacitance mode is enabled by setting VDAC_CHxCFG.HIGHCAPLOADEN to 1. The minimum load capacitance on the pin is 25 nF for this mode.

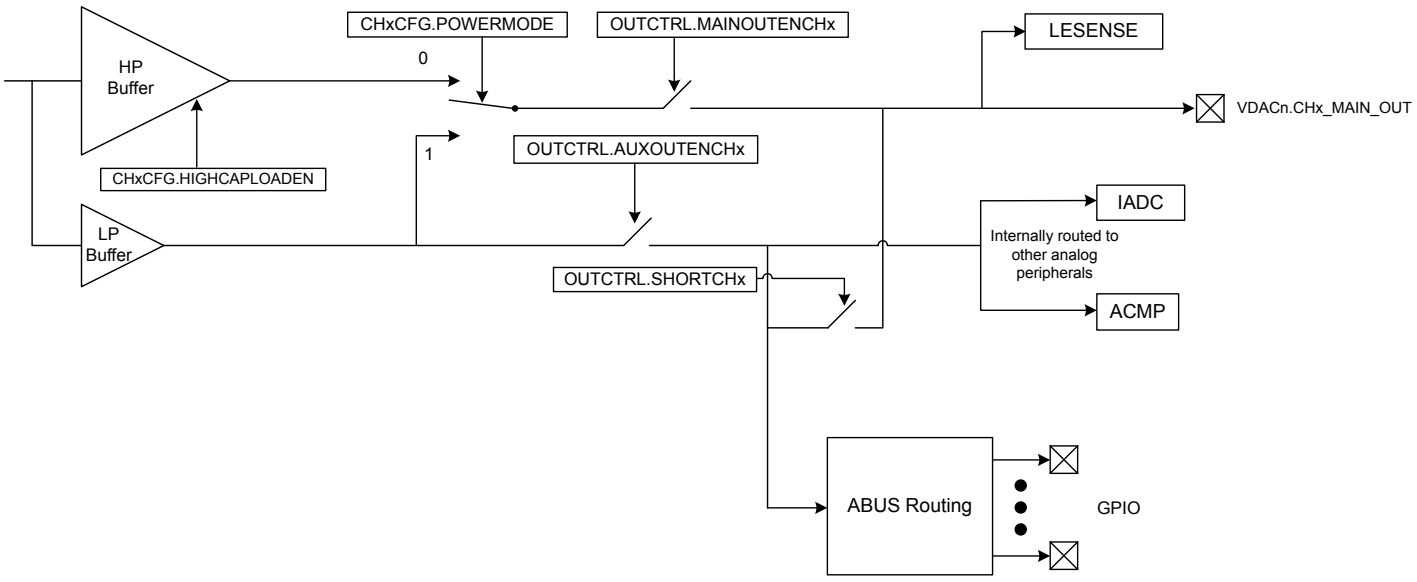


Figure 29.4. VDAC Block Power Modes Diagram

29.3.11 Warmup Time and Initial Conversion

When a channel is enabled, it needs to warm up. This is performed automatically when the channel is enabled if the CONVMODE in VDAC_CHxCFG is set to Continuous mode.

If the CONVMODE in VDAC_CHxCFG is set to Sample-off mode, then every sample conversion is preceded by warmup of the VDAC. Hence, the sample conversion time in case of sample-off mode includes the WARMUPTIME of the VDAC too.

The time allocated for warmup depends on the WARMUPTIME field in VDAC_CFG. WARMUPTIME is specified in prescaled VDACn_CLK cycles and should be set to a minimum of 3 us (3 clocks if CLK_DAC_PRESC = 1 MHz). Software is responsible for programming the correct value to WARMUPTIME before enabling a channel. If the time is programmed too short, an undefined voltage may be output until the analog portion of VDAC settles.

The CHxWARM bits in VDAC_STATUS are set when the warmup period has completed. A consequence of the warmup period is that in continuous mode, the recommended programming model is to either wait for CHxWARM status before starting conversions in order to make sure all samples have the same timing, or perform a dummy conversion to make the VDAC settle to a known voltage first.

29.3.12 Output Mode

The two VDAC channels can act as two separate single-ended channels or be combined into one differential channel. This is selected through the DIFF bit in VDAC_CFG.

- **Single Ended Output**

When operating in single ended mode, the channel 0 output is on VDAC_OUT0 and the channel 1 output is on VDAC_OUT1. The output voltage can be calculated using Figure 29.5 VDAC Single Ended Output Voltage on page 1186

$$V_{OUT} = V_{VDACn\_OUTx} - V_{SS} = V_{ref} \times CHxDATA/4096$$

Figure 29.5. VDAC Single Ended Output Voltage

where CHxDATA is a 12-bit unsigned integer.

- **Differential Output**

When operating in differential mode, both VDAC outputs are used to produce a differential output. The conversion uses the VDAC_CH0DATA buffer as the data source. The data written to VDAC_CH0DATA is interpreted as a two's complement number, with the MSB of the 12-bit value being the sign bit. Conceptually, a 'positive' and 'negative' output, with a common-mode voltage of $V_{ref}/2$ are generated from this information.

The positive output appears on VDAC_CH1 and the negative output appears on VDAC_CH0. The output voltage can be calculated using Figure 29.6 VDAC Differential Output Voltage on page 1186:

$$V_{OUT} = V_{VDACn\_CH1} - V_{VDACn\_CH0} = V_{ref} \times CH0DATA/2048$$

Figure 29.6. VDAC Differential Output Voltage

where CH0DATA is a 12-bit signed integer. The common mode voltage is $V_{ref}/2$.

When using differential mode, the user needs to program only CH0 settings to reflect to both channels. Any programmed settings in CH1 will be ignored in differential mode.

29.3.13 Internal Timers

VDAC has two internal timers that run independently of each other:

Conversion Timer

VDAC includes an internal conversion timer.This conversion timer is automatically started if a channel selects INTERNALTIMER as TRIGMODE in VDAC_CHxCFG reigster and the channel is enabled. The conversion timer will count the number of prescaled VDACn_CLK cycles programmed in VDAC_CFG.TIMEROVERFLOWPERIOD before wrapping and generating a conversion trigger. The timer overflow period is from 2-64 prescaled VDACn_CLK cycles. The conversion timer overflow automatically initiates the sample conversion if the CHxF FIFO is not empty.

INTERNALTIMEROF is also available as an asynchronous PRS signal, generating a pulse that lasts for 1 prescaled VDACn_CLK cycle. Note that since this timer runs off prescaled VDACn_CLK, the VDACn_CLK is always requested from the CMU when the internal conversion timer is used.

Refresh Timer

VDAC includes an internal low power refresh timer.This refresh timer is automatically started if a channel selects REFRESHTIMER as REFRESHSOURCE in VDAC_CHxCFG register and the channel is enabled. The refresh timer will count the number of VDACn_REFRESH_CLK cycles programmed in VDC_CFG.REFRESHPERIOD before wrapping and generating a refresh trigger. The refresh period is from 2-256 VDACn_REFRESH_CLK cycles. The refresh timer overflow automatically initiates the refresh.

REFRESHTIMEROF is also available as an asynchronous PRS signal, generating a pulse that lasts for 1 VDACn_REFRESH_CLK cycle. Note that since this timer runs off VDACn_REFRESH_CLK, which is asynchronous to VDACn_CLK, VDACn_CLK is only requested from the CMU when the refresh timer overflows. Also, as discussed in 29.3.4 Clock Selection, this timer always runs off a slow clock, hence this refresh conversion is low power.

29.3.14 FIFO

VDAC has two FIFOs, one for each channel. Each FIFO is 4 samples deep. Data is pushed into the FIFO by either the CPU or LDMA, and happens on the bus clock. Samples are popped from the FIFO on the VDACn_CLK domain by the VDAC core whenever there is a new conversion trigger. Both FIFOs support a data flush from the CPU.

CHxF FIFO Programming Model

Before enabling VDAC, set the low threshold Data Valid Level for each channel by programming VDAC_CHxCFG.FIFODVL Bit. If in DIFF Mode, the channel 1 DVL settings are ignored. Fill the VDAC_CHxF.DATA with the number of entries greater than programmed DVL to avoid triggering of DVL Interrupt. Details of the DVL Interrupt are explained in 29.3.18 Interrupts and Wakeup. The number of valid entries per channel can be read from VDAC_STATUS.CHxFIFOCNT (only if VDAC_CFG.ONDEMANDCLK is set to 1). The status of whether the FIFO is full or empty can be read from VDAC_STATUS.CHxFIFOFULL and VDAC_STATUS.CHxFIFOEMPTY respectively.

CHxF FIFO Flush Programming Model

In case there is incorrect data programmed in the FIFO or in the event of a FIFO overflow, the CPU can issue a flush by programming VDAC_CMD.CHxFIFOFLUSH. This flush bit resets both the write pointer (in the bus clock domain) and the read pointer (in rhe CLK_DAC domain) together. During a flush, new conversions should not be triggered on the channel whose FIFO contents are flushed. The flush status is reported with the VDAC_STATUS.CHxFIFOFLBUSY bit, which will remain high during the flush operation and return low when it is complete. Since the CPU issues flush to the CHx FIFO, there are several steps that should be followed in order to avoid any spurious voltage at VDAC Output:

- Disable the channel using VDAC_CMD.CHxDIS = 1
- Disable the VDAC channel in the LDMA to avoid any new CHxFIFO writes during flushing
- Issue the flush command and poll the VDAC_STATUS.CHxFIFOFLBUSY status bit
- After flushing is complete, re-enable the VDAC channel and optionally re-program the LDMA

29.3.15 Keepwarm Sub-modes

VDAC has two keepwarm modes that can be used with Sample-off Conversion Mode to define the behaviour of the analog portion of the VDAC between sample conversions. The keepwarm mode needs to be set along with Sample-off Conversion Mode before enabling the VDAC module. The two options are:

Bias Keepwarm

Bias Keepwarm Mode is primarily used to reduce kickback to the reference bias, in case it is shared with IADC. This mode can be activated by setting the VDAC_CFG.BIASKEEPWARM Bit to 1. During this mode, the VDAC Analog Bias remains ON in between sample conversions instead of shutting down after the hold time expires.

This mode is only relevant when using the internal 1.25 V or 2.5 V reference selections. This mode does not reduce the required warm-up time of the VDAC. Enabling this mode will typically cost about 4 uA of additional current when the VDAC is idle. The VDAC analog bias remains on irrespective of the channel enable/disable, until the VDAC module is disabled.

Channel Keepwarm

Channel Keepwarm Mode is primarily used to reduce kickback between the two channels, or to reduce the start-up time of the VDAC in-between sample conversions. This mode is activated per channel by setting the VDAC_CHxCFG.KEEPWARM bit to 1. During this mode, the VDAC remains warmed up in-between sample conversions and can convert new samples without incurring a warm up delay. This mode can only be used with Sample-off conversion mode. Setting this mode typically costs about 10-20 uA of additional current when the VDAC is idle.

29.3.16 LDMA Interface

The VDAC has two FIFOs (one per channel) which can be filled using the LDMA whenever there is space available in the FIFO. To facilitate this, the VDAC generates a DMA request per channel to the LDMA when the FIFO count reaches the lower threshold Data Valid Level (DVL). DVL is set by programming VDAC_CHxCFG.FIFODVL to set the watermark.

DMA REQ and SREQ

Both Channel 0 and 1 generate DMA requests in the bus clock domain only when the VDAC_CMD.CHxEN bit is set. In the case of DIFF mode, Channel 1 will not generate LDMA requests. The FIFOs are initially empty, hence as soon as the channels are enabled, each channel will send out a DMA REQ. The request is cleared when FIFO is filled beyond DVL+1.

If there is at least 1 valid space in the FIFO and the DMA channel is not active, each VDAC channel will also send a SREQ to LDMA. This SREQ is also gated with the channel enable.

EM2 Operation

When the system is in EM2, the VDAC can generate new DMA requests as well. This feature can be enabled by setting the VDAC_CFG.DMAWU bit to 1 before enabling VDAC. When enabled and the FIFO count in the read domain falls below the programmed DVL setting, the VDAC will generate a request to the EMU to enter an EM1 state. Once the EMU enters this state and the bus clock is available, VDAC generates the DMA REQ/SREQ per channel to request new data for conversion.

The VDAC keeps the request for data high until the FIFO count is above the programmed DVL setting. Once the condition is met, VDAC automatically pulls the request low and the system can return fully to EM2.

29.3.17 Sine Generation Mode

The VDAC contains an automatic sine-generation mode, which is enabled by setting the SINEMODE bit in VDAC_CFG. In this mode, the VDAC data from the FIFO is overridden with data from an internal hard-coded sine lookup table.

Sine mode is supported only for the fastest configuration of the VDAC in continuous mode. Therefore the CONVMODE bit in VDAC_CH0CFG needs to be set to CONTINUOUS and the CH0OUTHOLDTIME bit in VDAC_OUTTIMERCFG needs to be programmed to zero for channel 0 to use sine generation mode. The TRIGMODE and REFRESHSOURCE bitfields in VDAC_CHxCFG need to be programmed to NONE for channel 0 in order to avoid interference in sine output generation from other triggers. Other trigger modes are not supported. The SINE wave will always be output on channel 0 and therefore requires that this channel is enabled by writing 1 to CH0EN in the VDAC_CMD register. If DIFF is set in VDAC_CFG, the sine wave will additionally be output on channel 1, but inverted. Note that the first sample will only be available after the CHxWARMED bit is 1 in VDAC_STATUS register after setting CH0EN =1 in VDAC_CMD.

Each period, starting at 0 degrees, is made up of 16 samples and the frequency is given by Figure 29.7 VDAC Sine Generation Frequency on page 1188.

$$f_{sine} = f_{CLK\_DAC\_PRESC} / 32$$

Figure 29.7. VDAC Sine Generation Frequency

When DIFF is 0 and SINEMODE is 1 in VDAC_CFG, a sine wave will be generated on channel 0 but channel 1 can still be used independently as a single-ended DAC output. Channel 1 settings are ignored if DIFF is 1.

To configure the VDAC for sine wave output:

- Set VDAC_CFG.SINEMODE to 1 to put the VDAC in sine wave mode
- Set VDAC_CH0CFG.CONVMODE to CONTINUOUS
- Set VDAC_CH0CFG.TRIGMODE to NONE
- Set VDAC_CH0CFG.REFRESHSOURCE to NONE
- Set VDAC_OUTTIMERCFG.CH0OUTHOLDTIME to 0
- Configure VDAC_CFG.DIFF for single-ended or differential output
  - 0 = Single ended output (only channel 0 is used)
  - 1 = Enable differential output (both channel 0 and channel 1 will be used)
- Configure VDAC_CFG.SINERESET for desired behavior when halted
  - 0 = Sine wave output is not reset when halted, and will continue at the next sample when re-started
  - 1 = Sine wave output is reset to 0 degrees when halted, and output will return to Vref / 2
- Set VDAC_CFG.CH0EN to 1 to enable the VDAC module

29.3.17.1 Sine Generation - Software Control

The sine signal generation may be controlled by software using the SINEMODESTART and SINEMODESTOP commands in VDAC_CMD. When SINEMODESTART is set to 1, sine wave generation will be started. Setting SINEMODESTOP to 1 will halt the sine wave generation. If SINERESET in VDAC_CFG is set to 1, the sine output will be reset to 0 degrees when the SINEMODESTOP bit is set to 1, resulting in a voltage of Vref / 2 on the output channel(s). If SINERESET equals 0, a SINEMODESTOP command will stop progress of the sine wave at the sample currently being output. The sine will continue at the next sample when SINEMODESTART is issued again.

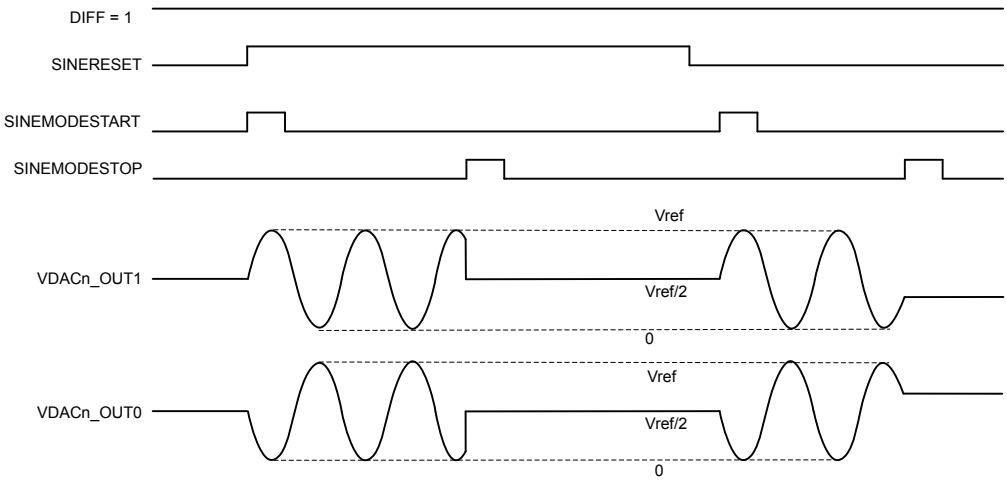


Figure 29.8. VDAC Sine Mode - Software Control

29.3.17.2 Sine Generation - PRS Control

The sine signal generation may alternatively be controlled by the VDAC's two PRS channels. Functionality and features are similar to the software-controlled mode. One PRS channel controls start and stop behavior, and the other can optionally control the DAC buffer output enable.

To enable PRS control of the sine generation, set the SINEMODEPRS bit in VDAC_CFG to 1. In the PRS registers, PRS_CONSUMER_VDACn_ASYNCTRIGCH0 and PRS_CONSUMER_VDACn_ASYNCTRIGCH1 should be configured to select the desired PRS channels that will control the VDAC operation.

The PRS channel selected for ASYNCTRIGCH0 is used to start and stop the sine wave output. When this PRS channel is 1, the sine wave generation will be active. The PRS channel selected for ASYNCTRIGCH1 may optionally be used to enable or disable the buffer outputs of both DAC channels. To enable buffer output control, set the OUTENPRS bit in VDAC_CFG to 1. When OUTENPRS is enabled, a logic 1 on ASYNCTRIGCH1 will enable the buffer output, and logic 0 on ASYNCTRIGCH1 will disable the output buffer and the output will go to a high-impedance state. If ASYNCTRIGCH0 is still active (high) while the output buffer is disabled, the digital sine wave table will continue to be piped into the VDAC even though the output at the pin has been disabled. If OUTENPRS in VDAC_CFG is logic 0 (disabled), the PRS channel associated with ASYNCTRIGCH1 will have no impact on the buffer output.

Note: When OUTENPRS is set to 1, ASYNCTRIGCH1 will control the buffer output on **both** DAC channels regardless of the setting of DIFF.

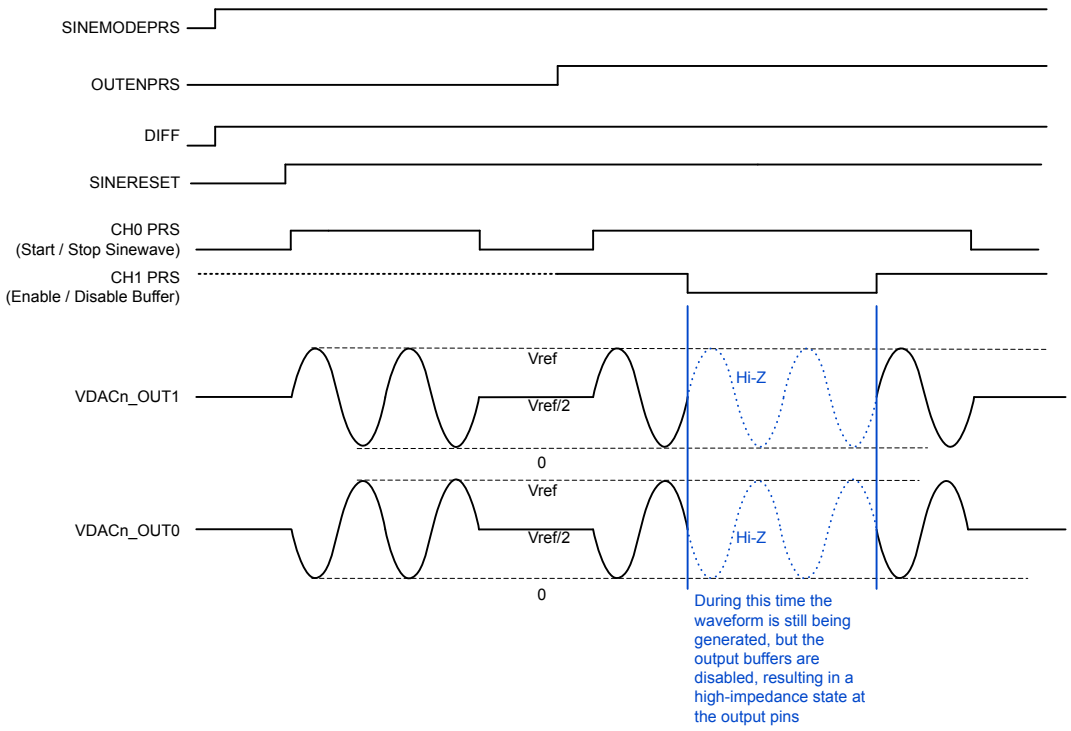


Figure 29.9. VDAC Sine Mode - PRS Control

29.3.18 Interrupts and Wakeup

The VDAC has several interrupt flags in the VDAC_IF register, indicating various state and error conditions related to output, FIFO status, and conversions. Each channel has a separate set of interrupt flags.

- **Conversion Done**

  The Conversion Done (CHxCD) interrupt flags are set when a conversion is complete. The flags are set after a channel has driven the output with the programmed code.
- **Data Valid Level (DVL)**

  The Data Valid Level (CHxDVL) interrupt flags are set when the FIFO Count reaches below or equal to Data Valid Level (DVL). The DVL number is programmed in CHxCFG.DVL Register Bits. These flags are initially set, and gets cleared when CHxF FIFO is filled with new set of data, thereby making count >= DVL. The FIFO Count can be read through VDAC_STATUS.CHxFIFOCNT Flag. Note that this fifo count is generated in the read domain, hence expect read synchronization delay for the count value.
- **Overflow/Underflow**

  If a write is attempted in CHxF FIFO whilst it is full, the channel overflow flag (CHxOF) will be set. If a new conversion is triggered (e.g. via PRS) before whilst CHxF FIFO is empty, the channel underflow flag (CHxUF) will be set. The FIFO full and empty status can be read through VDAC_STATUS.CHxFIFOFULL and VDAC_STATUS.CHxFIFOEMPTY Flags.
- **ABUS Conflict/Allocation Error**

  In case both channel 0 and channel 1 request the same Port and Pin for conversion output through ABUS, an ABUS Conflict Error IF (ABUSINPUTCONFLICT) will be set. In case the ABUS Allocation is not coherent with Programmed Port and Pin Setting for either channel, ABUS Allocation Error interrupt flag (ABUSALLOCERR) will be set.

Not all interrupt flags will wake up EMU to EM0 when the VDAC is operating in EM2. Only those interrupts that need interaction with the host will wakeup the CPU. For this, DVL, ABUSALLOCERR and/or ABUSINPUTCONFLICT are used as wakeup interrupt source too.

29.3.19 LESENSE Operation

VDAC Channel 0 can be controlled by LESENSE by programming the TRIGMODE field in VDAC_CH0CFG to LESENSE. In LESENSE mode the conversion data can come from either the VDAC_CHx0 FIFO or LESENSE registers, depending on the LESENSE configuration. The trigger events are also controlled by the LESENSE state machine.

When used for LESENSE, the MAIN output must be enabled and it will be driven to the main output pin. LESENSE may switch this output to connect to one of its defined DAC output pins as needed.

All the programmed VDAC settings (including the Keepwarm Modes) will be over-written by LESENSE configuration. See the LESENSE chapter for more information.

29.3.20 VDAC Output Configuration

The VDAC analog outputs can be routed either to specific fixed pins, to configurable GPIO through ABUS, or used internally by other blocks. These settings can be programmed in the VDAC_OUTCTRL register after VDAC is enabled but ideally before channels are enabled. The possible settings are as follows:

- VDAC_OUTCTRL.MAINOUTENCHx - Set this bit to route VDAC channel analog main output to the dedicated pin (CHx_MAIN_OUT). This is the preferred option for any DAC output. This option is also required when using VDAC CH0 in conjunction with the LESENSE switched outputs.
- VDAC_OUTCTRL.AUXOUTENCHx - Set this bit to route VDAC channel analog auxiliary output to internal blocks such as IADC and ACMP, or to the ABUS interconnect matrix, where it can be routed to any port I/O supporting ABUS. Note that the ABUS multiplexer adds significant impedance, and this option may not be suitable for certain loads or dynamic conditions.
- VDAC_OUTCTRL.SHORTCHx - Set this bit when using the high-power buffer option with the auxiliary outputs. This will short the MAIN and AUX outputs together. The MAIN output is still driven in this configuration.
- VDAC_OUTCTRL.ABUSPORTSELCHx - Select a particular ABUS port for the channel. Choose values from PORTA, PORTB, PORTC, PORTD
- VDAC_OUTCTRL.ABUSPINSELCHx - Select a particular ABUS pin for the channel. Program the pin in conjuction with the port to route the analog output to a particular GPIO.

Note: Once the channel is enabled, the output control settings in VDAC and the ABUS configuration in both VDAC and GPIO should not be changed until the channel is disabled. Changing these settings while enabled may cause spurious voltage generation on random GPIO.

ABUS Allocation Rules to VDAC in GPIO

- AEVEN0/BEVEN0/CDEVEN0 can only be allocated to CH0
- AEVEN1/BEVEN1/CDEVEN1 can only be allocated to CH1
- AODD0/BODD0/CDODD0 can only be allocated to CH0
- AODD1/BODD1/CDODD1 can only be allocated to CH1
- The port and pin requested by ABUSPORTSELCHx/ABUSPINSELCHx must match a bus allocated to VDAC CHx.

  For example, if AEVEN0 is the only bus allocated to CH0, the selected port must be port A, and the selected pin must be even.

More details on ABUS allocation can be found in GPIO chapter

29.4 VDAC Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	VDAC_IPVERSION	R	IPVERSION
0x004	VDAC_EN	RW ENABLE	Module Enable
0x008	VDAC_SWRST	RW SWRST	Software Reset Register
0x00C	VDAC_CFG	RW CONFIG	Config Register
0x010	VDAC_STATUS	RH	Status Register
0x014	VDAC_CH0CFG	RW CONFIG	Channel 0 Config Register
0x018	VDAC_CH1CFG	RW CONFIG	Channel 1 Config Register
0x01C	VDAC_CMD	W SYNC	Command Register
0x020	VDAC_IF	RWH INTFLAG	Interrupt Flag Register
0x024	VDAC_IEN	RW	Interrupt Enable Register
0x028	VDAC_CH0F	W	Channel 0 Data Write Fifo
0x02C	VDAC_CH1F	W	Channel 1 Data Write Fifo
0x030	VDAC_OUTCTRL	RW SYNC	DAC Output Control
0x034	VDAC_OUTTIMERCFG	RW CONFIG	DAC Out Timer Config Register
0x1000	VDAC_IPVERSION_SET	R	IPVERSION
0x1004	VDAC_EN_SET	RW ENABLE	Module Enable
0x1008	VDAC_SWRST_SET	RW SWRST	Software Reset Register
0x100C	VDAC_CFG_SET	RW CONFIG	Config Register
0x1010	VDAC_STATUS_SET	RH	Status Register
0x1014	VDAC_CH0CFG_SET	RW CONFIG	Channel 0 Config Register
0x1018	VDAC_CH1CFG_SET	RW CONFIG	Channel 1 Config Register
0x101C	VDAC_CMD_SET	W SYNC	Command Register
0x1020	VDAC_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1024	VDAC_IEN_SET	RW	Interrupt Enable Register
0x1028	VDAC_CH0F_SET	W	Channel 0 Data Write Fifo
0x102C	VDAC_CH1F_SET	W	Channel 1 Data Write Fifo
0x1030	VDAC_OUTCTRL_SET	RW SYNC	DAC Output Control
0x1034	VDAC_OUTTIMERCFG_SET	RW CONFIG	DAC Out Timer Config Register
0x2000	VDAC_IPVERSION_CLR	R	IPVERSION
0x2004	VDAC_EN_CLR	RW ENABLE	Module Enable
0x2008	VDAC_SWRST_CLR	RW SWRST	Software Reset Register
0x200C	VDAC_CFG_CLR	RW CONFIG	Config Register
0x2010	VDAC_STATUS_CLR	RH	Status Register
0x2014	VDAC_CH0CFG_CLR	RW CONFIG	Channel 0 Config Register
0x2018	VDAC_CH1CFG_CLR	RW CONFIG	Channel 1 Config Register

Offset	Name	Type	Description
0x201C	VDAC_CMD_CLR	W SYNC	Command Register
0x2020	VDAC_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2024	VDAC_IEN_CLR	RW	Interrupt Enable Register
0x2028	VDAC_CH0F_CLR	W	Channel 0 Data Write Fifo
0x202C	VDAC_CH1F_CLR	W	Channel 1 Data Write Fifo
0x2030	VDAC_OUTCTRL_CLR	RW SYNC	DAC Output Control
0x2034	VDAC_OUTTIMERCFG_CLR	RW CONFIG	DAC Out Timer Config Register
0x3000	VDAC_IPVERSION_TGL	R	IPVERSION
0x3004	VDAC_EN_TGL	RW ENABLE	Module Enable
0x3008	VDAC_SWRST_TGL	RW SWRST	Software Reset Register
0x300C	VDAC_CFG_TGL	RW CONFIG	Config Register
0x3010	VDAC_STATUS_TGL	RH	Status Register
0x3014	VDAC_CH0CFG_TGL	RW CONFIG	Channel 0 Config Register
0x3018	VDAC_CH1CFG_TGL	RW CONFIG	Channel 1 Config Register
0x301C	VDAC_CMD_TGL	W SYNC	Command Register
0x3020	VDAC_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3024	VDAC_IEN_TGL	RW	Interrupt Enable Register
0x3028	VDAC_CH0F_TGL	W	Channel 0 Data Write Fifo
0x302C	VDAC_CH1F_TGL	W	Channel 1 Data Write Fifo
0x3030	VDAC_OUTCTRL_TGL	RW SYNC	DAC Output Control
0x3034	VDAC_OUTTIMERCFG_TGL	RW CONFIG	DAC Out Timer Config Register

29.5 VDAC Register Description

29.5.1 VDAC_IPVERSION - IPVERSION

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x3																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x3	R	**IPVERSION**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

29.5.2 VDAC_EN - Module Enable

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status**
	When EN is cleared, DISABLING is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs and FIFOs.			
0	EN	0x0	RW	**VDAC Module Enable**
	Enable the VDAC module. When EN is cleared(disablement), it halts module operation immediately, and initialize the core domain such that when the is re-enabled, it starts cleanly.			

Value	Mode	Description
0	DISABLE	Disable
1	ENABLE	Enable

29.5.3 VDAC_SWRST - Software Reset Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	W
Name																															RESETTING	SWRST

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESETTING	0x0	R	**Software reset busy status**
	When SWRST command is issued, resetting logic sets RESETTING status immediately, and later it is cleared when reset process finishes.			
0	SWRST	0x0	W	**Software reset command**
	A software reset command field resets the module back to the initial condition, similar to a power on reset condition			

29.5.4 VDAC_CFG - Config Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x2			0x0	0x0	0x0	0x0		0x0			0x0	0x0					0x0								0x0		0x0	0x0	0x0	0x0
Access	RW	RW			RW	RW	RW	RW		RW			RW	RW					RW								RW		RW	RW	RW	RW
Name	OUTENPRS	WARMUPTIME			DBGHALT	ONDEMANDCLK	DMAWU	BIASKEEPWARM		REFRESHPERIOD			SINEMODEPRS	TIMEROVRFLOWPERIOD					PRESC								REFRSEL		CH0PRESCRST	SINERESET	SINEMODE	DIFF

Bit	Name	Reset	Access	Description
31	OUTENPRS	0x0	RW	**PRS Controlled Output Enable**
	Enable PRS Control of DAC output enable			
	Value	Mode		Description
	0	DISOUTENPRS		DAC output enable always on if the ch0 is selected
	1	ENOUTENPRS		DAC output enable controlled by PRS signal selected for CH1
30:28	WARMUPTIME	0x2	RW	**DAC Warmup Time**
	Number of prescaled CLK_DAC +1 cycles for the VDAC to Warmup. Default is (2+1) prescaled CLK_DAC cycles.			
27	DBGHALT	0x0	RW	**Debug Halt**
	VDAC behavior when halted by debugger			
	Value	Mode		Description
	0	NORMAL		Continue operation as normal during debug mode
	1	HALT		Complete the current conversion and then halt during debug mode
26	ONDEMANDCLK	0x0	RW	**Always allow clk_dac**
	Setting this bit to 1 always allows CLK_DAC from CMU			
25	DMAWU	0x0	RW	**VDAC DMA Wakeup**
	Set to enable wakeup from EM2 to EM1 for DMA to fill CHx FIFO Data			
24	BIASKEEPWARM	0x0	RW	**Bias Keepwarm Mode Enable**
	Set this bit to keep the Bias on for Analog portion of DAC in between conversions. Primary purpose, is to reduce kick-back to the reference shared with IADC. Relevant only for Sample-off Mode			
23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:20	REFRESHPERIOD	0x0	RW	**Refresh Timer Overflow Period**
	Select refresh counter period. A channel x will be refreshed with the period set in REFRESHPERIOD if the channel in VDACn_CHxCFG has its REFRESHSOURCE set to REFRESHTIMER			

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	CYCLES2		All channels with enabled refresh are refreshed every 2 CLK_REFRESH cycles
	1	CYCLES4		All channels with enabled refresh are refreshed every 4 CLK_REFRESH cycles
	2	CYCLES8		All channels with enabled refresh are refreshed every 8 CLK_REFRESH cycles
	3	CYCLES16		All channels with enabled refresh are refreshed every 16 CLK_REFRESH cycles
	4	CYCLES32		All channels with enabled refresh are refreshed every 32 CLK_REFRESH cycles
	5	CYCLES64		All channels with enabled refresh are refreshed every 64 CLK_REFRESH cycles
	6	CYCLES128		All channels with enabled refresh are refreshed every 128 CLK_REFRESH cycles
	7	CYCLES256		All channels with enabled refresh are refreshed every 256 CLK_REFRESH cycles
19	SINEMODEPRS	0x0	RW	**Sinemode prs**
	Enable and disable sinemode prs			
	Value	Mode		Description
	0	DISSINEMODEPRS		Sine mode prs is disabled
	1	ENSINEMODEPRS		Sine mode prs is enabled
18:16	TIMEROVRFLOWPERIOD	0x0	RW	**Internal Timer Overflow Period**
	Select internal timer overflow period. A channel x will be provided with a conversion trigger after the period set in TIMEROVRFLOWPERIOD if the channel in VDACn_CHxCFG has its TRIGMODE set to INTERNALTIMER			
	Value	Mode		Description
	0	CYCLES2		The Timer overflows every 2 Prescaled CLK_DAC cycles
	1	CYCLES4		The Timer overflows every 4 Prescaled CLK_DAC cycles
	2	CYCLES8		The Timer overflows every 8 Prescaled CLK_DAC cycles
	3	CYCLES16		The Timer overflows every 16 Prescaled CLK_DAC cycles
	4	CYCLES32		The Timer overflows every 32 Prescaled CLK_DAC cycles
	5	CYCLES64		The Timer overflows every 64 Prescaled CLK_DAC cycles
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:7	PRESC	0x0	RW	**Prescaler Setting for DAC clock**
	Selected DAC clock source is prescaled by PRESC+1 to generate prescaled CLK_DAC with 50% duty cycle			
6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	REFRSEL	0x0	RW	**Reference Selection**

Bit	Name	Reset	Access	Description
	Select reference for analog portion of DAC			
	Value	Mode		Description
	0	V125		Internal 1.25 V bandgap reference
	1	V25		Internal 2.5 V bandgap reference
	2	VDD		AVDD reference
	3	EXT		External pin reference
3	CH0PRESCRST	0x0	RW	**Channel 0 Start Reset Prescaler**
	Select if prescaler (determining prescaled CLK_DAC rate) is reset on channel 0 start.			
	Value	Mode		Description
	0	NORESETPRESC		Prescaler not reset on channel 0 start
	1	RESETPRESC		Prescaler reset on channel 0 start
2	SINERESET	0x0	RW	**Sine Wave Reset When inactive**
	In case SINERESET is 0, SINEMODESTOP will stop progress of the sine wave at the sample currently being output in sinemode. When Set to 1,the sine output will reset to 0 degrees when SINEMODESTOP			
1	SINEMODE	0x0	RW	**Sine Mode**
	Enable/disable sine mode.			
	Value	Mode		Description
	0	DISSINEMODE		Sine mode disabled. Sine reset to 0 degrees
	1	ENSINEMODE		Sine mode enabled
0	DIFF	0x0	RW	**Differential Mode**
	Select single ended or differential mode.			
	Value	Mode		Description
	0	SINGLEENDED		Single ended output
	1	DIFFERENTIAL		Differential output

29.5.5 VDAC_STATUS - Status Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0	0x0	0x0	0x0			0x0	0x0		0x0	0x0		0x0			0x0					0x0	0x0			0x0	0x0			0x0	0x0
Access	R	R	R	R	R	R			R	R		R	R		R			R					R	R			R	R			R	R
Name	SYNCBUSY	ABUSALLOCERR	SINEACTIVE	ABUSINPUTCONFLICT	CH1FIFOFLBUSY	CH0FIFOFLBUSY			CH1FIFOEMPTY	CH0FIFOEMPTY		CH1CURRENTSTATE	CH0CURRENTSTATE		CH1FIFOCNT			CH0FIFOCNT					CH1FIFOFULL	CH0FIFOFULL			CH1WARM	CH0WARM			CH1ENS	CH0ENS

Bit	Name	Reset	Access	Description
31	SYNCBUSY	0x0	R	**Sync Busy Combined**
	Indicates synchronization ongoing			
30	ABUSALLOCERR	0x0	R	**ABUS Allocation Error Status**
	1 if ABUS requested is not allocated			
29	SINEACTIVE	0x0	R	**Sine Wave Output Status on Channel**
	1 indicates that sine wave output is generated on CH0 and/or CH1(in diff mode)			
28	ABUSINPUTCONFLICT	0x0	R	**ABUS Input Conflict Status**
	1 if both CH0 and CH1 request the same ABUS			
27	CH1FIFOFLBUSY	0x0	R	**CH1 FIFO Flush Sync Busy**
	Indicates CH1 FIFO Flush Sync Busy			
26	CH0FIFOFLBUSY	0x0	R	**CH0 FIFO Flush Sync Busy**
	Indicates CH0 FIFO Flush Sync Busy			
25:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23	CH1FIFOEMPTY	0x0	R	**Channel 1 FIFO Empty Status**
	1 if FIFO for Channel 1 is empty			
22	CH0FIFOEMPTY	0x0	R	**Channel 0 FIFO Empty Status**
	1 if FIFO for Channel 0 is empty			
21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20	CH1CURRENTSTATE	0x0	R	**Channel 1 Current Status**
	0 when CH1 is in IDLE State. Set when CH1 is converting data			
19	CH0CURRENTSTATE	0x0	R	**Channel 0 Current Status**
	0 when CH0 is in IDLE State. Set when CH0 is converting data			
18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17:15	CH1FIFOCNT	0x0	R	**Channel 1 FIFO Valid Count**

Bit	Name	Reset	Access	Description
	Number of Valid Entries in Channel 1. This FIFO entries Count is generated in Read Domain hence expect Synchronization Delay. Need to be used only when VDAC_CFG.ONDEMANDCLK is set to 1.			
14:12	CH0FIFOCNT	0x0	R	**Channel 0 FIFO Valid Count**
	Number of Valid Entries in Channel 0. This FIFO entries Count is generated in Read Domain hence expect Synchronization Delay. Need to be used only when VDAC_CFG.ONDEMANDCLK is set to 1.			
11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	CH1FIFOFULL	0x0	R	**Channel 1 FIFO Full Status**
	1 if FIFO for Channel 1 is full			
8	CH0FIFOFULL	0x0	R	**Channel 0 FIFO Full Status**
	1 if FIFO for Channel 0 is full			
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	CH1WARM	0x0	R	**Channel 1 Warmed Status**
	This bit is set when channel 1 has warmed up			
4	CH0WARM	0x0	R	**Channel 0 Warmed Status**
	This bit is set when channel 0 has warmed up			
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	CH1ENS	0x0	R	**Channel 1 Enabled Status**
	This bit is set when channel 1 is enabled.			
0	CH0ENS	0x0	R	**Channel 0 Enabled Status**
	This bit is set when channel 0 is enabled.			

29.5.6 VDAC_CH0CFG - Channel 0 Config Register

Offset	Bit Position																															
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x0		0x0		0x0			0x0			0x0			0x0		0x0
Access																RW		RW		RW			RW			RW			RW		RW
Name																KEEPWARM		HIGHCAPLOADEN		FIFODVL			REFRESHSOURCE			TRIGMODE			POWERMODE		CONVMODE

Bit	Name	Reset	Access	Description
31:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	KEEPWARM	0x0	RW	**Channel 0 Keepwarm Mode Enable**
	Set this bit to keep the Channel 0 on in between conversion in sample-off mode. Primary purpose of this is to reduce kickback between Channel 0 and Channel 1 and to reduce startup time.			
15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14	HIGHCAPLOADEN	0x0	RW	**Channel 0 High Cap Load Mode Enable**
	Enables High Capacitance Load Mode for Channel 0. Should be enabled with VDAC_CH0CFG.POWERMODE=HIGHPOWER			
13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12:11	FIFODVL	0x0	RW	**Channel 0 FIFO Low Watermark**
	Set Channel 0 FIFO Low Threshold Data Valid Level (DVL)			
10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	REFRESHSOURCE	0x0	RW	**Channel 0 Refresh Source**
	Select Channel 0 Refresh Trigger			

Value	Mode	Description
0	NONE	No Refresh Source Selected for Channel 0.
1	REFRESHTIMER	Channel 0 Refresh triggered by Refresh Timer Overflow
2	SYNCPRS	Channel 0 Refresh triggered by Sync PRS. PRS Trigger should have the same clock group as VDAC.
3	ASYNCPRS	Channel 0 Refresh triggered by Async PRS

7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	TRIGMODE	0x0	RW	**Channel 0 Trigger Mode**
	Select Channel 0 Conversion Trigger			

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	NONE		No Conversion Trigger Source Selected for Channel 0
	1	SW		Channel 0 is triggered by Channel 0 FIFO (CH0F) write
	2	SYNCPRS		Channel 0 is triggered by Sync PRS input. PRS Trigger should have the same clock group as VDAC.
	3	LESENSE		Channel 0 is triggered by LESENSE
	4	INTERNALTIMER		Channel 0 is triggered by Internal Timer Overflow
	5	ASYNCPRS		Channel 0 is triggered by Async PRS input
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	POWERMODE	0x0	RW	**Channel 0 Power Mode**
	Enable Power Mode for Channel 0			
	Value	Mode		Description
	0	HIGHPOWER		Default is High Power Mode
	1	LOWPOWER		Set this bit for Low Power Mode
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	CONVMODE	0x0	RW	**Channel 0 Conversion Mode**
	Configure conversion mode			
	Value	Mode		Description
	0	CONTINUOUS		DAC channel 0 is set in continuous mode
	1	SAMPLEOFF		DAC channel 0 is set in sample/shut off mode

29.5.7 VDAC_CH1CFG - Channel 1 Config Register

Offset	Bit Position																															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x0		0x0		0x0			0x0			0x0			0x0		0x0
Access																RW		RW		RW			RW			RW			RW		RW
Name																KEEPWARM		HIGHCAPLOADEN		FIFODVL			REFRESHSOURCE			TRIGMODE			POWERMODE		CONVMODE

Bit	Name	Reset	Access	Description
31:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	KEEPWARM	0x0	RW	**Channel 1 Keepwarm Mode Enable**
	Set this bit to keep the Channel 1 on in between conversion in sample-off mode. Primary purpose of this is to reduce kickback between Channel 0 and Channel 1 and to reduce startup time.			
15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14	HIGHCAPLOADEN	0x0	RW	**Channel 1 High Cap Load Mode Enable**
	Enables High Capacitance Load Mode for Channel 1. Should be enabled with VDAC_CH1CFG.POWERMODE=HIGHPOWER.			
13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12:11	FIFODVL	0x0	RW	**Channel 1 FIFO Low Watermark**
	Set Channel 1 Low threshold Data Valid Level (DVL)			
10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	REFRESHSOURCE	0x0	RW	**Channel 1 Refresh Source**
	Select Channel 1 Refresh Trigger			

Value	Mode	Description
0	NONE	No Refresh Source Selected
1	REFRESHTIMER	CH1 Refresh Triggered by Refresh Timer Overflow
2	SYNCPRS	CH1 Refresh Triggered by Sync PRS. PRS Trigger should have the same clock group as VDAC.
3	ASYNCPRS	CH1 Refresh Triggered by Async PRS

7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	TRIGMODE	0x0	RW	**Channel 1 Trigger Mode**
	Select Channel 1 Conversion Trigger			

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	NONE		No Conversion Trigger Source Selected for Channel 1
	1	SW		Channel 1 is triggered by Channel 1 FIFO (CH1F) write
	2	SYNCPRS		Channel 1 is triggered by Sync PRS input.PRS Trigger should have the same clock group as VDAC.
	4	INTERNALTIMER		Channel 1 is triggered by Internal Timer Overflow
	5	ASYNCPRS		Channel 1 is triggered by Async PRS input
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	POWERMODE	0x0	RW	**Channel 1 Power Mode**
	Enable Low Power Mode for Channel 1			
	Value	Mode		Description
	0	HIGHPOWER		Default is High Power Mode
	1	LOWPOWER		Set this bit for Low Power Mode
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	CONVMODE	0x0	RW	**Channel 1 Conversion Mode**
	Configure conversion mode for Channel 1			
	Value	Mode		Description
	0	CONTINUOUS		DAC channel 1 is set in continuous mode
	1	SAMPLEOFF		DAC channel 1 is set in sample/shut off mode

29.5.8 VDAC_CMD - Command Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0	0x0	0x0	0x0			0x0	0x0			0x0	0x0
Access																					W(nB)	W(nB)	W(nB)	W(nB)			W(nB)	W(nB)			W(nB)	W(nB)
Name																					SINEMODESTOP	SINEMODESTART	CH1FIFOFLUSH	CH0FIFOFLUSH			CH1DIS	CH1EN			CH0DIS	CH0EN

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	SINEMODESTOP	0x0	W(nB)	**Stop Sine Wave Generation**
	Stop Sine Wave Generation			
10	SINEMODESTART	0x0	W(nB)	**Start Sine Wave Generation**
	Start Sine Wave Generation			
9	CH1FIFOFLUSH	0x0	W(nB)	**CH1 WFIFO Flush**
	Flush Channel 1 WFIFO			
8	CH0FIFOFLUSH	0x0	W(nB)	**CH0 WFIFO Flush**
	Flush Channel 0 WFIFO			
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	CH1DIS	0x0	W(nB)	**DAC Channel 1 Disable**
	Disables DAC Channel 1			
4	CH1EN	0x0	W(nB)	**DAC Channel 1 Enable**
	Enables DAC Channel 1			
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	CH0DIS	0x0	W(nB)	**DAC Channel 0 Disable**
	Disables DAC Channel 0			
0	CH0EN	0x0	W(nB)	**DAC Channel 0 Enable**
	Enables DAC Channel 0			

29.5.9 VDAC_IF - Interrupt Flag Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset						0x0					0x0	0x0		0x0									0x0	0x0			0x0	0x0			0x0	0x0
Access						RW					RW	RW		RW									RW	RW			RW	RW			RW	RW
Name						ABUSINPUTCONFLICT					CH1DVL	CH0DVL		ABUSALLOCERR									CH1UF	CH0UF			CH1OF	CH0OF			CH1CD	CH0CD

Bit	Name	Reset	Access	Description
31:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26	ABUSINPUTCONFLICT	0x0	RW	**ABUS Input Conflict Error Flag**
	Set if both CH0 and CH1 request same ABUS. Should only be enabled when using ABUS for VDAC Output.			
25:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	CH1DVL	0x0	RW	**CH1 Data Valid Level Interrupt Flag**
	Set when Channel 1 FIFO Count reaches Data Valid Level. Also used as Wakeup IRQ			
20	CH0DVL	0x0	RW	**CH0 Data Valid Level Interrupt Flag**
	Set when Channel 0 FIFO Count reaches Data Valid Level. Also used as Wakeup IRQ			
19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18	ABUSALLOCERR	0x0	RW	**ABUS Port Allocation Error Flag**
	Set if APORT requested is not allocated. Should only be enabled when using ABUS for VDAC Output.			
17:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	CH1UF	0x0	RW	**CH1 Data Underflow Interrupt Flag**
	Indicates channel 1 data underflow.			
8	CH0UF	0x0	RW	**CH0 Data Underflow Interrupt Flag**
	Indicates channel 0 data underflow.			
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	CH1OF	0x0	RW	**CH1 Data Overflow Interrupt Flag**
	Indicates channel 1 data overflow.			
4	CH0OF	0x0	RW	**CH0 Data Overflow Interrupt Flag**
	Indicates channel 0 data overflow.			
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

Bit	Name	Reset	Access	Description
1	CH1CD	0x0	RW	**CH1 Conversion Done Interrupt Flag**
	Indicates channel 1 conversion complete.			
0	CH0CD	0x0	RW	**CH0 Conversion Done Interrupt Flag**
	Indicates channel 0 conversion complete.			

29.5.10 VDAC_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset						0x0					0x0	0x0		0x0									0x0	0x0			0x0	0x0			0x0	0x0
Access						RW					RW	RW		RW									RW	RW			RW	RW			RW	RW
Name						ABUSINPUTCONFLICT					CH1DVL	CH0DVL		ABUSALLOCERR									CH1UF	CH0UF			CH1OF	CH0OF			CH1CD	CH0CD

Bit	Name	Reset	Access	Description
31:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26	ABUSINPUTCONFLICT	0x0	RW	**ABUS Input Conflict Interrupt Flag**
	Set to enable the ABUSINPUTCONFLICTIF Interrupt			
25:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	CH1DVL	0x0	RW	**CH1 Data Valid Level Interrupt Flag**
	Set to enable the CH1DVLIF Interrupt			
20	CH0DVL	0x0	RW	**CH0 Data Valid Level Interrupt Flag**
	Set to enable the CH0DVLIF Interrupt			
19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18	ABUSALLOCERR	0x0	RW	**ABUS Allocation Error Interrupt Flag**
	Set to enable the ABUSALLOCERRIF Interrupt			
17:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	CH1UF	0x0	RW	**CH1 Data Underflow Interrupt Flag**
	Set to enable the CH1UFIF Interrupt			
8	CH0UF	0x0	RW	**CH0 Data Underflow Interrupt Flag**
	Set to enable the CH0UFIF Interrupt			
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	CH1OF	0x0	RW	**CH1 Data Overflow Interrupt Flag**
	Set to enable the CH1OFIF Interrupt			
4	CH0OF	0x0	RW	**CH0 Data Overflow Interrupt Flag**
	Set to enable the CH0OFIF Interrupt			
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

Bit	Name	Reset	Access	Description
1	CH1CD	0x0	RW	**CH1 Conversion Done Interrupt Flag**
	Set to enable the CH1CDIF Interrupt			
0	CH0CD	0x0	RW	**CH0 Conversion Done Interrupt Flag**
	Set to enable the CH0CDIF Interrupt			

29.5.11 VDAC_CH0F - Channel 0 Data Write Fifo

Offset	Bit Position																															
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0					
Access																											W					
Name																											DATA					

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:0	DATA	0x0	W	**Channel 0 Data**
	This register writes the value which will be converted by DAC channel 0 in a FIFO.			

29.5.12 VDAC_CH1F - Channel 1 Data Write Fifo

Offset	Bit Position																															
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0					
Access																											W					
Name																											DATA					

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:0	DATA	0x0	W	**Channel 1 Data**
	This register writes the value which will be converted by DAC channel 1 into a FIFO			

29.5.13 VDAC_OUTCTRL - DAC Output Control

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0x0						0x0				0x0						0x0					0x0	0x0			0x0	0x0			0x0	0x0
Access		RW						RW				RW						RW					RW	RW			RW	RW			RW	RW
Name		ABUSPINSELCH1						ABUSPORTSELCH1				ABUSPINSELCH0						ABUSPORTSELCH0					SHORTCH1	SHORTCH0			AUXOUTENCH1	AUXOUTENCH0			MAINOUTENCH1	MAINOUTENCH0

Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
30:25	ABUSPINSELCH1	0x0	RW	**CH1 ABUS Pin Select**
	Set this to Select a particular ABUS Pin for Channel 1			
24:22	ABUSPORTSELCH1	0x0	RW	**CH1 ABUS Port Select**
	Set this to Select a particular ABUS Port for Channel 1			

Value	Mode	Description
0	NONE	No GPIO Selected for CH1 ABUS Output
1	PORTA	Port A Selected
2	PORTB	Port B Selected
3	PORTC	Port C Selected
4	PORTD	Port D Selected

21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20:15	ABUSPINSELCH0	0x0	RW	**CH0 ABUS Pin Select**
	Set this to Select a particular ABUS Pin for Channel 0			
14:12	ABUSPORTSELCH0	0x0	RW	**CH0 ABUS Port Select**
	Set this to Select a particular ABUS Port for Channel 0			

Value	Mode	Description
0	NONE	No GPIO Selected for CH0 ABUS Output
1	PORTA	Port A Selected
2	PORTB	Port B Selected
3	PORTC	Port C Selected
4	PORTD	Port D Selected

11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

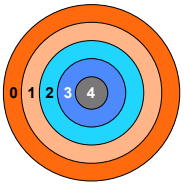
Bit	Name	Reset	Access	Description
9	SHORTCH1	0x0	RW	**CH0 Main and Alternative Output Short**
	Set this to short circuit Main and alternative Output of Channel 0. Usefull to connect Main and Alternative outputs together when DAC is enabled but not warmed up yet.			
8	SHORTCH0	0x0	RW	**CH1 Main and Alternative Output Short**
	Set this to short circuit Main and alternative Output of Channel 1. Usefull to connect Main and Alternative outputs together when DAC is enabled but not warmed up yet.			
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	AUXOUTENCH1	0x0	RW	**CH1 Alternative Output Enable**
	Set this to enable alternative output of Channel 1			
4	AUXOUTENCH0	0x0	RW	**CH0 Alternative Output Enable**
	Set this to enable alternative output of Channel 0			
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	MAINOUTENCH1	0x0	RW	**CH1 Main Output Enable**
	Set this to enable main output of Channel 1			
0	MAINOUTENCH0	0x0	RW	**CH0 Main Output Enable**
	Set this to enable main output of Channel 0			

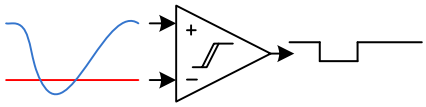
29.5.14 VDAC_OUTTIMERCFG - DAC Out Timer Config Register

Offset	Bit Position																													
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset								0x0															0x0								
Access								RW															RW								
Name								CH1OUTHOLDTIME															CH0OUTHOLDTIME								

Bit	Name	Reset	Access	Description
31:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24:15	CH1OUTHOLDTIME	0x0	RW	**CH1 Output Hold Time**
	Number of prescaled CLK_DAC cycles to drive the output of VDAC for Channel 1			
14:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:0	CH0OUTHOLDTIME	0x0	RW	**CH0 Output Hold Time**
	Number of prescaled CLK_DAC cycles to drive the output of VDAC for Channel 0			

30. ACMP - Analog Comparator





Quick Facts

What?

The ACMP (Analog Comparator) compares two analog signals and returns a digital value telling which is greater.

Why?

Applications often do not need to know the exact value of an analog signal, only if it has passed a certain threshold. Often the voltage must be monitored continuously, which requires extremely low power consumption.

How?

Available down to Energy Mode 3, the ACMP can wake up the system when input signals pass the threshold. The analog comparator can compare two analog signals or one analog signal and a highly configurable internal reference.

30.1 Introduction

The Analog Comparator compares the voltage of two analog inputs and outputs a digital signal indicating which input voltage is higher. Inputs can either be from internal references or from external pins. Response time, and thereby the current consumption, can be configured by altering the current supply to the comparator.

30.2 Features

- Internal and external input selections:
  - External port I/O routed via ABUS
  - Internal 1.25 V bandgap reference voltage with programmable divider
  - Internal 2.5 V bandgap reference voltage with programmable divider
  - AVDD supply voltage with programmable divider
  - VDAC auxiliary outputs
- Voltage supply monitoring
- Configurable hysteresis
- Selectable response time
- Operational in EM0 to EM3
- Asynchronous interrupt generation on selectable edges
- Configurable output state when inactive
- Output routing options:
  - Route to GPIO via DBUS
  - Route to most peripherals via PRS
  - Available to LESENSE logic

30.3 Functional Description

An overview of the ACMP is shown in Figure 30.1 ACMP Overview on page 1213.

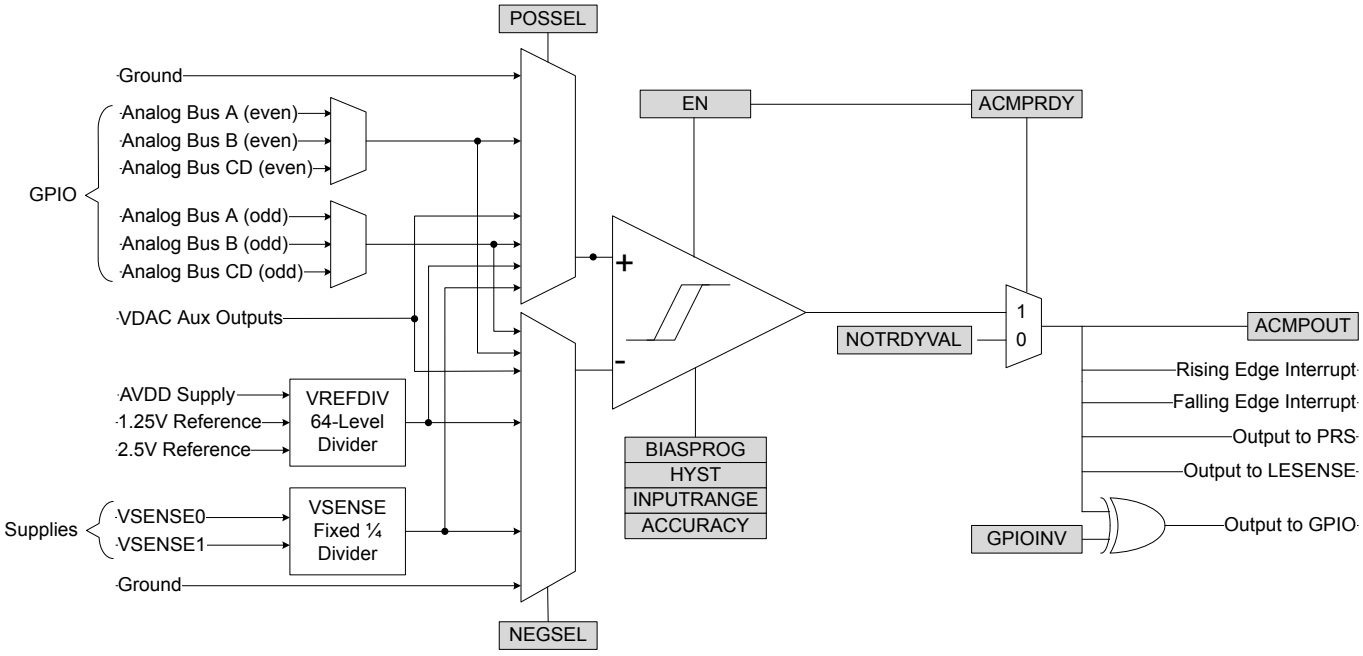


Figure 30.1. ACMP Overview

The comparator has two analog inputs: one positive and one negative. When the comparator is active, the output indicates which of the two input voltages is higher. When the voltage on the positive input is higher than the voltage on the negative input, the digital output is high and vice versa.

In addition to the comparator core, the ACMP front-end includes reference sources, voltage divider circuits, and input muxes to route signals to the positive and negative inputs. The output from the ACMP is available on both PRS and GPIO, in addition to being observable in the ACMP_STATUS register.

30.3.1 Configuration and Control

The ACMP is configured and controlled through three registers: ACMP_CFG, ACMP_CTRL, and ACMP_INPUTCTRL. Configuration through ACMP_CFG needs to happen before the ACMP is enabled.The control registers ACMP_CTRL and ACMP_INPUTCTRL can only be updated after the ACMP is enabled. The ACMP is enabled by setting the EN bit in ACMP_EN. If ACMP_CFG is updated when EN = 1, or ACMP_CTRL / ACMP_INPUTCTRL is updated while EN = 0, a bus fault is issued.

The input muxes are configured in the POSSEL / NEGSEL bitfields in ACMP_INPUTCTRL. All references and inputs are available in the modes defined for these two registers. The INPUTCTRL bit in ACMP_SYNCBUSY should be checked before writing to ACMP_INPUTCTRL. If the ACMP_SYNCBUSY_INPUTCTRL bit is 1, it means a previous write to the ACMP_INPUTCTRL register is pending, and software should wait until ACMP_SYNCBUSY_INPUTCTRL bit reads 0.

The POSSEL and NEGSEL muxes share several resources on the device, such as the VREFDIV and VSENSE divider circuits. Thus, there are some constraints on the POSSEL / NEGSEL configurations:

- POSSEL and NEGSEL cannot select an even numbered GPIO pin at the same time.
- POSSEL and NEGSEL cannot select an odd numbered GPIO pin at the same time.
- POSSEL and NEGSEL cannot both select a supply voltage via one of the VSENSE inputs.
- POSSEL and NEGSEL cannot both select an input using VREFDIV.
- If POSSEL = EXTPx, a low power reference (postfixed with LP) cannot be selected for NEGSEL.

If one of these constraints are violated, the INPUTCONFLICT status flag and INPUTCONFLICTIF interrupt flag will be set.

The ACMP also uses shared chip-level analog bus resources to connect to external GPIO pins. Which bus the ACMP is using depends on the configuration of POSSEL and NEGSEL. To allow the ACMP to control an analog bus, the bus must be allocated to ACMP in the GPIO module, using the GPIO_xBUSALLOC registers. For example, pin PB5 is an odd-numbered pin on port PB, and could connect via either analog bus BODD0 or BODD1. This is configured using the BODD0 or BODD1 field in GPIO_BBUSALLOC.

If the ACMP peripheral is trying to access a bus that has not been allocated to that instance of ACMP, the PORTALLOCERR status flag and PORTALLOCERRIF interrupt flag will be set.

30.3.2 Warmup Time

When the comparator is enabled or the input muxes are reconfigured, it requires some time to stabilize. On first enable (ACMP_EN_EN = 1), the comparator core requires 2.5 us to stabilize. In addition to this, any references selected may require some time to warm up. See table Table 30.1 Warmup Time on page 1214 for warmup times for the different references. When reconfiguring the ACMP (without disabling it), only the warmup times given in the table will be observed. When the comparator is ready for use, the ACMPRDY status flag and the ACMPRDY interrupt flag will be set.

Note: The hardware timeout is not sufficient to ensure glitch-free operation when using BIAS<=3. To avoid startup glitches, software should wait for 60 us after enabling the comparator to use the output.

During the warmup time and when the comparator is inactive, the comparator output will be set to the state defined by the NOTRDYVAL bit in ACMP_CTRL.

Table 30.1. Warmup Time

Reference	Warmup time
Low power reference: POSSEL / NEGSEL = *LP	10 us
VSENSE: POSSEL / NEGSEL = VSENSE*	5 us
VREF: POSSEL / NEGSEL = VREF*	2 us
None of the above	0.5 us

30.3.3 Response Time

There is a delay from when the input voltage changes polarity to when the output toggles. This delay is called the response time and can be altered by increasing or decreasing the bias current to the comparator through the BIASPROG bitfield in the ACMP_CFG register. The current and speed of the circuit increase as the value of BIASPROG is increased. See the part datasheet for specific current and response times releated to setting of BIASPROG.

30.3.4 Hysteresis

When the hysteresis level is set to a non-zero value, the digital output will not toggle until the positive input voltage is at a voltage equal to the hysteresis level above or below the negative input voltage (see Figure 30.2 Hysteresis on page 1215). This feature can be used avoid continual comparator output changes due to input noise when the positive and negative inputs are similar. Hysteresis requires the input difference to exceed the hysteresis threshold before the output can change and can reject limited amounts of noise. The hystersis in ACMP can be configured to three different levels (10 mV, 20 mV, 30 mV), and can be enabled on positive (rising), negative (falling), or both edges. Hysteresis is configured in the HYST bitfield in ACMP_CFG.

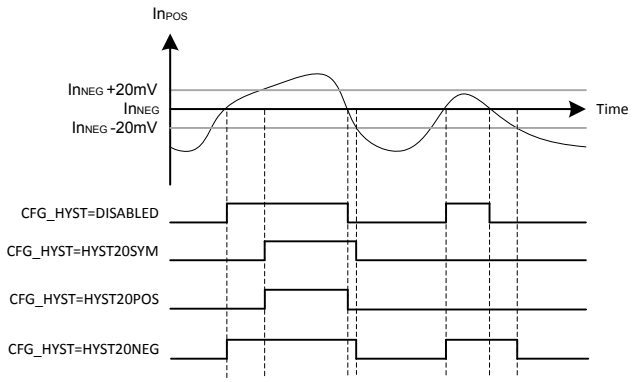


Figure 30.2. Hysteresis

30.3.5 LESENSE Interface

The ACMP can be controlled by LESENSE using its external override interface. In this mode, the LESENSE module will take control of the positive input mux control signal, and possibly the voltage divider, VREFDIV.

To enable this mode, INPUTCTRL_POSSEL needs to be set to EXTPA, EXTPB, EXTPC or EXTPD, according to which GPIO port will be used. In the LESENSE channel configuration, the LESENSE_CHx_INTERACT.OFFSET field will determine which pin on the selected port will be connected for that channel. For example, to connect the positive comparator input to PB03, INPUTCTRL_POSSEL would be set to EXTPB, and LESENSE_CHx_INTERACT.OFFSET to 0x03.

The negative input of the comparator in this mode is still connected according to INPUTCTRL_NEGSEL. LESENSE will take control of the voltage divider VREFDIV setting, if LESENSE_PERCTRL_ACMPxMODE is set to MUXTHRES. For the selected channel, VREFDIV will come from the 6 LSBs of LESENSE_CHx_INTERACT_THRES.

30.3.6 Supply Voltage Monitoring (VSENSE)

The ACMP can be used to monitor supply voltages. This is done by selecting VSENSE01DIV4(LP) or VSENSE11DIV4(LP) for either POSSEL or NEGSEL. Note that the input to the comparator core will be divided by 4, as illustrated in Figure 30.1 ACMP Overview on page 1213. To reduce energy consumption, a sample/hold circuit can be used to periodically sample the power supplies. To enable this, select VSENSE01DIV4LP or VSENSE11DIV4LP in POSSEL / NEGSEL. Because the sample/hold feature uses the comparator in a non-continuous fashion, enabling this will increase response times and reduce the accuracy of the comparator. The connections between VSENSE0 and VSENSE1 to power supplies are summarized in the table below.

Table 30.2. VSENSE connections

ACMP instance	VSENSE0	VSENSE1
ACMP0	AVDD	VDDIO0
ACMP1	DVDD	Not connected

30.3.7 VREFDIV Sources

The ACMP has two internal bandgap references: 2.5 V and 1.25 V. In addition, AVDD can be used as a reference. To select one of these references, configure POSSEL / NEGSEL to VREFDIVAVDD, VREFDIV1V25, or VREFDIV2V5. The ACMP also includes sample/hold functionality to reduce energy consumption. To enable the sample/hold feature, select VREFDIVAVDDLP, VREFDIV1V25LP, or VREFDIV2V5LP. These references can be divided by configuring VREFDIV in ACMP_INPUTCTRL. This division factor will be VREFDIV / 63, such that VREFOUT = VREFIN * (VREFDIV / 63).

30.3.8 Input Range and Accuracy Settings

By default, the ACMP can accept external rail-to-rail inputs, from 0 to AVDD. If external voltages will never be higher than AVDD - 0.7 V, the INPUTRANGE bit in ACMP_CFG can be set to 1 to reduce the power consumption of the block.

The ACMP also has an adjustable accuracy setting (ACCURACY in ACMP_CFG). ACCURACY is set to LOW by default, which conserves power, but may have degraded performance for rapidly changing analog Port selections in either the ACMP or the GPIO. ACCURACY can be set to HIGH to insure ACMP accuracy (at the expense of extra power consumption), when configuration changes are expected at a high rate (more than once per ms, for example), such as when scanning through channels.

30.3.9 Interrupts and PRS Output

The analog comparator includes independent output flags for rising edge (RISEIF) and falling edge (FALLIF) events. These will be set when a rising or falling edge is detected, respectively.

Three other interrupt sources are also available. PORTALLOCERRIF and INPUTCONFLICTIF are input configuration error flags, detailed in 30.3.1 Configuration and Control. The ACMPRDYIF flag indicates comparator stability after the warmup period.

The comparator output is available as an asynchronous PRS producer, and can be routed to other peripherals in the system via PRS.

30.3.10 Output to GPIO

The output from the comparator is available as alternate functions to the GPIO pins. Each pin connection can be enabled/disabled separately using the GPIO module control registers. See the device data sheet for the available locations for each signal.

The GPIO pin must also be set as output. The output to the GPIO can be inverted by setting the GPIOINV bit in ACMP_CTRL.

30.4 ACMP Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	ACMP_IPVERSION	R	IP Version ID
0x004	ACMP_EN	RW ENABLE	ACMP Enable
0x008	ACMP_SWRST	RW SWRST	Software Reset
0x00C	ACMP_CFG	RW CONFIG	Configuration Register
0x010	ACMP_CTRL	RW	Control Register
0x014	ACMP_INPUTCTRL	RW SYNC	Input Control Register
0x018	ACMP_STATUS	RH	Status Register
0x01C	ACMP_IF	RWH INTFLAG	Interrupt Flag Register
0x020	ACMP_IEN	RW	Interrupt Enable Register
0x024	ACMP_SYNCBUSY	RH	Syncbusy
0x1000	ACMP_IPVERSION_SET	R	IP Version ID
0x1004	ACMP_EN_SET	RW ENABLE	ACMP Enable
0x1008	ACMP_SWRST_SET	RW SWRST	Software Reset
0x100C	ACMP_CFG_SET	RW CONFIG	Configuration Register
0x1010	ACMP_CTRL_SET	RW	Control Register
0x1014	ACMP_INPUTCTRL_SET	RW SYNC	Input Control Register
0x1018	ACMP_STATUS_SET	RH	Status Register
0x101C	ACMP_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1020	ACMP_IEN_SET	RW	Interrupt Enable Register
0x1024	ACMP_SYNCBUSY_SET	RH	Syncbusy
0x2000	ACMP_IPVERSION_CLR	R	IP Version ID
0x2004	ACMP_EN_CLR	RW ENABLE	ACMP Enable
0x2008	ACMP_SWRST_CLR	RW SWRST	Software Reset
0x200C	ACMP_CFG_CLR	RW CONFIG	Configuration Register
0x2010	ACMP_CTRL_CLR	RW	Control Register
0x2014	ACMP_INPUTCTRL_CLR	RW SYNC	Input Control Register
0x2018	ACMP_STATUS_CLR	RH	Status Register
0x201C	ACMP_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2020	ACMP_IEN_CLR	RW	Interrupt Enable Register
0x2024	ACMP_SYNCBUSY_CLR	RH	Syncbusy
0x3000	ACMP_IPVERSION_TGL	R	IP Version ID
0x3004	ACMP_EN_TGL	RW ENABLE	ACMP Enable
0x3008	ACMP_SWRST_TGL	RW SWRST	Software Reset
0x300C	ACMP_CFG_TGL	RW CONFIG	Configuration Register
0x3010	ACMP_CTRL_TGL	RW	Control Register

Offset	Name	Type	Description
0x3014	ACMP_INPUTCTRL_TGL	RW SYNC	Input Control Register
0x3018	ACMP_STATUS_TGL	RH	Status Register
0x301C	ACMP_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3020	ACMP_IEN_TGL	RW	Interrupt Enable Register
0x3024	ACMP_SYNCBUSY_TGL	RH	Syncbusy

30.5 ACMP Register Description

30.5.1 ACMP_IPVERSION - IP Version ID

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x4																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x4	R	**IP version ID**

30.5.2 ACMP_EN - ACMP Enable

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status** When En is cleared, DISABLING is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs and FIFOs.
0	EN	0x0	RW	**Module enable** The ENABLE bit enables the module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit.

30.5.3 ACMP_SWRST - Software Reset

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	W
Name																															RESETTING	SWRST

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESETTING	0x0	R	**Software reset busy status**
0	SWRST	0x0	W	**Software reset**

30.5.4 ACMP_CFG - Configuration Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0	0x0						0x0									0x4	
Access															RW	RW						RW									RW	
Name															ACCURACY	INPUTRANGE						HYST									BIAS	

Bit	Name	Reset	Access	Description
31:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	ACCURACY	0x0	RW	**ACMP accuracy mode**

Setting ACCURACY to HIGH reduces the noise in the signal input path of the ACMP. Note, high frequency changes can cause the ACMP performance to degrade. For such uses, such as quickly scanning through multiple channels, this should be set to HIGH.

Value	Mode	Description
0	LOW	ACMP operates in low-accuracy mode but consumes less current.
1	HIGH	ACMP operates in high-accuracy mode but consumes more current.

Bit	Name	Reset	Access	Description
16	INPUTRANGE	0x0	RW	**Input Range**

Adjust performance of the comparator for a given input voltage range.

Value	Mode	Description
0	FULL	Use this setting when the input to the comparator core can be from 0 to AVDD.
1	REDUCED	It is recommended to use this setting when the input to the comparator core will always be less than AVDD-0.7V.

Bit	Name	Reset	Access	Description
15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:8	HYST	0x0	RW	**Hysteresis mode**

Set hysteresis mode and level.

Value	Mode	Description
0	DISABLED	Hysteresis disabled
1	SYM10MV	10mV symmetrical hysteresis
2	SYM20MV	20mV symmetrical hysteresis
3	SYM30MV	30mV symmetrical hysteresis
4	POS10MV	10mV hysteresis on positive edge transitions
5	POS20MV	20mV hysteresis on positive edge transitions

Bit	Name	Reset	Access	Description
	6	POS30MV		30mV hysteresis on positive edge transitions
	8	NEG10MV		10mV hysteresis on negative edge transitions
	9	NEG20MV		20mV hysteresis on negative edge transitions
	10	NEG30MV		30mV hysteresis on negative edge transitions
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	BIAS	0x4	RW	**Bias Configuration**
	These bits control the bias current level. See the datasheet for details.			

30.5.5 ACMP_CTRL - Control Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															GPIOINV	NOTRDYVAL

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	GPIOINV	0x0	RW	**Comparator GPIO Output Invert**
	Set this bit to 1 to invert the comparator alternate function output to GPIO.			
	Value	Mode		Description
	0	NOTINV		The comparator output to GPIO is not inverted
	1	INV		The comparator output to GPIO is inverted
0	NOTRDYVAL	0x0	RW	**Not Ready Value**
	The value of this bit is used as the comparator output when the comparator is not ready.			
	Value	Mode		Description
	0	LOW		ACMP output is 0 when the ACMP is not ready.
	1	HIGH		ACMP output is 1 when the ACMP is not ready.

30.5.6 ACMP_INPUTCTRL - Input Control Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0x0									0x0						0x0								0x0							
Access		RW									RW						RW								RW							
Name		CSRESSEL									VREFDIV						NEGSEL								POSSEL							

Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
30:28	CSRESSEL	0x0	RW	**Capacitive Sense Mode Internal Resistor**
	Deprecated capacitive sensing feature, not recommended for new designs			

Value	Mode	Description
0	RES0	Internal capacitive sense resistor value 0
1	RES1	Internal capacitive sense resistor value 1
2	RES2	Internal capacitive sense resistor value 2
3	RES3	Internal capacitive sense resistor value 3
4	RES4	Internal capacitive sense resistor value 4
5	RES5	Internal capacitive sense resistor value 5
6	RES6	Internal capacitive sense resistor value 6

27:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:16	VREFDIV	0x0	RW	**VREF division**
	Set division factor for VREFDIV. VREFOUT = VREFIN * (VREFDIV / 63)			
15:8	NEGSEL	0x0	RW	**Negative Input Select**
	Select negative input.			

Value	Mode	Description
0	VSS	VSS
16	VREFDIVAVDD	Divided AVDD
17	VREFDIVAVDDLP	Low-Power Divided AVDD
18	VREFDIV1V25	Divided 1V25 reference
19	VREFDIV1V25LP	Low-power Divided 1V25 reference
20	VREFDIV2V5	Divided 2V5 reference
21	VREFDIV2V5LP	Low-power Divided 2V5 reference
32	VSENSE01DIV4	VSENSE0 divided by 4
33	VSENSE01DIV4LP	Low-power VSENSE0 divided by 4

Bit	Name	Reset	Access	Description
	34	VSENSE1DIV4		VSENSE1 divided by 4
	35	VSENSE1DIV4LP		Low-power VSENSE1 divided by 4
	48	CAPSENSE		Deprecated capacitive sensing feature, not recommended for new designs
	64	VDACOUT0		VDAC0 channel 0 output
	65	VDACOUT1		VDAC0 channel 1 output
	128	PA0		Port A, Pin0
	129	PA1		Port A, Pin1
	130	PA2		Port A, Pin2
	131	PA3		Port A, Pin3
	132	PA4		Port A, Pin4
	133	PA5		Port A, Pin5
	134	PA6		Port A, Pin6
	135	PA7		Port A, Pin7
	136	PA8		Port A, Pin8
	137	PA9		Port A, Pin9
	138	PA10		Port A, Pin10
	139	PA11		Port A, Pin11
	140	PA12		Port A, Pin12
	141	PA13		Port A, Pin13
	142	PA14		Port A, Pin14
	143	PA15		Port A, Pin15
	144	PB0		Port B, Pin0
	145	PB1		Port B, Pin1
	146	PB2		Port B, Pin2
	147	PB3		Port B, Pin3
	148	PB4		Port B, Pin4
	149	PB5		Port B, Pin5
	150	PB6		Port B, Pin6
	151	PB7		Port B, Pin7
	152	PB8		Port B, Pin8
	153	PB9		Port B, Pin9
	154	PB10		Port B, Pin10
	155	PB11		Port B, Pin11
	156	PB12		Port B, Pin12
	157	PB13		Port B, Pin13
	158	PB14		Port B, Pin14

Bit	Name	Reset	Access	Description
	159	PB15		Port B, Pin15
	160	PC0		Port C, Pin0
	161	PC1		Port C, Pin1
	162	PC2		Port C, Pin2
	163	PC3		Port C, Pin3
	164	PC4		Port C, Pin4
	165	PC5		Port C, Pin5
	166	PC6		Port C, Pin6
	167	PC7		Port C, Pin7
	168	PC8		Port C, Pin8
	169	PC9		Port C, Pin9
	170	PC10		Port C, Pin10
	171	PC11		Port C, Pin11
	172	PC12		Port C, Pin12
	173	PC13		Port C, Pin13
	174	PC14		Port C, Pin14
	175	PC15		Port C, Pin15
	176	PD0		Port D, Pin0
	177	PD1		Port D, Pin1
	178	PD2		Port D, Pin2
	179	PD3		Port D, Pin3
	180	PD4		Port D, Pin4
	181	PD5		Port D, Pin5
	182	PD6		Port D, Pin6
	183	PD7		Port D, Pin7
	184	PD8		Port D, Pin8
	185	PD9		Port D, Pin9
	186	PD10		Port D, Pin10
	187	PD11		Port D, Pin11
	188	PD12		Port D, Pin12
	189	PD13		Port D, Pin13
	190	PD14		Port D, Pin14
	191	PD15		Port D, Pin15
7:0	POSSEL	0x0	RW	**Positive Input Select**
	Select positive input.			
	Value	Mode		Description

Bit	Name	Reset	Access	Description
	0	VSS		VSS
	16	VREFDIVAVDD		Divided AVDD
	17	VREFDIVAVDDLP		Low-Power Divided AVDD
	18	VREFDIV1V25		Divided 1V25 reference
	19	VREFDIV1V25LP		Low-power Divided 1V25 reference
	20	VREFDIV2V5		Divided 2V5 reference
	21	VREFDIV2V5LP		Low-power Divided 2V5 reference
	32	VSENSE01DIV4		VSENSE0 divided by 4
	33	VSENSE01DIV4LP		Low-power VSENSE0 divided by 4
	34	VSENSE11DIV4		VSENSE1 divided by 4
	35	VSENSE11DIV4LP		Low-power VSENSE1 divided by 4
	64	VDACOUT0		VDAC0 channel 0 output
	65	VDACOUT1		VDAC0 channel 1 output
	80	EXTPA		External interface, base is PA0.
	81	EXTPB		External interface, base is PB0.
	82	EXTPC		External interface, base is PC0.
	83	EXTPD		External interface, base is PD0.
	128	PA0		Port A, Pin0
	129	PA1		Port A, Pin1
	130	PA2		Port A, Pin2
	131	PA3		Port A, Pin3
	132	PA4		Port A, Pin4
	133	PA5		Port A, Pin5
	134	PA6		Port A, Pin6
	135	PA7		Port A, Pin7
	136	PA8		Port A, Pin8
	137	PA9		Port A, Pin9
	138	PA10		Port A, Pin10
	139	PA11		Port A, Pin11
	140	PA12		Port A, Pin12
	141	PA13		Port A, Pin13
	142	PA14		Port A, Pin14
	143	PA15		Port A, Pin15
	144	PB0		Port B, Pin0
	145	PB1		Port B, Pin1
	146	PB2		Port B, Pin2
	147	PB3		Port B, Pin3

Bit	Name	Reset	Access	Description
	148	PB4		Port B, Pin4
	149	PB5		Port B, Pin5
	150	PB6		Port B, Pin6
	151	PB7		Port B, Pin7
	152	PB8		Port B, Pin8
	153	PB9		Port B, Pin9
	154	PB10		Port B, Pin10
	155	PB11		Port B, Pin11
	156	PB12		Port B, Pin12
	157	PB13		Port B, Pin13
	158	PB14		Port B, Pin14
	159	PB15		Port B, Pin15
	160	PC0		Port C, Pin0
	161	PC1		Port C, Pin1
	162	PC2		Port C, Pin2
	163	PC3		Port C, Pin3
	164	PC4		Port C, Pin4
	165	PC5		Port C, Pin5
	166	PC6		Port C, Pin6
	167	PC7		Port C, Pin7
	168	PC8		Port C, Pin8
	169	PC9		Port C, Pin9
	170	PC10		Port C, Pin10
	171	PC11		Port C, Pin11
	172	PC12		Port C, Pin12
	173	PC13		Port C, Pin13
	174	PC14		Port C, Pin14
	175	PC15		Port C, Pin15
	176	PD0		Port D, Pin0
	177	PD1		Port D, Pin1
	178	PD2		Port D, Pin2
	179	PD3		Port D, Pin3
	180	PD4		Port D, Pin4
	181	PD5		Port D, Pin5
	182	PD6		Port D, Pin6
	183	PD7		Port D, Pin7
	184	PD8		Port D, Pin8

Bit	Name	Reset	Access	Description
	185	PD9		Port D, Pin9
	186	PD10		Port D, Pin10
	187	PD11		Port D, Pin11
	188	PD12		Port D, Pin12
	189	PD13		Port D, Pin13
	190	PD14		Port D, Pin14
	191	PD15		Port D, Pin15

30.5.7 ACMP_STATUS - Status Register

Offset	Bit Position																															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0		0x0
Access																												R	R	R		R
Name																												PORTALLOCERR	INPUTCONFLICT	ACMPRDY		ACMPOUT

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	PORTALLOCERR	0x0	R	**Port allocation error**
	The port selected by INPUTCTRL_POSSEL or INPUTCTRL_NEGSEL is not allocated to this ACMP. Port allocation needs to be configured in the GPIO module.			
3	INPUTCONFLICT	0x0	R	**INPUT conflict**
	INPUTCTRL_POSSEL and INPUTCTRL_NEGSEL is configured illegally.			
2	ACMPRDY	0x0	R	**Analog Comparator Ready**
	Analog comparator ready status.			
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	ACMPOUT	0x0	R	**Analog Comparator Output**
	Analog comparator output value.			

30.5.8 ACMP_IF - Interrupt Flag Register

Offset	Bit Position																														
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												PORTALLOCERR	INPUTCONFLICT	ACMPRDY	FALL	RISE

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	PORTALLOCERR	0x0	RW	**Port allocation error**
	The port selected by INPUTCTRL_POSSEL or INPUTCTRL_NEGSEL is not allocated to this ACMP. Port allocation needs to be configured in the GPIO module.			
3	INPUTCONFLICT	0x0	RW	**Input conflict**
	INPUTCTRL_POSSEL and INPUTCTRL_NEGSEL is configured illegally.			
2	ACMPRDY	0x0	RW	**ACMP ready Interrupt flag**
	Indicates that the analog comparator is ready and references have settled. Note at lower bias settings (BIAS<=3), the ACMPRDY bit may not be reliable and additional software wait may be required. Refer to "Warmup Time" section for additional information.			
1	FALL	0x0	RW	**Falling Edge Triggered Interrupt Flag**
	Indicates that there has been a falling edge on the analog comparator output.			
0	RISE	0x0	RW	**Rising Edge Triggered Interrupt Flag**
	Indicates that there has been a rising edge on the analog comparator output.			

30.5.9 ACMP_IEN - Interrupt Enable Register

Offset	Bit Position																														
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												PORTALLOCERR	INPUTCONFLICT	ACMPRDY	FALL	RISE

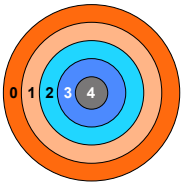
Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	PORTALLOCERR	0x0	RW	**Port allocation error interrupt enable**
3	INPUTCONFLICT	0x0	RW	**Input conflict interrupt enable**
2	ACMPRDY	0x0	RW	**ACMP ready interrupt enable**
1	FALL	0x0	RW	**Falling edge interrupt enable**
0	RISE	0x0	RW	**Rising edge interrupt enable**

30.5.10 ACMP_SYNCBUSY - Syncbusy

Offset	Bit Position																														
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																R
Name																																INPUTCTRL

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	INPUTCTRL	0x0	R	**Syncbusy for INPUTCTRL**
	Synchronization of INPUTCTRL ongoing			

31. KEYSCAN - Keyboard Scan



Quick Facts

What?

The keypad scanner autonomously scans through a matrix of keypad switches to detect key presses.

Why?

Operating independently from the main processor, the keypad scanner can save energy by interrupting the system only as needed. The KEYSCAN peripheral also saves energy by operating in a wake-on-press mode in EM2 and EM3.

How?

A traditional row/column matrix of switches is scanned using a finite state machine capable of debouncing and settling any key press prior to processor intervention.

31.1 Introduction

The KEYSCAN module connects through row and column GPIOs to an external mechanical keypad. The KEYSCAN supports up to 6 rows and up to 8 columns.

31.2 Features

- Performs scans in EM0 / EM1 to detect the row and column address of any key
- Wake up from sleep on key press in EM2 / EM3
- Supports up to a 6 x 8 matrix of keys in one or more keypads
- Interrupt sources for detected key press and non-press
- Multi-touch mode available for detecting mutiple simultaneous keys

31.3 Functional Description

An overview of the KEYSCAN module is shown in Figure 31.1 KEYSCAN Overview on page 1231. The KEYSCAN consists of rows and columns GPIOs connected to the KEYSCAN hardware. In EM0 and EM1, KEYSCAN can actively scan all switches in the matrix for active key presses. Any closed switches between rows and columns are detected, debounced, settled, and reported to the processor via the register interface. In EM2 and EM3, KEYSCAN simultaneously checks for any key press across the entire switch matrix. Any press will wake to an active state, allowing KEYSCAN to perform a scan and determine specific key location(s).

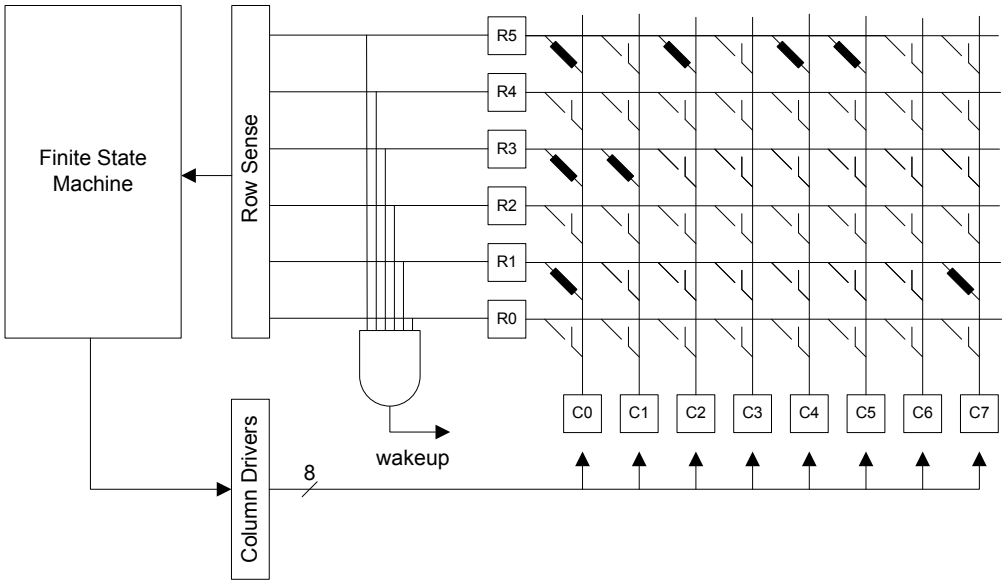


Figure 31.1. KEYSCAN Overview

31.3.1 Row and Column Configuration

Row and column signals are routed to physical pins using DBUS and set up in the GPIO configuration registers.

Columns are the output drivers and can be routed to any available port pin using the GPIO_KEYSCAN_COLOUTxROUTE register, where 'x' is the column number. Column outputs must also be individually enabled using the bits in GPIO_KEYSCAN_ROUTEN.

Rows are the input sense circuitry. They are routed via the GPIO_KEYSCAN_ROWSENSExROUTE registers, where 'x' is the row number. Row signals are only available on ports A and B.

Rows and columns should be routed starting at the lowest number. For example, a keypad requiring 3 rows and 2 columns should route ROW0, ROW1, ROW2, COL0, and COL1.

When the GPIO registers have been configured, the KEYSCAN_CFG.NUMROWS and KEYSCAN_CFG.NUMCOLS can then be configured to indicate the number of rows and columns to include in each scan.

31.3.2 Clocking and Timing

The timebase for the KEYSCAN peripheral is derived from EM01GRPACLK. The CFG.CLKDIV field should be used to divide EM01GRPACLK to produce a 500 Hz clock (2 ms ticks), which drives the timing of delays in the KEYSCAN state machine.

There are three timed delays used in the state machine, which are described in more detail in the following sections. Timing for these delays is configured by the SCANDLY, DEBDLY, and STABDLY fields in the DELAY register. Each of these fields specifies the number of 2 ms clock ticks for a different delay, with a range from 2 ms to 32 ms.

31.3.3 Scanning

Figure 31.2 KEYSCAN State Machine on page 1232 shows the state machine sequence for key press scanning. When a scanning sequence begins, the scan logic grounds one column at a time, starting with column 0. The column pin is grounded for up to DELAY.SCANDLY clocks. If no rows are detected as low (indicating a key press) the scan logic will proceed to ground the next column in sequence.

If a row is read as logic low while a column is grounded, the scan logic will implement a debounce delay of DELAY.DEBDLY clocks. If any rows are still grounded at the end of the debounce delay, a stabilization delay of DELAY.STABDLY clocks will then be implemented.

If the row readings between the start and end of the stabilizaton delay match, the scan logic determines that a key has been pressed, the scanner halts, and the IF.KEY interrupt flag is set. The details of which column is tested are reported in STATUS.COL, and the reading from all rows is stored in STATUS.ROW. The state machine will continue once IF.KEY is cleared.

For single-press scanning (CFG.SINGLEPRESS set to SINGLEPRESS), the next step in the scan sequence is to wait until all keys are unpressed. The state machine grounds all columns and waits until all rows read back '1'. When all keys are deteceted as unpressed, the scan logic will then continue scanning with the next column in sequence.

For multi-press scanning (CFG.SINGLEPRESS set to MULTIPRESS), scanning will proceed with the next column immediately when the IF.KEY interrupt is cleared. In multi-press mode, the IF.NOKEY interrupt flag will set once after a full scan cycle is completed without detecting a new key press, to indicate that all keys have been released and allow for subsequent press detections. Scans without a key press will not set this flag again until one or more new key presses have been detected.

Finally, the IF.SCANNED interrupt flag is set every time the scan logic completes an entire column-by-column scan without detecting any key presses.

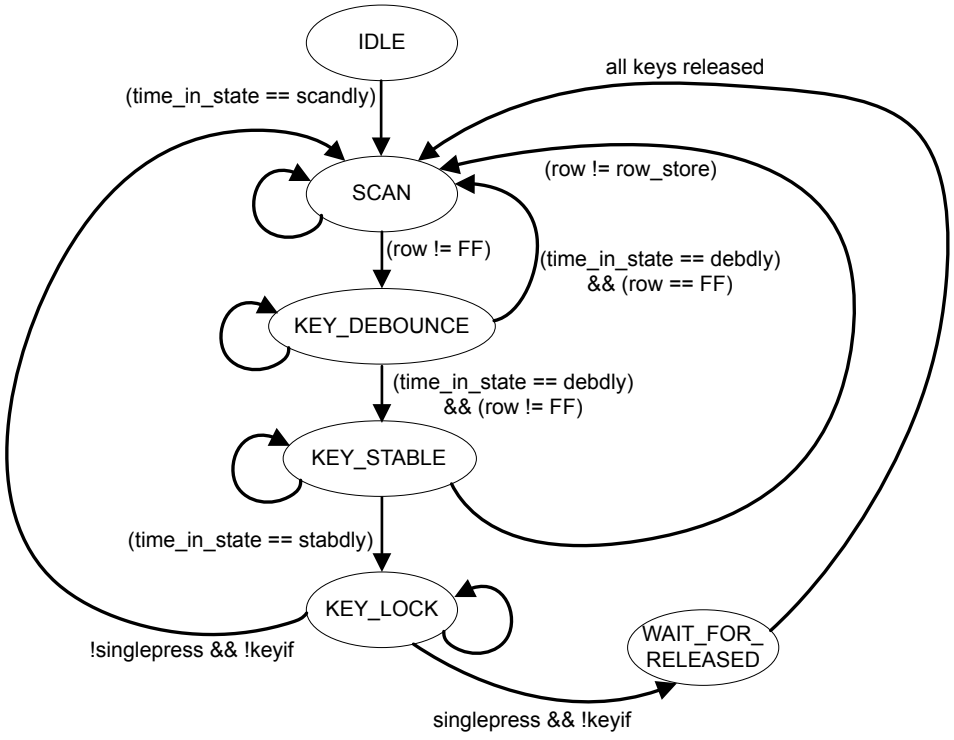


Figure 31.2. KEYSCAN State Machine

Figure 31.3 Single Press Mode on page 1233 shows how the scanner operates in single-press mode. One interrupt is generated upon key press. The usage model is for a single key press at a time. If multiple keys are pressed at the same time, it stops scanning at the first column that has a key press and shows all the pressed rows for that column. Scanning does not resume until all pressed keys are released.

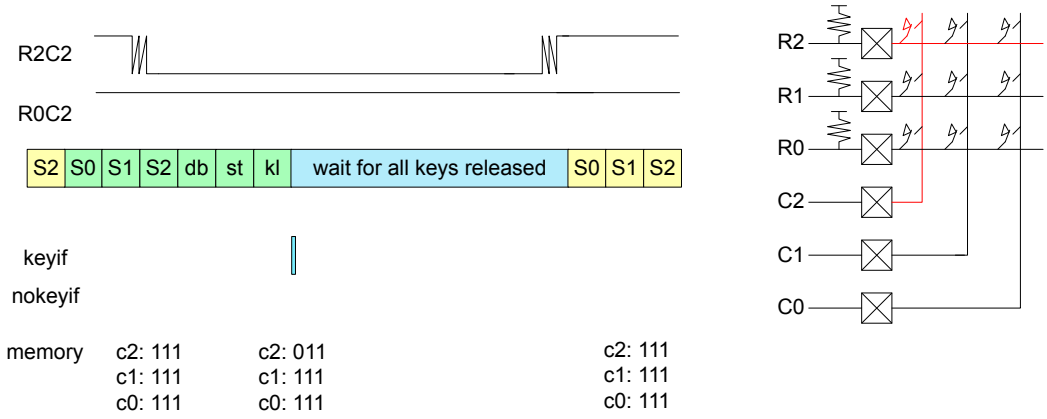


Figure 31.3. Single Press Mode

Figure 31.4 Multi-touch Mode with Multiple Key Presses on page 1233 shows two key presses that occur on the same column (R2C2 and R0C2). The first key press R2C2 was detected first and set the IF.KEY interrupt flag. The interrupt service routine read the status, updated an array in memory, and cleared the IF.KEY. Then the scanning continued until the IF.KEY was set again. Since both pressed keys are active on the same column, STATUS.ROW shows 010 for the low true key presses on R2C2 and R0C2. When both keys have been released and the scan cycles through without detecting any key presses, the IF.NOKEY flag is set.

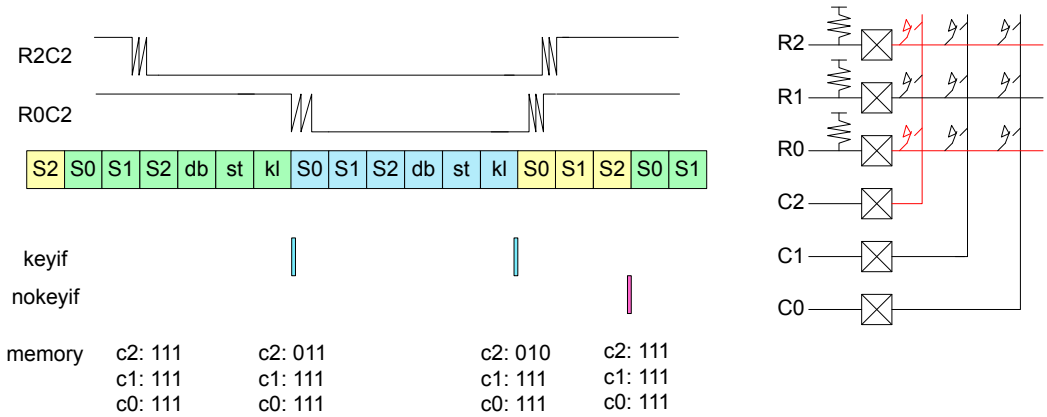


Figure 31.4. Multi-touch Mode with Multiple Key Presses

31.3.4 Wake on Key

The KEYSCAN module does not actively scan channels during EM2 or EM3, but it will operate in a wake-on-key mode when the IEN.WAKEUP interrupt is enabled. In this mode, all columns are driven low, and the logic waits for any of the row inputs to be pulled low. This event can wake the device from EM2 or EM3.

After waking up, the IF.WAKEUP flag will be set. Software may service this wakeup interrupt and start a new scan using the CMD.START command. Alternatively, if CMD.AUTOSTART has been used prior to sleeping, the keyscan will automatically be started each time the chip wakes up to EM0. Before going to sleep, software may use CMD.KEYSCANSTOP to turn off scanning if desired. If KEYSCANSTOP is not used, the keyscan module will still power down and stop scanning once the chip enters sleep. While most of the keyscan is powered down during EM2 and EM3, there is a small piece of logic that remains powered during EM2 and EM3 to allow a key press to wake the device.

A key press during sleep is unqualified, because no debouncing and settling are performed before waking the device. It is possible that noise in the mechanical keypad caused the wakeup without a real key press. In this case, after wakeup and scanning all columns, the IF.SCANNED interrupt comes in handy. IF.SCANNED would be set without IF.KEY or IF.NOKEY interrupts ever having been set. The IF.WAKEUP can be used to record to memory that the keypad has not yet completed any scans. Once IF.SCANNED fires without having seen any IF.KEY interrupts, this may indicate that the wakeup was due to noise instead of a key press, and the system can return to sleep.

31.4 KEYSCAN Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	KEYSCAN_IPVERSION	R	IPVERSION
0x004	KEYSCAN_EN	RW ENABLE	Enable
0x008	KEYSCAN_SWRST	RW SWRST	Software Reset
0x00C	KEYSCAN_CFG	RW CONFIG	Config
0x010	KEYSCAN_CMD	W SYNC	Command
0x014	KEYSCAN_DELAY	RW CONFIG	Delay
0x018	KEYSCAN_STATUS	RH	Status
0x01C	KEYSCAN_IF	RWH INTFLAG	Interrupt Flags
0x020	KEYSCAN_IEN	RW	Interrupt Enables
0x1000	KEYSCAN_IPVERSION_SET	R	IPVERSION
0x1004	KEYSCAN_EN_SET	RW ENABLE	Enable
0x1008	KEYSCAN_SWRST_SET	RW SWRST	Software Reset
0x100C	KEYSCAN_CFG_SET	RW CONFIG	Config
0x1010	KEYSCAN_CMD_SET	W SYNC	Command
0x1014	KEYSCAN_DELAY_SET	RW CONFIG	Delay
0x1018	KEYSCAN_STATUS_SET	RH	Status
0x101C	KEYSCAN_IF_SET	RWH INTFLAG	Interrupt Flags
0x1020	KEYSCAN_IEN_SET	RW	Interrupt Enables
0x2000	KEYSCAN_IPVERSION_CLR	R	IPVERSION
0x2004	KEYSCAN_EN_CLR	RW ENABLE	Enable
0x2008	KEYSCAN_SWRST_CLR	RW SWRST	Software Reset
0x200C	KEYSCAN_CFG_CLR	RW CONFIG	Config
0x2010	KEYSCAN_CMD_CLR	W SYNC	Command
0x2014	KEYSCAN_DELAY_CLR	RW CONFIG	Delay
0x2018	KEYSCAN_STATUS_CLR	RH	Status
0x201C	KEYSCAN_IF_CLR	RWH INTFLAG	Interrupt Flags
0x2020	KEYSCAN_IEN_CLR	RW	Interrupt Enables
0x3000	KEYSCAN_IPVERSION_TGL	R	IPVERSION
0x3004	KEYSCAN_EN_TGL	RW ENABLE	Enable
0x3008	KEYSCAN_SWRST_TGL	RW SWRST	Software Reset
0x300C	KEYSCAN_CFG_TGL	RW CONFIG	Config
0x3010	KEYSCAN_CMD_TGL	W SYNC	Command
0x3014	KEYSCAN_DELAY_TGL	RW CONFIG	Delay
0x3018	KEYSCAN_STATUS_TGL	RH	Status
0x301C	KEYSCAN_IF_TGL	RWH INTFLAG	Interrupt Flags

Offset	Name	Type	Description
0x3020	KEYSCAN_IEN_TGL	RW	Interrupt Enables

31.5 KEYSCAN Register Description

31.5.1 KEYSCAN_IPVERSION - IPVERSION

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IPVERSION**

31.5.2 KEYSCAN_EN - Enable

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status** When EN is cleared, DISABLING status is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and bit fields or registers dependent on hardware state, such as INTFLAGS and FIFOs.
0	EN	0x0	RW	**Enable** Enable

Value	Mode	Description
0	DISABLE	Stops clocking and resets peripheral core logic.
1	ENABLE	Enables clocking, and begins scanning if CFG.AUTOSTART is 0x1.

31.5.3 KEYSCAN_SWRST - Software Reset

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	W
Name																															RESETTING	SWRST

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESETTING	0x0	R	**Software reset busy status**
	Software reset busy status			
0	SWRST	0x0	W	**Software reset command**
	Software reset command			

31.5.4 KEYSCAN_CFG - Config

Offset	Bit Position																															
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0x2				0x5				0x0		0x0			0x1387F																	
Access		RW				RW				RW		RW			RW																	
Name		NUMCOLS				NUMROWS				AUTOSTART		SINGLEPRESS			CLKDIV																	

Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
30:28	NUMCOLS	0x2	RW	**Number of Columns**
	Number of Columns minus one (0x2 indicates 3 columns)			
27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26:24	NUMROWS	0x5	RW	**Number of Rows**
	Number of Rows minus 1(5: represents 6 rows)			

Value	Mode	Description
0	RSV1	1 Row is not supported; defaults to 3 instead
1	RSV2	2 Rows are not supported; defaults to 3 instead
2	ROW3	3 Rows
3	ROW4	4 Rows
4	ROW5	5 Rows
5	ROW6	6 Rows

Bit	Name	Reset	Access	Description
23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22	AUTOSTART	0x0	RW	**Automatically Start**
	Automatically start when EN is set to 0x1.			

Value	Mode	Description
0	AUTOSTARTDIS	Auto start is disabled
1	AUTOSTARTEN	Auto start is enabled

Bit	Name	Reset	Access	Description
21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20	SINGLEPRESS	0x0	RW	**Single Press**
	Generates one interrupt per keypress. Then waits for all keys to be released and IF.KEYIF=0x0 before it starts scanning again. When singlepress is not set, scanning continues once IF.KEYIF=0x0 which allows for muti-touch detection.			

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	MULTIPRESS		After KEYIF is set and then cleared, scanning will continue. This can give multiple interrupts for the same key press, but allow multiple key presses to be detected. To use this mode for multi-key detection, the ISR should update a section of memory of COLNUM bytes on each interrupt, until key release is detected. After key release, the section of memory where key presses are recorded can be processed.
	1	SINGLEPRESS		After KEYIF has been set and cleared, it will not set again until no key press is detected. This allows faster response since the ISR can start processing data as soon as the KEYIF is set.
19:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17:0	CLKDIV	0x1387F	RW	**Clock Divider**
	Divides the peripheral clock by (CLKDIV+1) to produce a timebase for the keyscan timers. The nominal target timebase is 2 ms.			

31.5.5 KEYSCAN_CMD - Command

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															W(nB)	W(nB)
Name																															KEYSCANSTOP	KEYSCANSTART

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	KEYSCANSTOP	0x0	W(nB)	**Keyscan Stop**
	Stop Keyscan			
0	KEYSCANSTART	0x0	W(nB)	**Keyscan Start**
	Start Keyscan			

31.5.6 KEYSCAN_DELAY - Delay

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0								0x0								0x0											
Access					RW								RW								RW											
Name					STABDLY								DEBDLY								SCANDLY											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:24	STABDLY Row stable delay	0x0	RW	**Row stable Delay**

Value	Mode	Description
0	STABDLY2	2ms Row Stable Delay
1	STABDLY4	4ms Row Stable Delay
2	STABDLY6	6ms Row Stable Delay
3	STABDLY8	8ms Row Stable Delay
4	STABDLY10	10ms Row Stable Delay
5	STABDLY12	12ms Row Stable Delay
6	STABDLY14	14ms Row Stable Delay
7	STABDLY16	16ms Row Stable Delay
8	STABDLY18	18ms Row Stable Delay
9	STABDLY20	20ms Row Stable Delay
10	STABDLY22	22ms Row Stable Delay
11	STABDLY24	24ms Row Stable Delay
12	STABDLY26	26ms Row Stable Delay
13	STABDLY28	28ms Row Stable Delay
14	STABDLY30	30ms Row Stable Delay
15	STABDLY32	32ms Row Stable Delay

23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	DEBDLY Debounce Delay	0x0	RW	**Debounce Delay**

Value	Mode	Description
0	DEBDLY2	2ms Debounce Delay
1	DEBDLY4	4ms Debounce Delay

Bit	Name	Reset	Access	Description
	2	DEBDLY6		6ms Debounce Delay
	3	DEBDLY8		8ms Debounce Delay
	4	DEBDLY10		10ms Debounce Delay
	5	DEBDLY12		12ms Debounce Delay
	6	DEBDLY14		14ms Debounce Delay
	7	DEBDLY16		16ms Debounce Delay
	8	DEBDLY18		18ms Debounce Delay
	9	DEBDLY20		20ms Debounce Delay
	10	DEBDLY22		22ms Debounce Delay
	11	DEBDLY24		24ms Debounce Delay
	12	DEBDLY26		26ms Debounce Delay
	13	DEBDLY28		28ms Debounce Delay
	14	DEBDLY30		30ms Debounce Delay
	15	DEBDLY32		32ms Debounce Delay
15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:8	SCANDLY Scan Delay	0x0	RW	**Scan Delay**
	Value	Mode		Description
	0	SCANDLY2		2ms Scan Delay
	1	SCANDLY4		4ms Scan Delay
	2	SCANDLY6		6ms Scan Delay
	3	SCANDLY8		8ms Scan Delay
	4	SCANDLY10		10ms Scan Delay
	5	SCANDLY12		12ms Scan Delay
	6	SCANDLY14		14ms Scan Delay
	7	SCANDLY16		16ms Scan Delay
	8	SCANDLY18		18ms Scan Delay
	9	SCANDLY20		20ms Scan Delay
	10	SCANDLY22		22ms Scan Delay
	11	SCANDLY24		24ms Scan Delay
	12	SCANDLY26		26ms Scan Delay
	13	SCANDLY28		28ms Scan Delay
	14	SCANDLY30		30ms Scan Delay
	15	SCANDLY32		32ms Scan Delay
7:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

31.5.7 KEYSCAN_STATUS - Status

Offset	Bit Position																															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x1				0x0										0x0											0x0					
Access	R	R				R										R											R					
Name	SYNCBUSY	NOKEY				COL										RUNNING											ROW					

Bit	Name	Reset	Access	Description
31	SYNCBUSY	0x0	R	**Sync Busy**
	Sync Busy			
30	NOKEY	0x1	R	**No Key pressed status**
	No Key pressed status. Requires EN=1 when reading.			
29:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26:24	COL	0x0	R	**Column Latched**
	column that corresponds to key press			
23:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	RUNNING	0x0	R	**Running**
	Running status. Requires EN=1 when reading.			
15:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:0	ROW	0x0	R	**Row detection**
	Rows (1: pull up for no key press, 0: key press)			

31.5.8 KEYSCAN_IF - Interrupt Flags

Offset	Bit Position																															
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													RW	RW	RW	RW
Name																													WAKEUP	SCANNED	KEY	NOKEY

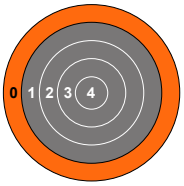
Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	WAKEUP	0x0	RW	**Wake up**
	During sleep all columns are grounded and any keypress on any row will cause a wakeup and set WAKEUP. During EM01 WAKEUP will not get set.			
2	SCANNED	0x0	RW	**Completed scan**
	SCANNED sets each time all columns are scaned and no keypress is detected. The start of the scan is either from column 0 (after reset), or from the column of the last keypress.			
1	KEY	0x0	RW	**A key was pressed**
	A key was pressed (This stops scan until the interrupt flag is cleared)			
0	NOKEY	0x0	RW	**No key was pressed**
	After a keypress is detected and IF.KEY interrupt cleared, NOKEY will set one time, after all the columns have been checked without detecting a keypress.			

31.5.9 KEYSCAN_IEN - Interrupt Enables

Offset	Bit Position																															
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													RW	RW	RW	RW
Name																													WAKEUP	SCANNED	KEY	NOKEY

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	WAKEUP	0x0	RW	**Wake up** A key press was detected during the first complete scan. This could be either after wake up from sleep or during the first complete scan after the first time the KEYSCAN module was enabled.
2	SCANNED	0x0	RW	**Completed Scanning** SCANNEDIF sets each time all columns are scaned and no keypress is detected. The start of the scan is either from column 0 (after reset), or from the column of the last keypress.
1	KEY	0x0	RW	**A Key was pressed** A Key was pressed (This stops scan until the interrupt flag is cleared)
0	NOKEY	0x0	RW	**No Key was pressed** After keypress is detected and keyif interrupt cleared, the NOKEYIF will set one time once all the columns are checked with no keypress.

32. MVP - Matrix Vector Processor



Quick Facts

What?

The Matrix Vector Processor accelerates floating point operations, particularly matrixed complex floating point multiplications and additions

Why?

The Matrix Vector Processor hardware can save energy by offloading heavily floating-point computational intensive operations, including the Angle-of-Arrival (AoA) MUSIC algorithm, Machine Learning (ML) and linear algebra

How?

The Matrix Vector Processor includes a dedicated hardware arithmetic logic unit (ALU), load/store unit (LSU), and sequencer.

32.1 Introduction

The Matrix Vector Processor (MVP) is designed to offload the major computationally intensive floating point operations, particularly matrixed complex floating point multiplications and additions. The MVP hardware supports the acceleration of the key Angle-of-Arrival (AoA) MUSIC (MUltiple SIgnal Classification) algorithm computations, as well as other heavily floating-point computational problems such as Machine Learning (ML) or linear algebra.

32.2 Features

- Instruction Set Architecture (ISA)
  - General purpose instruction set tailored towards algorithms built out of ALU, loop, and load/store instructions
  - Enables many high-level array functions, e.g.:
    - Matrix multiplication
    - Element-wise matrix multiplication
    - Matrix addition
    - Power series generation
    - Convolution
  - Program flexibility allows efficient iteration over N-dimensional array elements, including in-place processing of special ,atrix views:
    - Element-wise negate / conjugate
    - Transpose / adjoint / reverse
    - Matrix blocks (i.e., rectangular parts of matrix)
    - Matrix slices (i.e., taking rows, columns, or elements uniformly spaced within a matrix)
    - Row-major or column-major ordering
- Arithmetic Logic Unit (ALU)
  - Three 32-bit floating-point input operands, interpreted as real or complex numbers
    - Partial integer input support
      - One 32-bit floating-point output operands, interpreted as real or complex numbers
      - Register bank to hold all input/output operands
    - Includes 8 registers for temporary storage and/or accumulation
  - Hardware to support 1 complex floating point multiply-accumulate (MAC) per cycle
    - Four single-precision floating-point multipliers
    - Four single-precision floating-point adders
    - 6x performance of Cortex M33 FMAC operations
  - Operations supported at a rate of one operation per cycle:
    - Complex addition, multiplication, and MAC operations
    - Parallel real multiplication and MAC
    - Parallel real addition
    - Sum of 4 reals
    - Squared-magnitude of complex/real
    - Integer-to-float conversion
    - Conditional computation
  - Input transformations (per real/complex part of each input)
    - Negation (complex conjugate)
    - Zero-masking (real/imaginary part decomposition)
- Load/Store Unit (LSU)
  - Controls data streaming from memory-to-ALU and vice versa
  - Pipelined architecture to support two simultaneous 32-bit memory reads and one 32-bit memory write per cycle
  - Supports signed / unsigned 8-bit integer conversion for both load and store operations
  - First-party DMA ports
    - Used by load / store unit for handling accesses to external (system) memory addresses
    - Three independent 32-bit AHB manager ports for supporting 2 read channels and 1 write channel simultaneously
- Sequencer
  - Coordinates all MVP blocks to execute a sequence of instructions provided via the programming interface
  - Handles array iteration according to instruction sequence and static array configuration
  - Handles loop iteration according to instruction sequence and static loop configuration

- Programming interface
  - Control registers for starting / stopping engine
  - Status registers about ongoing and finished instruction sequences
    - Fault status
    - Useful information for debug
  - Breakpoint and stepping controls for debug
  - Interrupts and faults
    - Instruction sequence completion
    - Bus faults
    - Loop faults
    - Array faults
  - Array configuration registers
  - Loop configuration registers
  - Instruction queue registers
    - Array iteration
    - ALU operations
    - Looping

32.3 Functional Description

The Matrix Vector Processor (MVP) is a peripheral processor that can be used to accelerate the processing of floating point operations while offloading the primary CPU. At a high level, it consists of:

- A register interface for programming and controlling operations
- A sequencer (which includes the loop controllers) that manages execution of the program
- Array and bus controllers that manage addressing, loading, and storing of data in arrays stored in system memory
- The pipeline controller, ALU, and ALU register bank for processing data

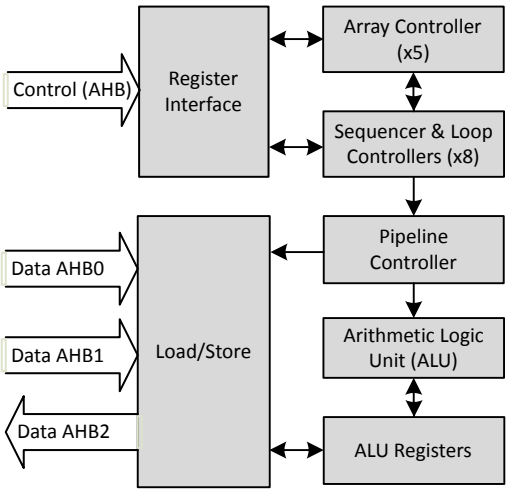


Figure 32.1. MVP Block Diagram

For most operations, software will program the MVP to address matrix (array) data in system memory and then process this data to perform a useful computation. The MVP provides three primary resources for controlling the operations that are fed through the ALU:

- Eight instructions, each of which encodes an ALU operation, load/store controls for the cycle, array increment controls, and loop controls
- Eight loop controllers, which can be used to form loops around a single or multiple instructions and can be nested to form complex sequences of ALU operations
- Five array controllers, each of which configure and control access to an independent matrix of data in system memory. The MVP supports arrays of up to 3 dimensions and each dimension can be independently incremented by the load/store streams, instruction, or loop controller on any given cycle

All of the MVP's control registers reside within its own register space, including the eight instructions (each of which is three 32-bit words). The MVP's register space has been organized such that DMA can be used to program all the configuration registers sequentially and initiate an operation with minimal CPU intervention. Once the program has successfully completed (by reaching an instruction with ENDPROG while completing all outstanding loops), the MVP interrupts the CPU with the PROGDONE interrupt. In the case of a fatal error (usually misprogramming), the MVP issues an error interrupt and terminate processing immediately.

33. Revision History

Revision 1.1

January, 2024

- 3.3.3 Interrupt Request Lines (IRQ): Added ULFRCO IRQ.
- 5.8.8 MSC_STATUS - Status Register: Fixed TIMEOUT, PENDING, and ERASEABORTED bit descriptions.
- 8.4.3.2 Lock Modes: Updated language regarding DPLL phase lock mode.
- 23. GPIO - General Purpose Input/Output: Updated references to GPIO interrupt flag clear and set registers.
- 24.3.7.2 SYNC Descriptor Structure: Added missing fields in SRC and DST words.
- 24.3.7.3 WRI Descriptor Structure: Added missing field in SRC word.
- 28.3.1 Interface Descriptions: Updated note clarifying the limitations of the VDAC interface and its use during idle phases.
- 28.3.5 Sensor Interaction: VDAC excitation note updated to specify use only with VDAC0.CH0_MAIN_OUT and also how its low impedance can be used to damp LC oscillation during idle phases.
- 30. ACMP - Analog Comparator, 28. LESENSE - Low Energy Sensor Interface: Deprecated capacitive sensing. Removed capacitive sense mode chapter, removed register description mentions of capacitive sensing, and removed other references of capacitive sensing throughout document. This feature is not recommended for new designs.

Revision 1.0

August, 2023

- 5.4.2.2 DEVINFO_PART - Part Info: Added SG family to FAMILY field.
- 5.6.6.4 SYSCFG_CHIPREVHW - Chip Revision, Hard-Wired: Corrected PARTNUMBER field reset value.
- 5.6.7 MPAHBRAM Register Map: Removed internal configuration registers from map.
- 7.5.25 CMU_LCDCLKCTRL - LCD Clock Control: Added register details.
- 11.3.8.3 Buck DC-DC Recommended Configuration Settings: Corrected recommended setting for DCDC_CTRL.IPKTMAXCTRL.
- 11.3.12.1 Linearization, Offset Correction, and Calibration: Added polynomial coefficients.
- 11.7 DCDC Register Description:
  - Updated register enumerations in EM01CTRL0.DRVSPEED and EM23CTRL0.DRVSPEED - removed settings other than default (no benefit to using other settings).
  - Corrected EM23CTRL0.IPKVAL enumeration options.
  - Added PFXMCTRL.IPKVAL enumerations.
- 22. IADC - Incremental Analog to Digital Converter: Normalized the names of clocks used inside the block.
- 30.3.2 Warmup Time: Added note about additional warm-up time required when BIAS <= 3.
- Transitioned to new block diagram format and color scheme.

Revision 0.1

March, 2023

Initial Release.

Appendix 1. Abbreviations

This section lists abbreviations used in this document.

Table 1.1. Abbreviations

Abbreviation	Description
ADC	Analog to Digital Converter
AES	Advanced Encryption Standard
AFC	Automatic Frequency Control
AGC	Automatic Gain Control
AHB	AMBA Advanced High-performance Bus. AMBA is short for "Advanced Microcontroller Bus Architecture".
APB	AMBA Advanced Peripheral Bus. AMBA is short for "Advanced Microcontroller Bus Architecture".
APC	Automatic Power Control
ASK	Amplitude Shift Keying
BLE	Bluetooth Low Energy
BLE-LR	Bluetooth Low Energy Long Range
BR	Baud Rate
BT	Bandwidth Time product
BUFC	Buffer Controller
BW	Bandwidth
CBC	Cipher Block Chaining (AES mode of operation)
CBC-MAC	Cipher Block Chaining - Message Authentication Code (AES mode of operation)
CC	Compare / Capture
CCA	Clear Channel Assessment
CFB	Cipher Feedback (AES mode of operation)
CHF	Channel Filter
CLK	Clock
CM3	ARM Cortex-M3
CM4	ARM Cortex-M4
CMD	Command
CMU	Clock Management Unit
CRC	Cyclic Redundancy Check
CTR	Counter mode (AES mode of operation)
CTRL	Control
DBG	Debug
DC	Direct Current
DEC	Decimator
DEMOD	Demodulator

Abbreviation	Description
DSA	Detection of Signal Arrival
DSSS	Direct Sequence Spread Spectrum
ECB	Electronic Code Book (AES mode of operation)
EFM32	Energy Friendly Microcontroller
EFR32	Wireless Gecko
EM	Energy Mode
EMU	Energy Management Unit
FEC	Forward Error Correction
FIR	Finite Impulse Response
FRC	Frame Controller
FSK	Frequency Shift Keying
FSM	Finite State Machine
GFSK	Gaussian Frequency Shift Keying
GMSK	Gaussian Minimum Shift Keying
GPIO	General Purpose Input / Output
HFRCO	High Frequency RC Oscillator
HFXO	High Frequency Crystal Oscillator
HW	Hardware
Hz	Hertz
IF	Intermediate Frequency
IFADC	Intermediate Frequency Analog to Digital Converter
ISR	Interrupt Service Routine
LFRCO	Low Frequency RC Oscillator
LFXO	Low Frequency Crystal Oscillator
LNA	Low Noise Amplifier
LO	Local Oscillator
MOD	Modulator
MODEM	Modulator and Demodulator
MSK	Minimum Shift Keying
NRZ	Non Return to Zero
NVIC	Nested Vector Interrupt Controller
OFB	Output Feedback Mode (AES mode of operation)
OOK	On Off Keying
OQPSK	Offset Quadrature Phase Shift Keying
OSR	Over-Sampling Ratio
PA	Power Amplifier
PD	Power Down

Abbreviation	Description
PHY	Physical Layer
PROTIMER	Protocol Timer
PRS	Peripheral Reflex System
PWM	Pulse Width Modulation
RAC	Radio Controller
RAM	Random Access Memory
RF	Radio Frequency
RMU	Reset Management Unit
RSM	Radio State Machine
RSSI	Received Signal Strength Indicator
RTC	Real Time Counter
RX	Receive
SEQ	Radio Sequencer
SPI	Serial Peripheral Interface
SRC	Sample Rate Converter
STIMER	Sequencer Timer
SW	Software
SYNTH	Synthesizer
TX	Transmit
XTAL	Crystal

Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support & Community
www.silabs.com/community

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

www.silabs.com