

EFM32G Reference Manual

Gecko Series

- 32-bit ARM Cortex-M3 processor running at up to 32 MHz
- Up to 128 kB Flash and 16 kB RAM memory
- Energy efficient and autonomous peripherals
- Ultra low power Energy Modes with sub-µA operation
- Fast wake-up time of only 2 μs

The EFM32G microcontroller series revolutionizes the 8- to 32-bit market with a combination of unmatched performance and ultra low power consumption in both active- and sleep modes. EFM32G devices consume as little as 180 μ A/MHz in run mode, and as little as 900 nA with a Real Time Counter running, Brown-out and full RAM and register retention.

EFM32G's low energy consumption outperforms any other available 8-, 16-, and 32-bit solution. The EFM32G includes autonomous and energy efficient peripherals, high overall chip- and analog integration, and the performance of the industry standard 32-bit ARM Cortex-M3 processor.



















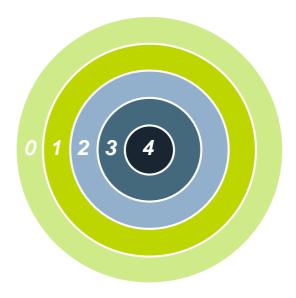
1 Energy Friendly Microcontrollers

1.1 Typical Applications

The EFM32G microcontroller is the ideal choice for demanding 8-, 16-, and 32-bit energy sensitive applications. These devices are developed to minimize the energy consumption by lowering both the power and the active time, over all phases of MCU operation. This unique combination of ultra low energy consumption and the performance of the 32-bit ARM Cortex-M3 processor, help designers get more out of the available energy in a variety of applications.

Ultra low energy EFM32G microcontrollers are perfect for:

- · Gas metering
- · Energy metering
- Water metering
- · Smart metering
- · Alarm and security systems
- Health and fitness applications
- Industrial and home automation



1.2 EFM32G Development

Because EFM32G use the Cortex-M3 CPU, embedded designers benefit from the largest development ecosystem in the industry, the ARM ecosystem. The development suite spans the whole design process and includes powerful debug tools, and some of the world's top brand compilers. Libraries with documentation and user examples shorten time from idea to market.

The range of EFM32G devices ensure easy migration and feature upgrade possibilities.



2 About This Document

This document contains reference material for the EFM32G series of microcontrollers. All modules and peripherals in the EFM32G series 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 pin-out, are covered in the device-specific datasheets.

2.1 Conventions

Register Names

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

TIMERn_CTRL - Control Register

The "n" denotes the numeric instance for modules that might have 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,

where x denotes the port instance (A,B,...).

Bit Fields

Registers contain one or more bit fields which can be 1 to 32 bits wide. Multi-bit fields are denoted with (x:y), where x is the start bit and y is the end bit.

Address

The address for each register can be found by adding the base address of the module (found in the Memory Map), 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 2.1 (p. 3).

Table 2.1. 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.
RW1H	Readable, writable and updated by hardware. 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.

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 0 unless otherwise stated in the Register Description. Reserved bits might be read as 1 in future devices.

Reset Value

The reset value denotes the value after reset.

Registers denoted with X have an unknown reset value and need to be initialized before use. Note that, before these registers are initialized, read-modify-write operations might result in undefined register values.

Pin Connections

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

USn_TX (USARTn TX pin)

The pin locations referenced in this document are given in the device-specific datasheet.

2.2 Related Documentation

Further documentation on the EFM32G family and the ARM Cortex-M3 can be found at the Silicon Laboratories and ARM web pages:

www.silabs.com

www.arm.com



3 System Overview

3.1 Introduction

The EFM32 MCUs are the world's most energy friendly microcontrollers. With a unique combination of the powerful 32-bit ARM Cortex-M3, innovative low energy techniques, short wake-up time from energy saving modes, and a wide selection of peripherals, the EFM32G microcontroller is well suited for any battery operated application, as well as other systems requiring high performance and low-energy consumption, see Figure 3.1 (p. 5) .

3.2 Block Diagram

Figure 3.1 (p. 5) shows the block diagram of EFM32G. The color indicates peripheral availability in the different energy modes, described in Section 3.4 (p. 7).

Figure 3.1. Block Diagram of EFM32G

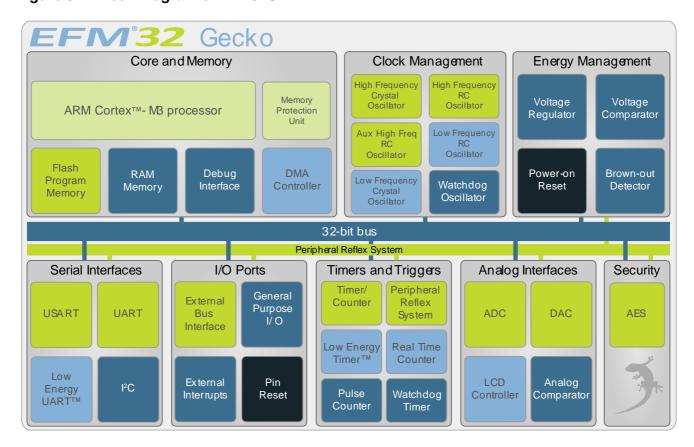
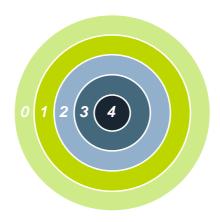


Figure 3.2. Energy Mode Indicator





Note

In the energy mode indicator, the numbers indicates Energy Mode, i.e EM0-EM4.

3.3 Features

3.3.1 MCU Features

ARM Cortex-M3 CPU platform

- High Performance 32-bit processor @ up to 32 MHz
- Memory Protection Unit
- · Wake-up Interrupt Controller

• Flexible Energy Management System

- 20 nA @ 3 V Shutoff Mode
- 0.6 μA @ 3 V Stop Mode, including Power-on Reset, Brown-out Detector, RAM and CPU retention
- 0.9 μA @ 3 V Deep Sleep Mode, including RTC with 32768 Hz oscillator, Power-on Reset, Brown-out Detector, RAM and CPU retention
- 45 μA/MHz @ 3 V Sleep Mode
- 180 μA/MHz @ 3 V Run Mode, with code executed from flash
- 128/64/32/16 KB Flash
- 16/8 KB RAM
- Up to 90 General Purpose I/O pins
 - Configurable push-pull, open-drain, pull-up/down, input filter, drive strength
 - Configurable peripheral I/O locations
 - 16 asynchronous external interrupts
- 8 Channel DMA Controller
 - Alternate/primary descriptors with scatter-gather/ping-pong operation
- 8 Channel Peripheral Reflex System
 - · Autonomous inter-peripheral signaling enables smart operation in low energy modes
- External Bus Interface (EBI)
 - Up to 4x64 MB of external memory mapped space
- Integrated LCD Controller for up to 4×40 Segments
 - · Voltage boost, adjustable contrast adjustment and autonomous animation feature
- Hardware AES with 128/256-bit Keys in 54/75 cycles
- Communication interfaces
 - 3x Universal Synchronous/Asynchronous Receiver/Transmitter
 - UART/SPI/SmartCard (ISO 7816)/IrDA
 - Triple buffered full/half-duplex operation
 - 4-16 data bits
 - 1x Universal Asynchronous Receiver/Transmitter
 - Triple buffered full/half-duplex operation
 - 8-9 data bits
 - 2x Low Energy UART
 - Autonomous operation with DMA in Deep Sleep Mode
 - 1x I²C Interface with SMBus support
 - Address recognition in Stop Mode
- Timers/Counters
 - 3x 16-bit Timer/Counter
 - 3 Compare/Capture/PWM channels
 - Dead-Time Insertion on TIMER0
 - 16-bit Low Energy Timer
 - 24-bit Real-Time Counter
 - 3x 8-bit Pulse Counter



- · Asynchronous pulse counting/quadrature decoding
- Watchdog Timer with dedicated RC oscillator @ 50 nA
- Ultra low power precision analog peripherals
 - 12-bit 1 Msamples/s Analog to Digital Converter
 - · 8 input channels and on-chip temperature sensor
 - · Single ended or differential operation
 - Conversion tailgating for predictable latency
 - 12-bit 500 ksamples/s Digital to Analog Converter
 - 2 single ended channels/1 differential channel
 - 2x Analog Comparator
 - Programmable speed/current
 - · Capacitive sensing with up to 8 inputs
 - Supply Voltage Comparator

3.3.2 System Features

- Ultra efficient Power-on Reset and Brown-Out Detector
- 2-pin Serial Wire Debug interface
 - 1-pin Serial Wire Viewer
- Temperature range -40 85°C
- Single power supply 1.98 3.8 V
- Packages
 - QFN32
 - QFN64
 - TQFP48
 - TQFP64
 - LQFP100
 - LFBGA112
 - · Full wafer

3.4 Energy Modes

There are five different Energy Modes (EM0-EM4) in the EFM32G, see Table 3.1 (p. 8). The EFM32G is designed to achieve a high degree of autonomous operation in low energy modes. The intelligent combination of peripherals, RAM with data retention, DMA, low-power oscillators, and short wake-up time, makes it attractive to remain in low energy modes for long periods and thus saving energy consumption.

Tip

Throughout this document, the first figure in every module description contains an Energy Mode Indicator showing which energy mode(s) the module can operate (see Table 3.1 (p. 8)).



Table 3.1. Energy Mode Description

Energy Mode Name Description



EM0 – Energy Mode 0 (Run mode)

In EM0, the CPU is running and consuming as little as 180 μ A/MHz, when running code from flash. All peripherals can be active.



EM1 – Energy Mode 1 (Sleep Mode) In EM1, the CPU is sleeping and the power consumption is only 45 μ A/MHz. All peripherals, including DMA, PRS and memory system, are still available.



EM2 – Energy Mode 2 (Deep Sleep Mode) In EM2 the high frequency oscillator is turned off, but with the 32.768 kHz oscillator running, selected low energy peripherals (LCD, RTC, LETIMER, PCNT, LEUART, $\rm l^2C$, WDOG and ACMP) are still available. This gives a high degree of autonomous operation with a current consumption as low as 0.9 μA with RTC enabled. Power-on Reset, Brown-out Detection and full RAM and CPU retention is also included.



EM3 - Energy Mode 3 (Stop Mode)

In EM3, the low-frequency oscillator is disabled, but there is still full CPU and RAM retention, as well as Power-on Reset, Pin reset and Brown-out Detection, with a consumption of only 0.6 μA . The low-power ACMP, asynchronous external interrupt, PCNT, and I^2C can wake-up the device. Even in this mode, the wake-up time is a few microseconds.



EM4 – Energy Mode 4 (Shutoff Mode) In EM4, the current is down to 20 nA and all chip functionality is turned off except the pin reset and the Power-On Reset. All pins are put into their reset state.

3.5 Product Overview

Table 3.2 (p. 8) shows a device overview of the EFM32G Microcontroller Series, including peripheral functionality. For more information, the reader is referred to the device specific datasheets.

Table 3.2. EFM32G Microcontroller Series

EFM32G Part #	Flash	RAM	GPIO(pins)	ГСБ	USART+UART	LEUART	l²c	Timer(PWM)	LETIMER	RTC	PCNT	Watchdog	ADC(pins)	DAC(pins)	ACMP(pins)	AES	EBI	Package
200F16	16	8	24	-	2	1	1	2 (6)	1	1	1	1	1 (4)	1 (1)	2 (5)	-	-	QFN32
200F32	32	8	24	-	2	1	1	2 (6)	1	1	1	1	1 (4)	1 (1)	2 (5)	-	-	QFN32
200F64	64	16	24	-	2	1	1	2 (6)	1	1	1	1	1 (4)	1 (1)	2 (5)	-	-	QFN32
210F128	128	16	24	-	2	1	1	2 (6)	1	1	1	1	1 (4)	1 (1)	2 (5)	Υ	-	QFN32



EFM32G Part#	Flash	RAM	GPIO(pins)	ГСД	USART+UART	LEUART	l²C	Timer(PWM)	LETIMER	RTC	PCNT	Watchdog	ADC(pins)	DAC(pins)	ACMP(pins)	AES	EBI	Package
230F32	32	8	56	-	3	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	-	QFN64
230F64	64	16	56	-	3	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (16)	Υ	-	QFN64
230F128	128	16	56	-	3	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	-	QFN64
280F32	32	8	85	-	3+1	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (16)	Υ	Υ	LQFP100
280F64	64	16	85	-	3+1	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y	LQFP100
280F128	128	16	85	-	3+1	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y	LQFP100
290F32	32	8	90	-	3+1	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (16)	Υ	Y	LFBGA112
290F64	64	16	90	-	3+1	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y	LFBGA112
290F128	128	16	90	-	3+1	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y	LFBGA112
800F128	128	16	90	4x40	3+1	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (16)	Υ	Y ¹	Wafer
840F32	32	8	56	4x24	3	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (8)	Y	-	QFN64
840F64	64	16	56	4x24	3	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (8)	Y	-	QFN64
840F128	128	16	56	4x24	3	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (8)	Y	-	QFN64
880F32	32	8	85	4x40	3+1	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (16)	Υ	Y 1	LQFP100
880F64	64	16	85	4x40	3+1	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y ¹	LQFP100
880F128	128	16	85	4x40	3+1	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y ¹	LQFP100
890F32	32	8	90	4x40	3+1	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (16)	Υ	Y ¹	LFBGA112
890F64	64	16	90	4x40	3+1	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y ¹	LFBGA112
890F128	128	16	90	4x40	3+1	2	1	3 (9)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y ¹	LFBGA112

¹EBI and LCD share pins in the part. Only a reduced pin count LCD driver can be used simultaneously with the EBI.

3.6 Device Revision

The device revision number is read from the ROM Table. The major revision number and the chip family number is read from PID0 and PID1 registers. The minor revision number is extracted from the PID2 and PID3 registers, as illustrated in Figure 3.3 (p. 10). The Fam[5:2] and Fam[1:0] must be combined to complete the chip family number, while the Minor Rev[7:4] and Minor Rev[3:0] must be combined to form the complete revision number.



Figure 3.3. Revision Number Extraction

PID2 (0xE00FFFE8)				
31:8	7:4	3:0		
	Minor Rev[7:4]			
	[]			

PID0 (0xE00FFFE0)					
31:8	7:6	5:0			
	Fam[1:0]	Major Rev[5:0]			

PID3 (0xE00FFFEC)				
31:8	7:4	3:0		
	Minor Rev[3:0]			

PID1 (0xE00F)	FFE4)
31:4	3:0
	Fam[5:2]

For the latest revision of the Gecko family, the chip family number is 0x00 and the major revision number is 0x01. The minor revision number is to be interpreted according to Table 3.3 (p. 10).

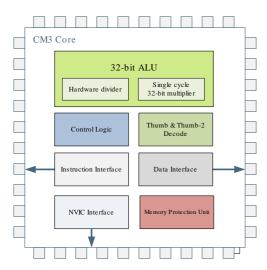
Table 3.3. Minor Revision Number Interpretation

Minor Rev[7:0]	Revision
0x00	A
0x01	В
0x02	С
0x03	D
0x04	Е



4 System Processor





Quick Facts

What?

The industry leading Cortex-M3 processor from ARM is the CPU in the EFM32G microcontrollers.

Why?

The ARM Cortex-M3 is designed for exceptional 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, the Cortex-M3 makes the EFM32G devices perfect for 8- to 32-bit applications. The processor is featuring a Harvard architecture, 3 stage pipeline, single cycle instructions, Thumb-2 instruction set support, and fast interrupt handling.

4.1 Introduction

The ARM Cortex-M3 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-M3 implemented is revision r2p0.

4.2 Features

- · Harvard Architecture
 - Separate data and program memory buses (No memory bottleneck as for a single-bus system)
- 3-stage pipeline
- Thumb-2 instruction set
 - · Enhanced levels of performance, energy efficiency, and code density
- · Single-cycle multiply and efficient divide instructions
 - 32-bit multiplication in a single cycle
 - Signed and unsigned divide operations between 2 and 12 cycles
- · Atomic bit manipulation with bit banding
 - · Direct access to single bits of data
 - Two 1MB bit banding regions for memory and peripherals mapping to 32MB alias regions
 - · Atomic operation which cannot be interrupted by other bus activities
- 1.25 DMIPS/MHz
- · Memory Protection Unit
 - Up to 8 protected memory regions
- 24-bit System Tick Timer for Real-Time Operating System (RTOS)
- 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- and 16-bit architectures



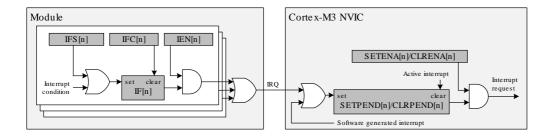
- · Unaligned data storage and access
 - Continuous storage of data requiring different byte lengths
 - Data access in a single core clock cycle
- · Integrated power modes
 - Sleep Now mode for immediate transfer to low power state
 - Sleep on Exit mode for entry into low power state after the servicing of an interrupt
 - Ability to extend power savings to other system components
- Optimized for low latency, nested interrupts

4.3 Functional Description

For a full functional description of the ARM Cortex-M3 (r2p0) implementation in the EFM32G family, the reader is referred to the *EFM32G Cortex-M3 Reference Manual*.

4.3.1 Interrupt Operation

Figure 4.1. Interrupt Operation



The EFM32G devices have up to 30 interrupt request lines (IRQ) which are connected to the Cortex-M3. Each of these lines (shown in Table 4.1 (p. 12)) are 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 IFS/IFC registers. 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/CLRPND bits in ISPRO/ICPR0) in the Cortex-M3 NVIC. The pending bit is then qualified with an enable bit (set/cleared with SETENA/CLRENA bits in ISERO/ICER0) before generating an interrupt request to the core. Figure 4.1 (p. 12) illustrates the interrupt system. For more information on how the interrupts are handled inside the Cortex-M3, the reader is referred to the *EFM32G Cortex-M3 Reference Manual*.

Table 4.1. Interrupt Request Lines (IRQ)

IRQ#	Source
0	DMA
1	GPIO_EVEN
2	TIMER0
3	USARTO_RX
4	USARTO_TX
5	ACMP0/ACMP1
6	ADC0
7	DAC0
8	I2C0
9	GPIO_ODD

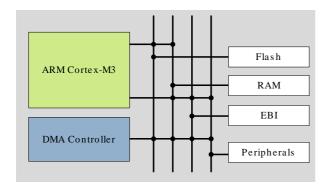


IRQ#	Source
10	TIMER1
11	TIMER2
12	USART1_RX
13	USART1_TX
14	USART2_RX
15	USART2_TX
16	UART0_RX
17	UART0_TX
18	LEUART0
19	LEUART1
20	LETIMER0
21	PCNT0
22	PCNT1
23	PCNT2
24	RTC
25	СМИ
26	VCMP
27	LCD
28	MSC
29	AES



5 Memory and Bus System





Quick Facts

What?

A low latency memory system, including low energy flash and RAM with data retention, makes extended use of low-power energymodes possible.

Why?

RAM retention reduces the need for storing data in flash and enables frequent use of the ultra low energy modes EM2 and EM3 with as little as 0.6 µA current consumption.

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.

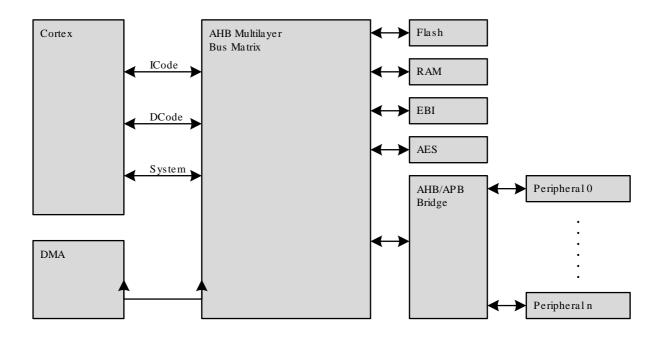
5.1 Introduction

The EFM32G contains an AMBA AHB Bus system allowing bus masters to access the memory mapped address space. A multilayer AHB bus matrix, using a Round-robin arbitration scheme, connects the master bus interfaces to the AHB slaves (Figure 5.1 (p. 15)). The bus matrix allows several AHB slaves 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 AHB bus masters are:

- Cortex-M3 ICode: Used for instruction fetches from Code memory (0x00000000 0x1FFFFFFF).
- Cortex-M3 DCode: Used for debug and data access to Code memory (0x00000000 0x1FFFFFFF).
- Cortex-M3 System: Used for instruction fetches, data and debug access to system space (0x20000000 0xDFFFFFFF).
- DMA: Can access EBI, SRAM, Flash and peripherals (0x00000000 0xDFFFFFFF).



Figure 5.1. EFM32G Bus System

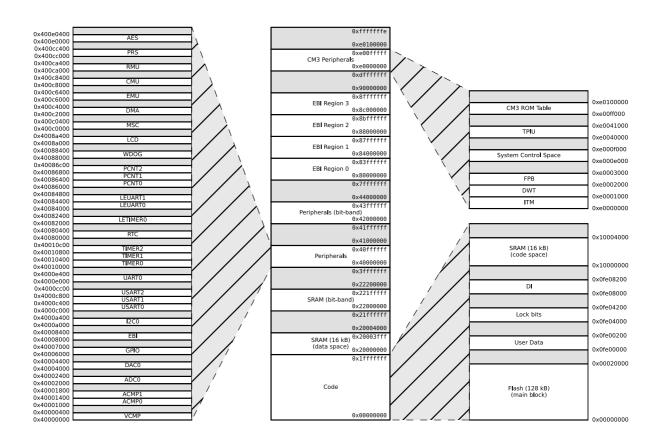


5.2 Functional Description

The memory segments are mapped together with the internal segments of the Cortex-M3 into the system memory map shown by Figure 5.2 (p. 16)



Figure 5.2. System Address Space



The embedded SRAM is located at address 0x20000000 in the memory map of the EFM32G. When running code located in SRAM starting at this address, the Cortex-M3 uses the System bus to fetch instructions. This results in reduced performance as the Cortex-M3 accesses stack, other data in SRAM and peripherals using the System bus. To be able to run code from SRAM efficiently, the SRAM is also mapped in the code space at address 0x10000000. When running code from this space, the Cortex-M3 fetches instructions through the I/D-Code bus interface, leaving the System bus for data access. The SRAM mapped into the code space can however only be accessed by the CPU, i.e. not the DMA.

5.2.1 Bit-banding

The SRAM bit-band alias and peripheral bit-band alias regions are located at 0x22000000 and 0x42000000 respectively. Read and write operations to these regions are converted into masked single-bit reads and atomic single-bit writes to the embedded SRAM and peripherals of the EFM32G.

The standard approach to modify a single register or SRAM bit in the aliased regions, requires software to read the value of the byte, half-word or word containing the bit, modify the bit, and then write the byte, half-word or word back to the register or SRAM address. Using bit-banding, this read-modify-write can be done in a single atomic operation. As read-writeback, bit-masking and bit-shift operations are not necessary in software, code size is reduced and execution speed improved.

The bit-band regions allows addressing each individual bit in the SRAM and peripheral areas of the memory map. To set or clear a bit in the embedded SRAM, write a 1 or a 0 to the following address:

Memory SRAM Area Set/Clear Bit $bit_address = 0x22000000 + (address - 0x20000000) \times 32 + bit \times 4,$ (5.1)



where *address* is the address of the 32-bit word containing the bit to modify, and *bit* is the index of the bit in the 32-bit word.

To modify a bit in the Peripheral area, use the following address:

Memory Peripheral Area Bit Modification

$$bit_address = 0x42000000 + (address - 0x40000000) \times 32 + bit \times 4,$$
 (5.2)

where address and bit are defined as above.

Note that the AHB-peripheral AES does not support bit-banding.

5.2.2 Peripherals

The peripherals are mapped into the peripheral memory segment, each with a fixed size address range according to Table 5.1 (p. 17), Table 5.2 (p. 17) and Table 5.3 (p. 18).

Table 5.1. Memory System Core Peripherals

Core peripherals	Core peripherals					
Address Range	Module Name					
0xE0041000 - 0xE0080FFF	ETM					
0x400E0000 - 0x400E03FF	AES					
0x400CA000 - 0x400CA3FF	RMU					
0x400C8000 - 0x400C83FF	СМИ					
0x400C6000 - 0x400C63FF	EMU					
0x400C4000 - 0x400C43FF	USB					
0x400C2000 - 0x400C3FFF	DMA					
0x400C1C00 - 0x400C1FFF	FPUEH					
0x400C0000 - 0x400C03FF	MSC					
0x40008000 - 0x400083FF	EBI					

Table 5.2. Memory System Low Energy Peripherals

Low Energy peripherals	
Address Range	Module Name
0x4008C000 - 0x4008C3FF	LESENSE
0x4008A000 - 0x4008A3FF	LCD
0x40088000 - 0x400883FF	WDOG
0x40086800 - 0x40086BFF	PCNT2
0x40086400 - 0x400867FF	PCNT1
0x40086000 - 0x400863FF	PCNT0
0x40084400 - 0x400847FF	LEUART1
0x40084000 - 0x400843FF	LEUART0
0x40082000 - 0x400823FF	LETIMER0
0x40081000 - 0x400813FF	BURTC
0x40080000 - 0x400803FF	RTC



Table 5.3. Memory System Peripherals

Peripherals	
Address Range	Module Name
0x400CC000 - 0x400CC3FF	PRS
0x40010C00 - 0x40010FFF	TIMER3
0x40010800 - 0x40010BFF	TIMER2
0x40010400 - 0x400107FF	TIMER1
0x40010000 - 0x400103FF	TIMER0
0x4000E400 - 0x4000E7FF	UART1
0x4000E000 - 0x4000E3FF	UART0
0x4000C800 - 0x4000CBFF	USART2
0x4000C400 - 0x4000C7FF	USART1
0x4000C000 - 0x4000C3FF	USART0
0x4000A400 - 0x4000A7FF	I2C1
0x4000A000 - 0x4000A3FF	12C0
0x40006000 - 0x40006FFF	GPIO
0x40004000 - 0x400043FF	DAC0
0x40002000 - 0x400023FF	ADC0
0x40001400 - 0x400017FF	ACMP1
0x40001000 - 0x400013FF	ACMP0
0x40000000 - 0x400003FF	VCMP

5.2.3 Bus Matrix

The Bus Matrix connects the memory segments to the bus masters:

- Code: CPU instruction or data fetches from the code space
- · System: CPU read and write to the SRAM, EBI and peripherals
- DMA: Access to EBI, SRAM, Flash and peripherals

5.2.3.1 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 slave are eliminated. Round-robin does not assign a fixed priority to each bus master. The arbiter does not insert any bus wait-states.

5.2.3.2 Access Performance

The Bus Matrix is a multi-layer energy optimized AMBA AHB compliant bus with an internal bandwidth equal to 4 times a single AHB-bus.

The Bus Matrix accepts new transfers initiated by each master in every clock cycle without inserting any wait-states. The slaves, however, may insert wait-states depending on their internal throughput and the clock frequency.

The Cortex-M3, the DMA Controller, and the peripherals run on clocks that can be prescaled separately. When accessing a peripheral which runs on a frequency equal to or faster than the HFCORECLK, the number of wait cycles per access, in addition to master arbitration, is given by:



Memory Wait Cycles with Clock Equal or Faster than HFCORECLK

$$N_{\text{cycles}} = 2 + N_{\text{slave cycles}},$$
 (5.3)

where N_{slave cycles} is the wait cycles introduced by the slave.

When accessing a peripheral running on a clock slower than the HFCORECLK, wait-cycles are introduced to allow the transfer to complete on the peripheral clock. The number of wait cycles per access, in addition to master arbitration, is given by:

Memory Wait Cycles with Clock Slower than CPU

$$N_{\text{cycles}} = (2 + N_{\text{slave cycles}}) \times f_{\text{HFCORECLK}} / f_{\text{HFPERCLK}},$$
 (5.4)

where N_{slave cycles} is the number of wait cycles introduced by the slave.

For general register access, $N_{\text{slave cycles}} = 1$.

More details on clocks and prescaling can be found in Chapter 11 (p. 94).

5.3 Access to Low Energy Peripherals (Asynchronous Registers)

5.3.1 Introduction

The Low Energy Peripherals are capable of running when the high frequency oscillator and core system is powered off, i.e. in energy mode EM2 and in some cases also EM3. This enables the peripherals to perform tasks while the system energy consumption is minimal.

The Low Energy Peripherals are:

- · Liquid Crystal Display driver LCD
- Low Energy Timer LETIMER
- Low Energy UART LEUART
- Pulse Counter PCNT
- Real Time Counter RTC
- · Watchdog WDOG

All Low Energy Peripherals are memory mapped, with automatic data synchronization. Because the Low Energy Peripherals are running on clocks asynchronous to the core clock, there are some constraints on how register accesses can be done, as described in the following sections.

5.3.1.1 Writing

Every Low Energy Peripheral has one or more registers with data that needs to be synchronized into the Low Energy clock domain to maintain data consistency and predictable operation. Due to synchronization, the write operation requires 3 positive edges of the clock of the Low Energy Peripheral being accessed. Such registers are marked "Asynchronous" in their description header.

See Figure 5.3 (p. 20) for a more detailed overview of the writing operation.

After writing data to a register which value is to be synchronized into the Low Energy clock domain, a corresponding busy flag in the <module_name>_SYNCBUSY register (e.g. RTC_SYNCBUSY) is set. This flag is set as long as synchronization is in progress and is cleared upon completion.

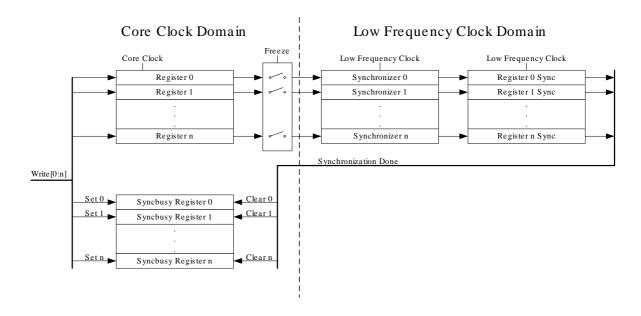


Note

Subsequent writes to the same register before the corresponding busy flag is cleared is not supported. Write before the busy flag is cleared may result in undefined behavior.

In general, the SYNCBUSY register only needs to be observed if there is a risk of multiple write access to a register (which must be prevented). It is not required to wait until the relevant flag in the SYNCBUSY register is cleared after writing a register. E.g EM2 can be entered immediately after writing a register.

Figure 5.3. Write operation to Low Energy Peripherals



5.3.1.2 Reading

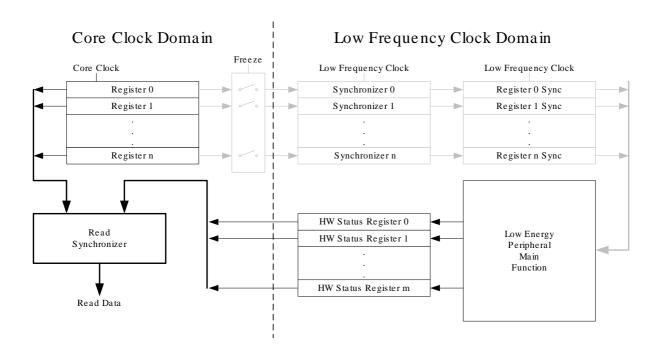
When reading from Low Energy Peripherals, the data is synchronized regardless of the originating clock domain. Registers updated/maintained by the Low Energy Peripheral are read directly from the Low Energy clock domain. Registers residing in the core clock domain, are read from the core clock domain. See Figure 5.4 (p. 21) for a more detailed overview of the read operation.

Note

Writing a register and then immediately reading back the value of the register may give the impression that the write operation is complete. This is not necessarily the case. Please refer to the SYNCBUSY register for correct status of the write operation to the Low Energy Peripheral.



Figure 5.4. Read operation from Low Energy Peripherals



5.3.2 FREEZE register

For Low Energy Peripherals there is a <module_name>_FREEZE register (e.g. RTC_FREEZE), containing a bit named REGFREEZE. If precise control of the synchronization process is required, this bit may be utilized. When REGFREEZE is set, the synchronization process is halted, allowing the software to write multiple Low Energy registers before starting the synchronization process, thus providing precise control of the module update process. The synchronization process is started by clearing the REGFREEZE bit.

5.4 Flash

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

- Up to 128 kB of memory
- Page size of 512 bytes (minimum erase unit)
- Minimum 20 000 erase cycles
- More than 10 years data retention at 85°C
- · Lock-bits for memory protection
- · Data retention in any state

5.5 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 used to transfer data between the SRAM, Flash and peripherals.

- Up to 16 kB memory
- · Bit-band access support
- 4 kB blocks may be individually powered down when not in use



• Data retention of the entire memory in EM0 to EM3

5.6 Device Information (DI) Page

The DI page contains calibration values, a unique identification number and other useful data. See the table below for a complete overview.

Table 5.4. Device Information Page Contents

DI Address	Register	Description
0x0FE08020	CMU_LFRCOCTRL	Register reset value.
0x0FE08028	CMU_HFRCOCTRL	Register reset value.
0x0FE08030	CMU_AUXHFRCOCTRL	Register reset value.
0x0FE08040	ADC0_CAL	Register reset value.
0x0FE08048	ADC0_BIASPROG	Register reset value.
0x0FE08050	DAC0_CAL	Register reset value.
0x0FE08058	DAC0_BIASPROG	Register reset value.
0x0FE08060	ACMP0_CTRL	Register reset value.
0x0FE08068	ACMP1_CTRL	Register reset value.
0x0FE08078	CMU_LCDCTRL	Register reset value.
0x0FE081B0	DI_CRC	[15:0]: DI data CRC-16.
0x0FE081B2	CAL_TEMP_0	[7:0] Calibration temperature (°C).
0x0FE081B4	ADC0_CAL_1V25	[14:8]: Gain for 1V25 reference, [6:0]: Offset for 1V25 reference.
0x0FE081B6	ADC0_CAL_2V5	[14:8]: Gain for 2V5 reference, [6:0]: Offset for 2V5 reference.
0x0FE081B8	ADC0_CAL_VDD	[14:8]: Gain for VDD reference, [6:0]: Offset for VDD reference.
0x0FE081BA	ADC0_CAL_5VDIFF	[14:8]: Gain for 5VDIFF reference, [6:0]: Offset for 5VDIFF reference.
0x0FE081BC	ADC0_CAL_2XVDD	[14:8]: Reserved (gain for this reference cannot be calibrated), [6:0]: Offset for 2XVDD reference.
0x0FE081BE	ADC0_TEMP_0_READ_1V25	[15:4] Temperature reading at 1V25 reference, [3:0] Reserved.
0x0FE081C8	DAC0_CAL_1V25	[22:16]: Gain for 1V25 reference, [13:8]: Channel 1 offset for 1V25 reference, [5:0]: Channel 0 offset for 1V25 reference.
0x0FE081CC	DAC0_CAL_2V5	[22:16]: Gain for 2V5 reference, [13:8]: Channel 1 offset for 2V5 reference, [5:0]: Channel 0 offset for 2V5 reference.
0x0FE081D0	DAC0_CAL_VDD	[22:16]: Reserved (gain for this reference cannot be calibrated), [13:8]: Channel 1 offset for VDD reference, [5:0]: Channel 0 offset for VDD reference.
0x0FE081D4	RESERVED	[31:0] Reserved
0x0FE081D8	RESERVED	[31:0] Reserved
0x0FE081DC	HFRCO_CALIB_BAND_1	[7:0]: Tuning for the 1.2 MHZ HFRCO band.
0x0FE081DD	HFRCO_CALIB_BAND_7	[7:0]: Tuning for the 6.6 MHZ HFRCO band.
0x0FE081DE	HFRCO_CALIB_BAND_11	[7:0]: Tuning for the 11 MHZ HFRCO band.
	· · · · · · · · · · · · · · · · · · ·	

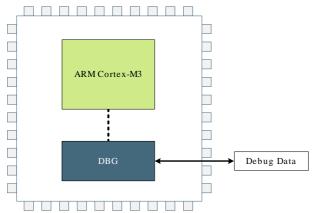


DI Address	Register	Description
0x0FE081DF	HFRCO_CALIB_BAND_14	[7:0]: Tuning for the 14 MHZ HFRCO band.
0x0FE081E0	HFRCO_CALIB_BAND_21	[7:0]: Tuning for the 21 MHZ HFRCO band.
0x0FE081E1	HFRCO_CALIB_BAND_28	[7:0]: Tuning for the 28 MHZ HFRCO band.
0x0FE081E7	MEM_INFO_PAGE_SIZE	[7:0] Flash page size in bytes coded as 2 ^ ((MEM_INFO_PAGE_SIZE + 10) & 0xFF). le. the value 0xFF = 512 bytes.
0x0FE081F0	UNIQUE_0	[31:0] Unique number.
0x0FE081F4	UNIQUE_1	[63:32] Unique number.
0x0FE081F8	MEM_INFO_FLASH	[15:0]: Flash size, kbyte count as unsigned integer (eg. 128).
0x0FE081FA	MEM_INFO_RAM	[15:0]: Ram size, kbyte count as unsigned integer (eg. 16).
0x0FE081FC	PART_NUMBER	[15:0]: EFM32 part number as unsigned integer (eg. 230).
0x0FE081FE	PART_FAMILY	[7:0]: EFM32 part family number (Gecko = 71, Giant Gecko = 72, Tiny Gecko = 73, Leopard Gecko=74, Wonder Gecko=75).
0x0FE081FF	PROD_REV	[7:0]: EFM32 Production ID.



6 DBG - Debug Interface





Quick Facts

What?

The DBG (Debug Interface) is used to program and debug EFM32G devices.

Why?

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

How?

The Cortex-M3 supports advanced debugging features. EFM32G devices only use 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 EFM32G devices include hardware debug support through a 2-pin serial-wire debug (SWD) interface. 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-M3 Technical Reference Manual
- ARM CoreSight Components Technical Reference Manual
- ARM Debug Interface v5 Architecture Specification

6.2 Features

- Flash Patch and Breakpoint (FPB) unit
 - Implement breakpoints and code patches
- Data Watch point and Trace (DWT) unit
 - · Implement watch points, trigger resources and system profiling
- Instrumentation Trace Macrocell (ITM)
 - Application-driven trace source that supports printf style debugging

6.3 Functional Description

There are three debug pins and four trace pins available on the device. Operation of these pins are described in the following section.

6.3.1 Debug Pins

The following pins are the debug connections for the device:



- Serial Wire Clock input (SWCLK): This pin is enabled after reset and has a built-in pull down.
- Serial Wire Data Input/Output (SWDIO): This pin is enabled after reset and has a built-in pull-up.
- Serial Wire Viewer (SWV): This pin is disabled after reset.

The debug pins can be enabled and disabled through GPIO_ROUTE, see Section 28.3.2.1 (p. 405). Please remeberer that upon disabling, debug contact with the device is lost. Also note that, because the debug pins have pull-down and pull-up enabled by default, leaving them enabled might increase the current consumption with up to $200~\mu A$ if left connected to supply or ground.

6.3.2 Debug and EM2/EM3

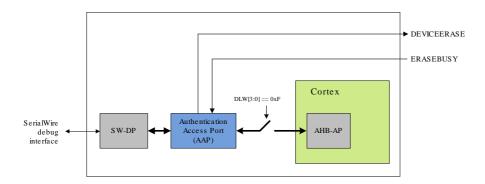
Leaving the debugger connected when issuing a WFI or WFE to enter EM2 or EM3 will make the system enter a special EM2. This mode differs from regular EM2 and EM3 in that the high frequency clocks are still enabled, and certain core functionality is still powered in order to maintain debug-functionality.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

6.4 Debug Lock and Device Erase

The debug access to the Cortex-M3 is locked by clearing the Debug Lock Word (DLW) and resetting the device, see Section 7.3.2 (p. 31).

When debug access is locked, the debug interface remains accessible but the connection to the Cortex-M3 core and the whole bus-system is blocked as shown in Figure 6.2 (p. 26). This mechanism is controlled by the Authentication Access Port (AAP) as illustrated by Figure 6.1 (p. 25). The AAP is only accessible from a debugger and not from the core.

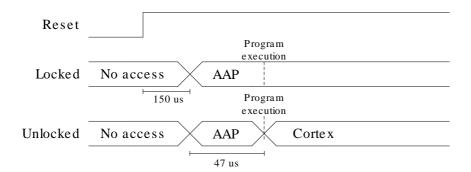
Figure 6.1. AAP - Authentication Access Port



The debugger can access the AAP-registers, and only these registers just after reset, for the time of the AAP-window outlined in Figure 6.2 (p. 26). If the device is locked, access to the core and bus-system is blocked even after code execution starts, and the debugger can only access the AAP-registers. If the device is not locked, the AAP is no longer accessible after code execution starts, and the debugger can access the core and bus-system normally.



Figure 6.2. Device Unlock



If the device is locked, it can be unlocked by writing a valid key to the AAP_CMDKEY register and then setting the DEVICEERASE bit of the AAP_CMD register via the debug interface. The commands are not executed before AAP_CMDKEY is invalidated, so this register should be cleared to to start the erase operation. This operation erases the main block of flash, all lock bits are reset and debug access through the AHB-AP is enabled. The operation takes 40 ms to complete. Note that the SRAM contents will also be deleted during a device erase, while the UD-page is not erased.

Even if the device is not locked, the can device can be erased through the AAP, using the above procedure during the AAP window. This can be useful if the device has been programmed with code that, e.g., disables the debug interface pins on start-up, or does something else that prevents communication with a debugger.

If the device is locked, the debugger may read the status from the AAP_STATUS register. When the ERASEBUSY bit is set low after DEVICEERASE of the AAP_CMD register is set, the debugger may set the SYSRESETREQ bit in the AAP_CMD register. After reset, the debugger may resume a normal debug session through the AHB-AP. If the device is not locked, the device erase starts when the AAP window closes, so it is not possible to poll the status.



6.5 Register Map

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

Offset	Name	Туре	Description
0x000	AAP_CMD	W1	Command Register
0x004	AAP_CMDKEY	W1	Command Key Register
0x008	AAP_STATUS	R	Status Register
0x0FC	AAP_IDR	R	AAP Identification Register

6.6 Register Description

6.6.1 AAP_CMD - Command Register

Offset															Bi	it Po	siti	on														
0x000	31	30	29	28	27	56	22	24	23	22	21	20	19	9	17	16	15	4	13	12	=	10	6	8	7	9	2	4	က	7	-	0
Reset																														,	0	0
Access																															W1	W1
Name																															SYSRESETREQ	DEVICEERASE

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	SYSRESETREQ	0	W1	System Reset Request
	A system reset reques	t is generated wher	n set to 1. This reg	gister is write enabled from the AAP_CMDKEY register.
0	DEVICEERASE	0	W1	Erase the Flash Main Block, SRAM and Lock Bits
				erased, the SRAM is cleared and then the Lock Bit (LB) page is erased.

When set, all data and program code in the main block is erased, the SRAM is cleared and then the Lock Bit (LB) page is erased. This also includes the Debug Lock Word (DLW), causing debug access to be enabled after the next reset. The information block User Data page (UD) is left unchanged, but the User data page Lock Word (ULW) is erased. This register is write enabled from the AAP_CMDKEY register.

6.6.2 AAP_CMDKEY - Command Key Register

Offset															Bit	t Po	siti	on														
0x004	31	30	29	28	27	26	25	24	23	22	21	70	19	18	17	16	15	4	13	12	11	10	6	∞	7	9	2	4	က	2	-	0
Reset																0000000	000000000000000000000000000000000000000															
Access																7	- >															
Name																WEITEKEY																

Bit	Name	Reset	Access	Description
31:0	WRITEKEY	0x00000000	W1	CMD Key Register



Bit	Name	Reset A	ccess	Description
		ast be written to this register to we ecute the command.	rite enable	the AAP_CMD register. After AAP_CMD is written, this register should
	Value	Mode	Desc	ription
	0xCFACC118	WRITEEN	Enab	le write to AAP_CMD

6.6.3 AAP_STATUS - Status Register

Offset															Bi	t Pc	siti	on														
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	∞	7	9	2	4	ю	2	-	0
Reset				,						•		•			,	,																0
Access																																~
Name																																ERASEBUSY

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure co	ompatibility with fo	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	ERASEBUSY	0	R	Device Erase Command Status
	This bit is set when a	device erase is exec	cuting.	

6.6.4 AAP_IDR - AAP Identification Register

Offset															Bi	t Po	siti	on														
0x0FC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	80	7	9	2	4	က	2	-	0
Reset																	OXIBEROOO															
Access																٥	۲															
Name																																

Bit	Name	Reset	Access	Description
31:0	ID	0x16E60001	R	AAP Identification Register
	Access port identification re	egister in compliand	e with the ARI	M ADI v5 specification (JEDEC Manufacturer ID) .



7 MSC - Memory System Controller



01100101011100110110100101100111

01101110001000010100010101101101

Quick Facts

What?

The user can perform Flash memory read, read configuration and write operations through the Memory System Controller (MSC).

Why?

The MSC allows the application code, user data and flash lock bits to be stored in non-volatile Flash memory. Certain memory system functions, such as program memory wait-states and bus faults are also 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.

Application code may dynamically scale between high energy optimization and high code execution performance through advanced read modes.

7.1 Introduction

The Memory System Controller (MSC) is the program memory unit of the EFM32G microcontroller. The flash memory is readable and writable from both the Cortex-M3 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. Additionally, the information block is available for special user data and flash lock bits. There is also a read-only page in the information block containing system and device calibration data. Read and write operations are supported in the energy modes EM0 and EM1.

7.2 Features

- · AHB read interface
 - Scalable access performance to optimize the Cortex-M3 code interface
 - Zero wait-state access up to 16 MHz and one wait-state for 16 MHz and above
 - Advanced energy optimization functionality



- Conditional branch target prefetch suppression
- · Cortex-M3 disfolding of if-then (IT) blocks
- DMA read support in EM0 and EM1
- · Command and status interface
 - · Flash write and erase
 - Accessible from Cortex-M3 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
 - Need for special time scaling registers eliminated
 - · Configurable interrupt erase abort
 - · Improved interrupt predictability
 - · Memory and bus fault control
- · Security features
 - · Lockable debug access
 - · Page lock bits
 - User data lock bits
- End-of-write and end-of-erase interrupts

7.3 Functional Description

The size of the main block is device dependent. The largest size available is 128 kB (256 pages). The information block has 512 bytes available for user data. The information block also contains chip configuration data located in a reserved area. The main block is mapped to address 0x000000000 and the information block is mapped to address 0x0FE00000. Table 7.1 (p. 30) outlines how the Flash is mapped in the memory space. All Flash memory is organized into 512 byte pages.

Table 7.1. MSC Flash Memory Mapping

Block	Page	Base address	Write/Erase by	Software readable	Purpose/Name	Size
Main ¹	0	0x00000000	Software, debug	Yes	User code and data	16 KB - 128 kB
			Software, debug	Yes		
	255	0x0001FE00	Software, debug	Yes		
Reserved	-	0x00020000	-	-	Reserved for flash expansion	~24 MB
Information	0	0x0FE00000	Software, debug	Yes	User Data (UD)	512 B
	-	0x0FE00200	-	-	Reserved	
	1	0x0FE04000	Debug only	Yes	Lock Bits (LB)	512 B
	-	0x0FE04200	-	-	Reserved	
	2	0x0FE08000	-	Yes	Device Information (DI)	512 B
	-	0x0FE08200	-	-	Reserved	
Reserved	-	0x0FE10000	-	-	Reserved for flash expansion	Rest of code space

¹Block/page erased by a device erase



7.3.1 User Data (UD) Page Description

This is the user data page in the information block. The page can be erased and written by software. The page is erased by the ERASEPAGE command of the MSC_WRITECMD register. Note that the page is not erased by a device erase operation. The device erase operation is described in Section 6.4 (p. 25).

7.3.2 Lock Bits (LB) Page Description

This page contains the following information:

- Debug Lock Word (DLW)
- User data page Lock Word (ULW)
- Main block Page Lock Words (PLWs)

The words in this page are organized as shown in Table 7.2 (p. 31):

Table 7.2. Lock Bits Page Structure

127	DLW
126	ULW
N	PLW[N]
1	PLW[1]
0	PLW[0]

Word 127 is the debug lock word (DLW). Bit 0 of this word is the debug lock bit. If this bit is 1, then debug access is enabled. Debug access to the core is disabled from power-on reset until the DLW is evaluated immediately before the Cortex-M3 starts execution of the user application code. If the bit is 0, then debug access to the core remains blocked.

Word 126 is the user page lock word (ULW). Bit 0 of this word is the page lock bit. The lock bits can be reset by a device erase operation initiated from the Authentication Access Port (AAP) registers. The AAP is described in more detail in Section 6.4 (p. 25) . Note that the AAP is only accessible from the debug interface, and cannot be accessed from the Cortex-M3 core.

There are 32 page lock bits per page lock word (PLW). Bit 0 refers to the first page and bit 31 refers to the last page within a PLW. Thus, PLW[0] contains lock bits for page 0-31 in the main block. Similarly, PLW[1] contains lock bits for page 32-63 and so on. A page is locked when the bit is 0. A locked page cannot be erased or written.

The lock bits can be reset by a device erase operation initiated from the Authentication Access Port (AAP) registers. The AAP is described in more detail in Section 6.4 (p. 25). Note that the AAP is only accessible from the debug interface, and cannot be accessed from the Cortex-M3 core.

7.3.3 Device Information (DI) Page

This read-only page holds the calibration data for the oscillator and other analog peripherals from the production test as well as a unique device ID. The page is further described in Section 5.6 (p. 22).

7.3.4 Post-reset Behavior

Calibration values are automatically written to registers by the MSC before application code startup. The values are also available to read from the DI page for later reference by software. Other information such as the device ID and production date is also stored in the DI page and is readable from software.



7.3.4.1 One Wait-state Access

After reset, the HFCORECLK is normally 14 MHz from the HFRCO and the MODE field of the MSC_READCTRL register is set to WS1 (one wait-state). The reset value must be WS1 as an uncalibrated HFRCO may produce a frequency higher than 16 MHz. Software must not select a zero wait-state mode unless the clock is guaranteed to be 16 MHz or below, otherwise the resulting behavior is undefined. If a HFCORECLK frequency above 16 MHz is to be set by software, the MODE field of the MSC_READCTRL register must be set to WS1 or WS1SCBTP before the core clock is switched to the higher frequency clock source.

When changing to a lower frequency, the MODE field of the MSC_READCTRL register can be set to WS0 or WS0SCBTP, but only after the frequency transition is completed. If the HFRCO is used, wait until the oscillator is stable on the new frequency. Otherwise, the behavior is unpredictable.

7.3.4.2 Zero Wait-state Access

At 16 MHz and below, read operations from flash may be performed without any wait-states. Zero wait-state access greatly improves code execution performance at frequencies from 16 MHz and below. By default, the Cortex-M3 uses speculative prefetching and If-Then block folding to maximize code execution performance at the cost of additional flash accesses and energy consumption.

7.3.4.3 Suppressed Conditional Branch Target Prefetch (SCBTP)

MSC offers a special instruction fetch mode which optimizes energy consumption by cancelling Cortex-M3 conditional branch target prefetches. Normally, the Cortex-M3 core prefetches both the next sequential instruction and the instruction at the branch target address when a conditional branch instruction reaches the pipeline decode stage. This prefetch scheme improves performance while one extra instruction is fetched from memory at each conditional branch, regardless of whether the branch is taken or not. To optimize for low energy, the MSC can be configured to cancel these speculative branch target prefetches. With this configuration, energy consumption is more optimal, as the branch target instruction fetch is delayed until the branch condition is evaluated.

The performance penalty with this mode enabled is source code dependent, but is normally less than 1% for core frequencies from 16 MHz and below. To enable the mode at frequencies from 16 MHz and below write WS0SCBTP to the MODE field of the MSC_READCTRL register. For frequencies above 16 MHz, use the WS1SCBTP mode. An increased performance penalty per clock cycle must be expected compared to WS0SCBTP mode. The performance penalty in WS1SCBTP mode depends greatly on the density and organization of conditional branch instructions in the code.

7.3.4.4 Cortex-M3 If-Then Block Folding

The Cortex-M3 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-M3 Technical Reference Manual for details. Normally, it is expected that this feature is most efficient at core frequencies above 16 MHz. Folding is enabled by default.

7.3.5 Erase and Write Operations

Both page erase and write operations require that the address is written into the MSC_ADDRB register. For erase operations, the address may be any within the page to be erased. Load the address by writing 1 to the LADDRIM bit in the MSC_WRITECMD register. The LADDRIM bit only has to be written once when loading the first address. After each word is written the internal address register ADDR will be incremented automatically by 4. The INVADDR bit of the MSC_STATUS register is set if the loaded address is outside the flash and the LOCKED bit of the MSC_STATUS register is set if the page



addressed is locked. Any attempts to command erase of or write to the page are ignored if INVADDR or the LOCKED bits of the MSC_STATUS register are set.

When a word is written to the MSC_WDATA register, the WDATAREADY bit of the MSC_STATUS register is cleared. When this status bit is set, software or DMA may write the next word.

A single word write is commanded by setting the WRITEONCE bit of the MSC_WRITECMD register. The operation is complete when the BUSY bit of the MSC_STATUS register is cleared and control of the flash is handed back to the AHB interface, allowing application code to resume execution.

For a DMA write the software must write the first word to the MSC_WDATA register and then set the WRITETRIG bit of the MSC_WRITECMD register. DMA triggers when the WDATAREADY bit of the MSC_STATUS register is set.

It is possible to write words twice between each erase by keeping at 1 the bits that are not to be changed. Let us take as an example writing two 16 bit values, 0xAAAA and 0x5555. To safely write them in the same flash word this method can be used:

- Write 0xFFFFAAAA (word in flash becomes 0xFFFFAAAA)
- Write 0x5555FFFF (word in flash becomes 0x5555AAAA)

Note that there is a maximum of two writes to the same word between each erase due to a physical limitation of the flash.

Note

The WRITEONCE, WRITETRIG and ERASEPAGE bits in the MSC_WRITECMD 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

The MSC_WDATA and MSC_ADDRB registers are not retained when entering EM2 or lower energy modes.



7.4 Register Map

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

Offset	Name	Туре	Description
0x000	MSC_CTRL	RW	Memory System Control Register
0x004	MSC_READCTRL	RW	Read Control Register
0x008	MSC_WRITECTRL	RW	Write Control Register
0x00C	MSC_WRITECMD	W1	Write Command Register
0x010	MSC_ADDRB	RW	Page Erase/Write Address Buffer
0x018	MSC_WDATA	RW	Write Data Register
0x01C	MSC_STATUS	R	Status Register
0x02C	MSC_IF	R	Interrupt Flag Register
0x030	MSC_IFS	W1	Interrupt Flag Set Register
0x034	MSC_IFC	W1	Interrupt Flag Clear Register
0x038	MSC_IEN	RW	Interrupt Enable Register
0x03C	MSC_LOCK	RW	Configuration Lock Register

7.5 Register Description

7.5.1 MSC_CTRL - Memory System Control Register

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	œ	7	9	2	4	က	2	-	0
Reset										•		•											,									-
Access																																W.
Name																																BUSFAULT

Bit	Name	Reset	Access	Description								
31:1	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)								
0	BUSFAULT	1	RW	Bus Fault Response Enable								
	When this bit is	set, the memory system ge	enerates bus erro	or response.								
	Value	Mode	Des	cription								
	0	GENERATE	A bu	us fault is generated on access to unmapped code and system space.								
	1	IGNORE	Acc	Accesses to unmapped address space is ignored.								

7.5.2 MSC_READCTRL - Read Control Register

Offset															Bi	t Pc	siti	on														
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	=	10	6	8	7	9	2	4	က	2	-	0
Reset																					-										0×1	
Access																															R M	
Name																															MODE	



Bit	Name	Reset	Access	Description
31:3	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
2:0	MODE	0x1	RW	Read Mode

If software wants to set a core clock frequency above 16 MHz, this register must be set to WS1 or WS1SCBTP before the core clock is switched to the higher frequency. When changing to a lower frequency, this register can be set to WS0 or WS0SCBTP after the frequency transition has been completed. After reset, the core clock is 14 MHz from the HFRCO but the MODE field of MSC_READCTRL register is set to WS1. This is because the HFRCO may produce a frequency above 16 MHz before it is calibrated. If the HFRCO is used as clock source, wait until the oscillator is stable on the new frequency to avoid unpredictable behavior.

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. This mode is required for a core frequency above 16 MHz.
2	WSOSCBTP	Zero wait-states inserted with the Suppressed Conditional Branch Target Prefetch (SCBTP) function enabled. SCBTP saves energy by delaying the Cortex' conditional branch target prefetches until the conditional branch instruction is in the execute stage. When the instruction reaches this stage, the evaluation of the branch condition is completed and the core does not perform a speculative prefetch of both the branch target address and the next sequential address. With the SCBTP function enabled, one instruction fetch is saved for each branch not taken, with a negligible performance penalty.
3	WS1SCBTP	One wait-state access with SCBTP enabled.

7.5.3 MSC_WRITECTRL - Write Control Register

Offset	Bit Position		
0x008	33 34 37 38 39 30 30 31 32 33 34 35 36 37 38 40 <td>-</td> <td>0</td>	-	0
Reset		0	0
Access		RW	RW
Name		IRQERASEABORT	WREN

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure co	empatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	IRQERASEABORT	0	RW	Abort Page Erase on Interrupt
	When this bit is set to 1,	any Cortex interre	upt aborts any cur	rent page erase operation.
0	WREN	0	RW	Enable Write/Erase Controller
	When this bit is set, the I	MSC write and er	ase functionality is	s enabled.

7.5.4 MSC_WRITECMD - Write Command Register

Offset								·							Bi	t Pc	siti	on														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	8	7	9	2	4	ю	2	-	0
Reset															,								•			•		0	0	0	0	0
Access																												W	Ž	Ž	W1	W N
Name																												WRITETRIG	WRITEONCE	WRITEEND	ERASEPAGE	LADDRIM



Bit	Name	Bosot	A 00000	Description
DIL	name	Reset	Access	Description
31:5	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
4	WRITETRIG	0	W1	Word Write Sequence Trigger
	Functions like MSC_0 within the 30 µs timed	_	, but will set MSC	_STATUS_WORDTIMEOUT if no new data is written to MSC_WDATA
3	WRITEONCE	0	W1	Word Write-Once Trigger
				dd 4 to ADDR and write the next word if available within a 30 μs timeout. R is set to the base of the page.
2	WRITEEND	0	W1	End Write Mode
	Write 1 to end write m	node when using the	WRITETRIG com	nmand.
1	ERASEPAGE	0	W1	Erase Page
	Erase any user define in order to use this co		the MSC_ADDRE	3 register. The WREN bit in the MSC_WRITECTRL register must be set
0	LADDRIM	0	W1	Load MSC_ADDRB into ADDR
		•		C_ADDRB register. The internal address register ADDR is incremented ncremented past the page boundary, ADDR is set to the base of the page.

7.5.5 MSC_ADDRB - Page Erase/Write Address Buffer

Offset		Bit Position																												
0x010	33	8 8	28	27	26	25	24	23	22	21	20	6	9	17	16	4	ć	12	7	10	6	8	7	9	2	4	ю	7	-	0
Reset	00000000000000000000000000000000000000																													
Access	≥ 2																													
Name	ADDRB																													
Bit	Nan	e					Re	set			A	ссе	ss		Desc	ript	ior	1												
31:0	ADDI	RB					0x0	00000	0000		R'	W			Page Erase or Write Address Buffer															

This register holds the page address for the erase or write operation. This register is loaded into the internal MSC_ADDR register when the LADDRIM field in MSC_WRITECMD is set. The MSC_ADDR register is not readable. This register is not retained when

7.5.6 MSC_WDATA - Write Data Register

entering EM2 or lower energy modes.

Offset	Bit Position																															
0x018	31	8	53	28	27	26	25	24	23	22	21	20	19	8	17	16	15	4	13	12	7	10	တ	ω	7	9	2	4	က	7	-	0
Reset	00000000000000000000000000000000000000																															
Access																<u> </u>	2															
Name																V TA CIW																



Bit	Name	Reset	Access	Description
31:0	WDATA	0x00000000	RW	Write Data
	The data to be written to t is set. This register is not		_	register must be written when the WDATAREADY bit of MSC_STATUS wer energy modes.

7.5.7 MSC_STATUS - Status Register

Offset															Bi	t Po	siti	on														
0x01C	31	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	80	7	9	2	4	ю	2	-	0
Reset		,		•					•			•					•						•				0	0	-	0	0	0
Access																											2	~	22	~	~	<u>~</u>
Name																											ERASEABORTED	WORDTIMEOUT	WDATAREADY	INVADDR	LOCKED	BUSY

Bit	Name	Reset	Access	Description								
31:6	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)								
5	ERASEABORTED	0	R	The Current Flash Erase Operation Aborted								
	When set, the current er	ase operation wa	s aborted by interi	rupt.								
4	WORDTIMEOUT	0	R	Flash Write Word Timeout								
	•	AHB interface. 7		n the timeout. The flash write operation timed out and access to the when the ERASEPAGE, WRITETRIG or WRITEONCE commands in								
3	WDATAREADY											
		_		y MSC Flash Write Controller and the register may be updated with the en writing to MSC_WDATA.								
2	INVADDR	0	R	Invalid Write Address or Erase Page								
	Set when software atten	npts to load an inv	valid (unmapped) a	address into ADDR.								
1	LOCKED	0	R	Access Locked								
	When set, the last erase	or write is aborte	ed due to erase/wr	ite access constraints.								
0	BUSY	0	R	Erase/Write Busy								
	When set, an erase or w	rite operation is i	n progress and ne	w commands are ignored.								
	<u> </u>											

7.5.8 MSC_IF - Interrupt Flag Register

Offset															Bi	t Po	siti	on														
0x02C	31	30	53	28	27	56	22	24	23	22	21	20	19	9	17	16	15	4	13	12	=	10	တ	∞	7	9	2	4	က	2	-	0
Reset																															0	0
Access																															œ	~
Name																															WRITE	ERASE

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure comp	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)



Bit	Name	Reset	Access	Description
1	WRITE	0	R	Write Done Interrupt Read Flag
	Set when a write is done.			
0	ERASE	0	R	Erase Done Interrupt Read Flag
	Set when erase is done.			

7.5.9 MSC_IFS - Interrupt Flag Set Register

Offset															Bi	t Pc	siti	on														
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	œ	7	9	2	4	ю	2	_	0
Reset																•															0	0
Access																															W	W1
Name																															WRITE	ERASE

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	WRITE	0	W1	Write Done Interrupt Set
	Set the write done	bit and generate interru	ıpt.	
0	ERASE	0	W1	Erase Done Interrupt Set
	Set the erase done	e bit and generate interr	upt.	

7.5.10 MSC_IFC - Interrupt Flag Clear Register

Offset															Bi	t Pc	siti	on														
0x034	31	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	8	7	-	0
Reset					•				•							•	•				-				•						0	0
Access																															N N	M
Name																															WRITE	ERASE

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure cor	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	WRITE	0	W1	Write Done Interrupt Clear
	Clear the write done bit.			
0	ERASE	0	W1	Erase Done Interrupt Clear
	Clear the erase done bit.			



7.5.11 MSC_IEN - Interrupt Enable Register

Offset				,	,		,					,			Bit	t Po	siti	on							,							
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	တ	œ	7	9	2	4	က	2	-	0
Reset																															0	0
Access																															RW	N N
Name																															WRITE	ERASE

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure com	patibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	WRITE	0	RW	Write Done Interrupt Enable
	Enable the write done int	errupt.		
0	ERASE	0	RW	Erase Done Interrupt Enable
	Enable the erase done in	iterrupt.		

7.5.12 MSC_LOCK - Configuration Lock Register

Offset	Bit Po					osition																										
0x03C	31	98	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	41	13	12	1	10	6	8	7	9	2	4	က	2	-	0
Reset																								000	000000							
Access																								Š	≩ Ƴ							
Name) 	LOCKKEY							

Bit	Name	Reset Acces	
31:16	Reserved	To ensure compatibility wi	th future devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	LOCKKEY	0x0000 RW	Configuration Lock

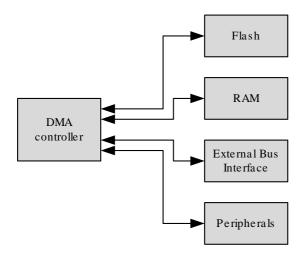
Write any other value than the unlock code to lock access to MSC_CTRL, MSC_READCTRL and MSC_WRITECTRL. Write the unlock code to enable access. When reading the register, bit 0 is set when the lock is enabled.

Mode	Value	Description
Read Operation		
UNLOCKED	0	MSC registers are unlocked.
LOCKED	1	MSC registers are locked.
Write Operation		
LOCK	0	Lock MSC registers.
UNLOCK	0x1B71	Unlock MSC registers.



8 DMA - DMA Controller





Quick Facts

What?

The DMA controller can move data without CPU intervention, effectively reducing the energy consumption for a data transfer.

Why?

The DMA can perform data transfers more energy efficiently than the CPU and allows autonomous operation in low energy modes. The LEUART can for instance provide full UART communication in EM2, consuming only a few µA by using the DMA to move data between the LEUART and RAM.

How?

The DMA controller has multiple highly configurable, prioritized DMA channels. Advanced transfer modes such as ping-pong and scatter-gather make it possible to tailor the controller to the specific needs of an application.

8.1 Introduction

The Direct Memory Access (DMA) controller performs memory 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 for example when moving data from the USART to RAM or from the External Bus Interface (EBI) to the DAC. The DMA controller uses the PL230 µDMA controller licensed from ARM¹. Each of the PL230s channels on the EFM32 can be connected to any of the EFM32 peripherals.

8.2 Features

- The DMA controller is accessible as a memory mapped peripheral
- Possible data transfers include
 - RAM/EBI/Flash to peripheral
 - · RAM/EBI to Flash
 - Peripheral to RAM/EBI
 - RAM/EBI/Flash to RAM/EBI
- The DMA controller has 8 independent channels
- Each channel has one (primary) or two (primary and alternate) descriptors
- The configuration for each channel includes
 - Transfer mode
 - Priority
 - Word-count
 - Word-size (8, 16, 32 bit)
- The transfer modes include
 - Basic (using the primary or alternate DMA descriptor)

¹ARM PL230 homepage [http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0417a/index.html]

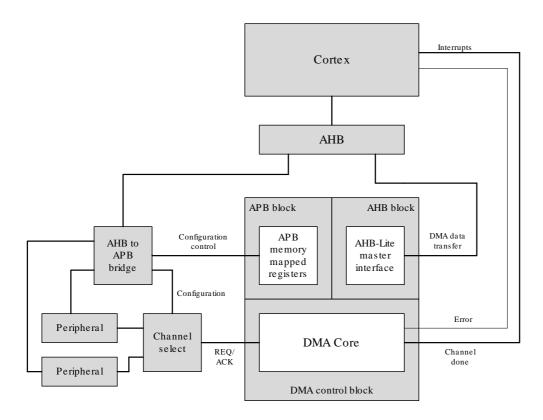


- Ping-pong (switching between the primary or alternate DMA descriptors, for continuous data flow to/from peripherals)
- Scatter-gather (using the primary descriptor to configure the alternate descriptor)
- · Each channel has a programmable transfer length
- Channels 0 and 1 support looped transfers
- Channel 0 supports 2D copy
- A DMA channel can be triggered by any of several sources:
 - Communication modules (USART, UART, LEUART)
 - Timers (TIMER)
 - Analog modules (DAC, ACMP, ADC)
 - External Bus Interface (EBI)
 - Software
- Programmable mapping between channel number and peripherals any DMA channel can be triggered by any of the available sources
- · Interrupts upon transfer completion
- Data transfer to/from LEUART in EM2 is supported by the DMA, providing extremely low energy consumption while performing UART communications

8.3 Block Diagram

An overview of the DMA and the modules it interacts with is shown in Figure 8.1 (p. 41).

Figure 8.1. DMA Block Diagram



The DMA Controller consists of four main parts:

- An APB block allowing software to configure the DMA controller
- An AHB block allowing the DMA to read and write the DMA descriptors and the source and destination data for the DMA transfers
- A DMA control block controlling the operation of the DMA, including request/acknowledge signals for the connected peripherals



A channel select block routing the right peripheral request to each DMA channel

8.4 Functional Description

The 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 DMA work autonomously with the LEUART for data transfer in EM2 without having to wake up the processor core from sleep.

The DMA Controller contains 8 independent channels. Each of these channels can be connected to any of the available peripheral trigger sources by writing to the configuration registers, see Section 8.4.1 (p. 42). In addition, each channel can be triggered by software (for large memory transfers or for debugging purposes).

What the DMA Controller should do (when one of its channels is triggered) is configured through channel descriptors residing in system memory. Before enabling a channel, the software must therefore take care to write this configuration to memory. When a channel is triggered, the DMA Controller will first read the channel descriptor from system memory, and then it will proceed to perform the memory transfers as specified by the descriptor. The descriptor contains the memory address to read from, the memory address to write to, the number of bytes to be transferred, etc. The channel descriptor is described in detail in Section 8.4.3 (p. 52).

In addition to the basic transfer mode, the DMA Controller also supports two advanced transfer modes; ping-pong and scatter-gather. Ping-pong transfers are ideally suited for streaming data for high-speed peripheral communication as the DMA will be ready to retrieve the next incoming data bytes immediately while the processor core is still processing the previous ones (and similarly for outgoing communication). Scatter-gather involves executing a series of tasks from memory and allows sophisticated schemes to be implemented by software.

Using different priority levels for the channels and setting the number of bytes after which the DMA Controller re-arbitrates, it is possible to ensure that timing-critical transfers are serviced on time.

8.4.1 Channel Select Configuration

The channel select block allows selecting which peripheral's request lines (dma_req, dma_sreq) to connect to each DMA channel.

This configuration is done by software through the control registers DMA_CH0_CTRL-DMA_CH7_CTRL, with SOURCESEL and SIGSEL components. SOURCESEL selects which peripheral to listen to and SIGSEL picks which output signals to use from the selected peripheral.

All peripherals are connected to dma_req. When this signal is triggered, the DMA performs a number of transfers as specified by the channel descriptor (2^R). The USARTs are additionally connected to the dma_sreq line. When only dma_sreq is asserted but not dma_req, then the DMA will perform exactly one transfer only (given that dma_sreq is enabled by software).

Note

A DMA channel should not be active when the clock to the selected peripheral is off.

8.4.2 DMA control

8.4.2.1 DMA arbitration rate

You can configure when the controller arbitrates during a DMA transfer. This enables you to reduce the latency to service a higher priority channel.



The controller provides four bits that configure how many AHB bus transfers occur before it re-arbitrates. These bits are known as the R_power bits because the value you enter, R, is raised to the power of two and this determines the arbitration rate. For example, if R = 4 then the arbitration rate is 2^4 , that is, the controller arbitrates every 16 DMA transfers.

Table 8.1 (p. 43) lists the arbitration rates.

Table 8.1. AHB bus transfer arbitration interval

R_power	Arbitrate after x DMA transfers
b0000	x = 1
b0001	x = 2
b0010	x = 4
b0011	x=8
b0100	x = 16
b0101	x = 32
b0110	x = 64
b0111	x = 128
b1000	x = 256
b1001	x = 512
b1010 - b1111	x = 1024
	'

Note

You must take care not to assign a low-priority channel with a large R_power because this prevents the controller from servicing high-priority requests, until it re-arbitrates.

The number of dma transfers N that need to be done is specified by the user. When $N > 2^R$ and is not an integer multiple of 2^R then the controller always performs sequences of 2^R transfers until $N < 2^R$ remain to be transferred. The controller performs the remaining N transfers at the end of the DMA cycle.

You store the value of the R_power bits in the channel control data structure. See Section 8.4.3.3 (p. 55) for more information about the location of the R_power bits in the data structure.

8.4.2.2 Priority

When the controller arbitrates, it determines the next channel to service by using the following information:

- · the channel number
- the priority level, default or high, that is assigned to the channel.

You can configure each channel to use either the default priority level or a high priority level by setting the DMA_CHPRIS register.

Channel number zero has the highest priority and as the channel number increases, the priority of a channel decreases. Table 8.2 (p. 43) lists the DMA channel priority levels in descending order of priority.

Table 8.2. DMA channel priority

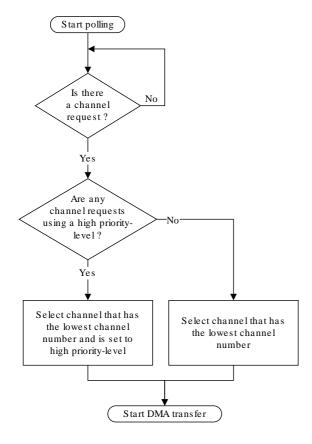
Channel	Priority level	Descending order of
number	setting	channel priority
0	High	Highest-priority DMA channel



Channel	Priority level	Descending order of
number	setting	channel priority
1	High	-
2	High	-
3	High	-
4	High	-
5	High	-
6	High	-
7	High	-
0	Default	-
1	Default	-
2	Default	-
3	Default	-
4	Default	-
5	Default	-
6	Default	-
7	Default	Lowest-priority DMA channel

After a DMA transfer completes, the controller polls all the DMA channels that are available. Figure 8.2 (p. 44) shows the process it uses to determine which DMA transfer to perform next.

Figure 8.2. Polling flowchart





8.4.2.3 DMA cycle types

The cycle_ctrl bits control how the controller performs a DMA cycle. You can set the cycle_ctrl bits as Table 8.3 (p. 45) lists.

Table 8.3. DMA cycle types

cycle_ctrl	Description
b000	Channel control data structure is invalid
b001	Basic DMA transfer
b010	Auto-request
b011	Ping-pong
b100	Memory scatter-gather using the primary data structure
b101	Memory scatter-gather using the alternate data structure
b110	Peripheral scatter-gather using the primary data structure
b111	Peripheral scatter-gather using the alternate data structure

Note

The cycle_ctrl bits are located in the channel_cfg memory location that Section 8.4.3.3 (p. 55) describes.

For all cycle types, the controller arbitrates after 2^R DMA transfers. If you set a low-priority channel with a large 2^R value then it prevents all other channels from performing a DMA transfer, until the low-priority DMA transfer completes. Therefore, you must take care when setting the R_power, that you do not significantly increase the latency for high-priority channels.

8.4.2.3.1 Invalid

After the controller completes a DMA cycle it sets the cycle type to invalid, to prevent it from repeating the same DMA cycle.

8.4.2.3.2 Basic

In this mode, you configure the controller to use either the primary or the alternate data structure. After you enable the channel C and the controller receives a request for this channel, then the flow for this DMA cycle is as follows:

- 1. The controller performs 2^R transfers. If the number of transfers remaining becomes zero, then the flow continues at step 3 (p. 45).
- 2. The controller arbitrates:
 - if a higher-priority channel is requesting service then the controller services that channel
 - if the peripheral or software signals a request to the controller then it continues at step 1 (p. 45).
- 3. The controller sets dma_done[C] HIGH for one HFCORECLK cycle. This indicates to the host processor that the DMA cycle is complete.

8.4.2.3.3 Auto-request

When the controller operates in this mode, it is only necessary for it to receive a single request to enable it to complete the entire DMA cycle. This enables a large data transfer to occur, without significantly increasing the latency for servicing higher priority requests, or requiring multiple requests from the processor or peripheral.

You can configure the controller to use either the primary or the alternate data structure. After you enable the channel C and the controller receives a request for this channel, then the flow for this DMA cycle is as follows:



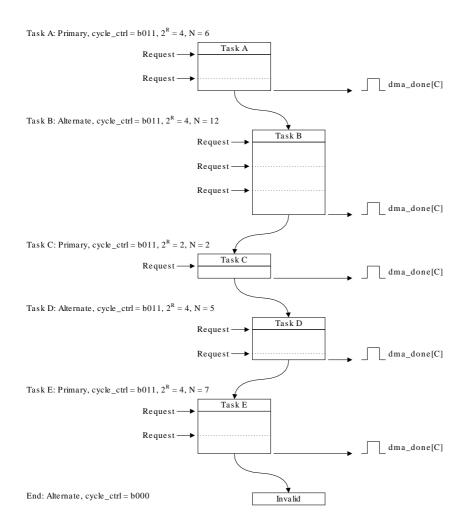
- 1. The controller performs 2^R transfers for channel C. If the number of transfers remaining is zero the flow continues at step 3 (p. 46).
- 2. The controller arbitrates. When channel C has the highest priority then the DMA cycle continues at step 1 (p. 46) .
- 3. The controller sets dma_done[C] HIGH for one HFCORECLK cycle. This indicates to the host processor that the DMA cycle is complete.

8.4.2.3.4 Ping-pong

In ping-pong mode, the controller performs a DMA cycle using one of the data structures (primary or alternate) and it then performs a DMA cycle using the other data structure. The controller continues to switch from primary to alternate to primary... until it reads a data structure that is invalid, or until the host processor disables the channel.

Figure 8.3 (p. 46) shows an example of a ping-pong DMA transaction.

Figure 8.3. Ping-pong example



In Figure 8.3 (p. 46):

Task A 1. The host processor configures the primary data structure for task A.

- 2. The host processor configures the alternate data structure for task B. This enables the controller to immediately switch to task B after task A completes, provided that a higher priority channel does not require servicing.
- 3. The controller receives a request and performs four DMA transfers.



- 4. The controller arbitrates. After the controller receives a request for this channel, the flow continues if the channel has the highest priority.
- 5. The controller performs the remaining two DMA transfers.
- 6. The controller sets <code>dma_done[C]</code> HIGH for one <code>HFCORECLK</code> cycle and enters the arbitration process.

After task A completes, the host processor can configure the primary data structure for task C. This enables the controller to immediately switch to task C after task B completes, provided that a higher priority channel does not require servicing.

After the controller receives a new request for the channel and it has the highest priority then task B commences:

- Task B 7. The controller performs four DMA transfers.
 - 8. The controller arbitrates. After the controller receives a request for this channel, the flow continues if the channel has the highest priority.
 - 9. The controller performs four DMA transfers.
 - 10. The controller arbitrates. After the controller receives a request for this channel, the flow continues if the channel has the highest priority.
 - 11. The controller performs the remaining four DMA transfers.
 - 12. The controller sets $dma_done[C]$ HIGH for one HFCORECLK cycle and enters the arbitration process.

After task B completes, the host processor can configure the alternate data structure for task D.

After the controller receives a new request for the channel and it has the highest priority then task C commences:

Task C 13.The controller performs two DMA transfers.

14.The controller sets dma_done[C] HIGH for one HFCORECLK cycle and enters the arbitration process.

After task C completes, the host processor can configure the primary data structure for task E.

After the controller receives a new request for the channel and it has the highest priority then task D commences:

- Task D 15.The controller performs four DMA transfers.
 - 16. The controller arbitrates. After the controller receives a request for this channel, the flow continues if the channel has the highest priority.
 - 17. The controller performs the remaining DMA transfer.
 - 18. The controller sets $dma_done[C]$ HIGH for one HFCORECLK cycle and enters the arbitration process.

After the controller receives a new request for the channel and it has the highest priority then task E commences:

Task E 19.The controller performs four DMA transfers.

- 20. The controller arbitrates. After the controller receives a request for this channel, the flow continues if the channel has the highest priority.
- 21. The controller performs the remaining three DMA transfers.
- 22. The controller sets $dma_done[C]$ HIGH for one HFCORECLK cycle and enters the arbitration process.



If the controller receives a new request for the channel and it has the highest priority then it attempts to start the next task. However, because the host processor has not configured the alternate data structure, and on completion of task D the controller set the cycle_ctrl bits to b000, then the ping-pong DMA transaction completes.

Note

You can also terminate the ping-pong DMA cycle in Figure 8.3 (p. 46), if you configure task E to be a basic DMA cycle by setting the cycle_ctrl field to 3'b001.

8.4.2.3.5 Memory scatter-gather

In memory scatter-gather mode the controller receives an initial request and then performs four DMA transfers using the primary data structure. After this transfer completes, it starts a DMA cycle using the alternate data structure. After this cycle completes, the controller performs another four DMA transfers using the primary data structure. The controller continues to switch from primary to alternate to primary... until either:

- the host processor configures the alternate data structure for a basic cycle
- it reads an invalid data structure.

Note

After the controller completes the N primary transfers it invalidates the primary data structure by setting the cycle_ctrl field to b000.

The controller only asserts dma_done[C] when the scatter-gather transaction completes using an autorequest cycle.

In scatter-gather mode, the controller uses the primary data structure to program the alternate data structure. Table 8.4 (p. 48) lists the fields of the channel_cfg memory location for the primary data structure, that you must program with constant values and those that can be user defined.

Table 8.4. channel_cfg for a primary data structure, in memory scatter-gather mode

Bit	Field	Value	Description	
Constant-value fields:				
[31:30}	dst_inc	b10	Configures the controller to use word increments for the address	
[29:28]	dst_size	b10	Configures the controller to use word transfers	
[27:26]	src_inc	b10	Configures the controller to use word increments for the address	
[25:24]	src_size	b10	Configures the controller to use word transfers	
[17:14]	R_power	b0010	Configures the controller to perform four DMA transfers	
[3]	next_useburst	0	For a memory scatter-gather DMA cycle, this bit must be set to zero	
[2:0]	cycle_ctrl	b100	Configures the controller to perform a memory scatter-gather DMA cycle	
User defi	ned values:			
[23:21]	dst_prot_ctrl	=	Configures the state of HPROT ¹ when the controller writes the destination data	
[20:18]	src_prot_ctrl	=	Configures the state of HPROT when the controller reads the source data	
[13:4]	n_minus_1	N ²	Configures the controller to perform N DMA transfers, where N is a multiple of four	

ARM PL230 homepage [http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0417a/index.html]

See Section 8.4.3.3 (p. 55) for more information.

Figure 8.4 (p. 49) shows a memory scatter-gather example.

²Because the R_power field is set to four, you must set N to be a multiple of four. The value given by N/4 is the number of times that you must configure the alternate data structure.

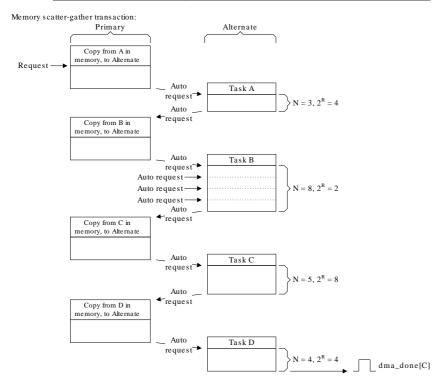


Figure 8.4. Memory scatter-gather example

Initialization: 1. Configure primary to enable the copy A, B, C, and D operations: cycle_ctrl = b100, $2^R = 4$, N = 16.

2. Write the primary source data to memory, using the structure shown in the following table.

	src_data_end_ptr	dst_data_end_ptr	channel_cfg	Unused
Data for Task A	0x0A000000	0x0AE00000	cycle_ctrl = b101, 2 ^R = 4, N = 3	0xXXXXXXXX
Data for Task B	0x0B000000	0x0BE00000	$cycle_ctrl = b101, 2^R = 2, N = 8$	0xXXXXXXXX
Data for Task C	0x0C000000	0x0CE00000	$cycle_ctrl = b101, 2^R = 8, N = 5$	0xXXXXXXXX
Data for Task D	0x0D000000	0x0DE00000	cycle_ctrl = b010, 2 ^R = 4, N = 4	0xXXXXXXXX



In Figure 8.4 (p. 49):

Initialization

- 1. The host processor configures the primary data structure to operate in memory scatter-gather mode by setting cycle_ctrl to b100. Because a data structure for a single channel consists of four words then you must set 2^R to 4. In this example, there are four tasks and therefore N is set to 16.
- 2. The host processor writes the data structure for tasks A, B, C, and D to the memory locations that the primary src_data_end_ptr specifies.
- 3. The host processor enables the channel.

The memory scatter-gather transaction commences when the controller receives a request on $dma_req[$] or a manual request from the host processor. The transaction continues as follows:

Primary, copy A

- 1. After receiving a request, the controller performs four DMA transfers. These transfers write the alternate data structure for task A.
- 2. The controller generates an auto-request for the channel and then arbitrates.

Task A

3. The controller performs task A. After it completes the task, it generates an auto-request for the channel and then arbitrates.

Primary, copy B

- 4. The controller performs four DMA transfers. These transfers write the alternate data structure for task B.
- 5. The controller generates an auto-request for the channel and then arbitrates.

Task B

6. The controller performs task B. After it completes the task, it generates an auto-request for the channel and then arbitrates.

Primary, copy C

7. The controller performs four DMA transfers. These transfers write the alternate data structure for task C.



	8. The controller generates an auto-request for the channel and then arbitrates.
Task C	9. The controller performs task C. After it completes the task, it generates an auto-request for the channel and then arbitrates.
Primary, copy D	10. The controller performs four DMA transfers. These transfers write the alternate data structure for task D.
	11. The controller sets the cycle_ctrl bits of the primary data structure to b000, to indicate that this data structure is now invalid.
	12. The controller generates an auto-request for the channel and then arbitrates.
Task D	13.The controller performs task D using an auto-request cycle.
	14. The controller sets dma_done[C] HIGH for one HFCORECLK cycle and enters the arbitration process.

8.4.2.3.6 Peripheral scatter-gather

In peripheral scatter-gather mode the controller receives an initial request from a peripheral and then it performs four DMA transfers using the primary data structure. It then immediately starts a DMA cycle using the alternate data structure, without re-arbitrating.

Note

These are the only circumstances, where the controller does not enter the arbitration process after completing a transfer using the primary data structure.

After this cycle completes, the controller re-arbitrates and if the controller receives a request from the peripheral that has the highest priority then it performs another four DMA transfers using the primary data structure. It then immediately starts a DMA cycle using the alternate data structure, without rearbitrating. The controller continues to switch from primary to alternate to primary... until either:

- the host processor configures the alternate data structure for a basic cycle
- it reads an invalid data structure.

Note

After the controller completes the N primary transfers it invalidates the primary data structure by setting the cycle_ctrl field to b000.

The controller asserts <code>dma_done[C]</code> when the scatter-gather transaction completes using a basic cycle.

In scatter-gather mode, the controller uses the primary data structure to program the alternate data structure. Table 8.5 (p. 50) lists the fields of the channel_cfg memory location for the primary data structure, that you must program with constant values and those that can be user defined.

Table 8.5. channel_cfg for a primary data structure, in peripheral scatter-gather mode

Bit	Field	Value	Description			
Constant	Constant-value fields:					
[31:30]	dst_inc	b10	Configures the controller to use word increments for the address			
[29:28]	dst_size	b10	Configures the controller to use word transfers			
[27:26]	src_inc	b10	Configures the controller to use word increments for the address			
[25:24]	src_size	b10	Configures the controller to use word transfers			
[17:14]	R_power	b0010	Configures the controller to perform four DMA transfers			
[2:0]	cycle_ctrl	b110	Configures the controller to perform a peripheral scatter-gather DMA cycle			
User defi	User defined values:					
[23:21]	dst_prot_ctrl	-	Configures the state of HPROT when the controller writes the destination data			



Bit	Field	Value	Description
[20:18]	src_prot_ctrl	-	Configures the state of HPROT when the controller reads the source data
[13:4]	n_minus_1	N ¹	Configures the controller to perform N DMA transfers, where N is a multiple of four
[3]	next_useburst	-	When set to 1, the controller sets the chnl_useburst_set [C] bit to 1 after the alternate transfer completes

¹Because the R_power field is set to four, you must set N to be a multiple of four. The value given by N/4 is the number of times that you must configure the alternate data structure.

See Section 8.4.3.3 (p. 55) for more information.

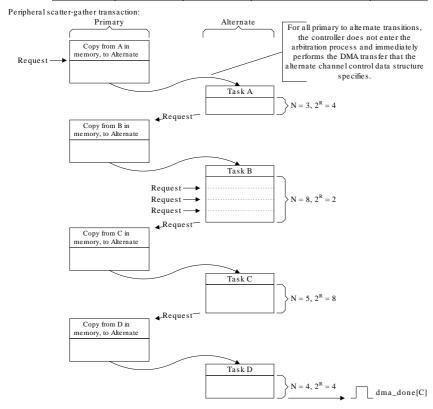
Figure 8.5 (p. 51) shows a peripheral scatter-gather example.

Figure 8.5. Peripheral scatter-gather example

 $Initialization: \ \ 1.\ Configure\ primary\ to\ enable\ the\ copy\ A,\ B,\ C,\ and\ D\ operations: cycle_ctrl=b110,\ 2^R=4,\ N=16.$

2. Write the primary source data in memory, using the structure shown in the following table.

	src_data_end_ptr	dst_data_end_ptr	channel_cfg	Unused
Data for Task A	0x0A000000	0x0AE00000	cycle_ctrl = b111, 2 ^R = 4, N = 3	0xXXXXXXXX
Data for Task B	0x0B000000	0x0BE00000	cycle_ctrl = b111, 2 ^R = 2, N = 8	0xXXXXXXXX
Data for Task C	0x0C000000	0x0CE00000	cycle_ctrl = b111, 2 ^R = 8, N = 5	0xXXXXXXXX
Data for Task D	0x0D000000	0x0DE00000	cycle_ctrl = b001, 2 ^R = 4, N = 4	0xXXXXXXXX



In Figure 8.5 (p. 51):

Initialization

- 1. The host processor configures the primary data structure to operate in peripheral scatter-gather mode by setting cycle_ctrl to b110. Because a data structure for a single channel consists of four words then you must set 2^R to 4. In this example, there are four tasks and therefore N is set to 16.
- 2. The host processor writes the data structure for tasks A, B, C, and D to the memory locations that the primary src_data_end_ptr specifies.
- 3. The host processor enables the channel.

The peripheral scatter-gather transaction commences when the controller receives a request on $dma_req[$]. The transaction continues as follows:



Prin	nary	CO	nv A
	iiai y	, cc	Руи

1. After receiving a request, the controller performs four DMA transfers. These transfers write the alternate data structure for task A.

Task A

- 2. The controller performs task A.
- 3. After the controller completes the task it enters the arbitration process.

After the peripheral issues a new request and it has the highest priority then the process continues with:

Primary, copy B

4. The controller performs four DMA transfers. These transfers write the alternate data structure for task B.

Task B

- 5. The controller performs task B. To enable the controller to complete the task, the peripheral must issue a further three requests.
- 6. After the controller completes the task it enters the arbitration process.

After the peripheral issues a new request and it has the highest priority then the process continues with:

Primary, copy C

7. The controller performs four DMA transfers. These transfers write the alternate data structure for task C.

Task C

- 8. The controller performs task C.
- 9. After the controller completes the task it enters the arbitration process.

After the peripheral issues a new request and it has the highest priority then the process continues with:

Primary, copy D

- 10. The controller performs four DMA transfers. These transfers write the alternate data structure for task D.
 - 11. The controller sets the cycle_ctrl bits of the primary data structure to b000, to indicate that this data structure is now invalid.

Task D

- 12. The controller performs task D using a basic cycle.
- 13.The controller sets dma_done[C] HIGH for one HFCORECLK cycle and enters the arbitration process.

8.4.2.4 Error signaling

If the controller detects an ERROR response on the AHB-Lite master interface, it:

- disables the channel that corresponds to the ERROR
- sets dma_err HIGH.

After the host processor detects that dma_err is HIGH, it must check which channel was active when the ERROR occurred. It can do this by:

1. Reading the DMA_CHENS register to create a list of disabled channels.

When a channel asserts $dma_done[$] then the controller disables the channel. The program running on the host processor must always keep a record of which channels have recently asserted their $dma_done[$] outputs.

2. It must compare the disabled channels list from step 1 (p. 52), with the record of the channels that have recently set their dma_done[] outputs. The channel with no record of dma_done[C] being set is the channel that the ERROR occurred on.

8.4.3 Channel control data structure

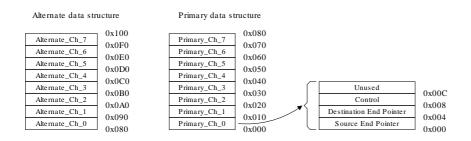
You must provide an area of system memory to contain the channel control data structure. This system memory must:



- provide a contiguous area of system memory that the controller and host processor can access
- have a base address that is an integer multiple of the total size of the channel control data structure.

Figure 8.6 (p. 53) shows the memory that the controller requires for the channel control data structure, when all 8 channels and the optional alternate data structure are in use.

Figure 8.6. Memory map for 8 channels, including the alternate data structure



This structure in Figure 8.6 (p. 53) uses 256 bytes of system memory. The controller uses the lower 8 address bits to enable it to access all of the elements in the structure and therefore the base address must be at 0xxxxxxx00.

You can configure the base address for the primary data structure by writing the appropriate value in the DMA_CTRLBASE register.

You do not need to set aside the full 256 bytes if all dma channels are not used or if all alternate descriptors are not used. If, for example, only 4 channels are used and they only need the primary descriptors, then only 64 bytes need to be set aside.

Table 8.6 (p. 53) lists the address bits that the controller uses when it accesses the elements of the channel control data structure.

Table 8.6. Address bit settings for the channel control data structure

Address bits				·	
	[7]	[6]	[5]	[4]	[3:0]
	А	C[2]	C[1]	C[0]	0x0, 0x4, or 0x8

Where:

A Selects one of the channel control data structures:

A = 0 Selects the primary data structure.

A = 1 Selects the alternate data structure.

C[2:0] Selects the DMA channel.

Address[3:0] Selects one of the control elements:

0x0 Selects the source data end pointer.

0x4 Selects the destination data end pointer.

0x8 Selects the control data configuration.

0xC The controller does not access this address location. If required, you can enable the host processor to use this memory location as system memory.

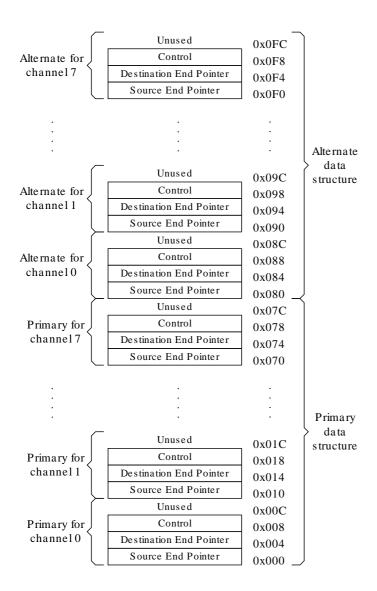
Note

It is not necessary for you to calculate the base address of the alternate data structure because the DMA_ALTCTRLBASE register provides this information.



Figure 8.7 (p. 54) shows a detailed memory map of the descriptor structure.

Figure 8.7. Detailed memory map for the 8 channels, including the alternate data structure



The controller uses the system memory to enable it to access two pointers and the control information that it requires for each channel. The following subsections will describe these 32-bit memory locations and how the controller calculates the DMA transfer address.

8.4.3.1 Source data end pointer

The src_data_end_ptr memory location contains a pointer to the end address of the source data. Figure 8.7 (p. 54) lists the bit assignments for this memory location.

Table 8.7. src_data_end_ptr bit assignments

Bit	Name	Description
[31:0]	src_data_end_ptr	Pointer to the end address of the source data

Before the controller can perform a DMA transfer, you must program this memory location with the end address of the source data. The controller reads this memory location when it starts a 2^R DMA transfer.

Note

The controller does not write to this memory location.



8.4.3.2 Destination data end pointer

The dst_data_end_ptr memory location contains a pointer to the end address of the destination data. Table 8.8 (p. 55) lists the bit assignments for this memory location.

Table 8.8. dst_data_end_ptr bit assignments

Bit	Name	Description
[31:0]	dst_data_end_ptr	Pointer to the end address of the destination data

Before the controller can perform a DMA transfer, you must program this memory location with the end address of the destination data. The controller reads this memory location when it starts a 2^R DMA transfer.

Note

The controller does not write to this memory location.

8.4.3.3 Control data configuration

For each DMA transfer, the channel_cfg memory location provides the control information for the controller. Figure 8.8 (p. 55) shows the bit assignments for this memory location.

Figure 8.8. channel_cfg bit assignments

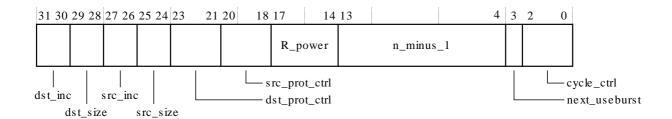


Table 8.9 (p. 55) lists the bit assignments for this memory location.

Table 8.9. channel_cfg bit assignments

Bit	Name	Description			
[31:30]	dst_inc	Destination address increment.			
		The address increment depends	s on the source data width as follows:		
		Source data width = byte	b00 = byte.		
			b01 = halfword.		
			b10 = word.		
			b11 = no increment. Address remains set to the value that the dst_data_end_ptr memory location contains.		
		Source data width = halfword	b00 = reserved.		
			b01 = halfword.		
			b10 = word.		
			b11 = no increment. Address remains set to the value that the dst_data_end_ptr memory location contains.		
		Source data width = word	b00 = reserved.		
			b01 = reserved.		
			b10 = word.		



Bit	Name	Description				
				b11 = no increment. Address remains set to the value that the dst_data_end_ptr memory location contains.		
[29:28]	dst_size	Destination	data size.			
		Note	ou must set dst_size	to contain the same value that src_size contains.		
[27:26]	src_inc		bits to control the source address increment. The address increment depends on the data width as follows:			
		Source data	a width = byte	b00 = byte.		
				b01 = halfword.		
				b10 = word.		
		Source data	a width = halfword	b11 = no increment. Address remains set to the value that the src_data_end_ptr memory location contains. b00 = reserved.		
				b01 = halfword.		
			b10 = word.			
		Source data	a width = word	b11 = no increment. Address remains set to the value that the src_data_end_ptr memory location contains. b00 = reserved.		
				b01 = reserved.		
				b10 = word.		
				b11 = no increment. Address remains set to the value that the src_data_end_ptr memory location contains.		
[25:24]	src_size	Set the bits	to match the size of	the source data:		
		b00 = byte				
		b01 = halfw	rord			
		b10 = word	b10 = word			
		b11 = reserved.				
[23:21]	dst_prot_ctrl	Set the bits to control the state of HPROT when the controller writes the destination data.				
		Bit [23] Bit [22] Bit [21]	[22] This bit has no effect on the DMA.			
			0 = HPROT is LOW	and the access is non-privileged.		
				I and the access is privileged.		
[20:18]	src_prot_ctrl	Set the bits	to control the state of	of HPROT when the controller reads the source data.		
		Bit [20] Bit [19] Bit [18]	This bit has no effective This bit has no effective Controls the state of			
		[]		and the access is non-privileged.		
				and the access is privileged.		
[17:14]	R_power			any DMA transfers can occur before the controller re-arbitrates.		
		b0000 b0001 b0010 b0011 b0100 b0101 b0110 b0111	Arbitrates after Arbitrates after Arbitrates after Arbitrates after Arbitrates after Arbitrates after	r each DMA transfer. 2 DMA transfers. 4 DMA transfers. 8 DMA transfers. 116 DMA transfers. 132 DMA transfers. 164 DMA transfers. 1128 DMA transfers.		



Bit	Name	Description	n		
		b1000 b1001 b1010 - b1			
[13:4]	n_minus_1	during the DMA transfer because the maximum transfer size is 1024. Prior to the DMA cycle commencing, these bits represent the total number of DMA transfers that the DMA cycle contains. You must set these bits according to the size of DMA cycle that			
		you require	e. value indicates the number of DMA transfers, minus one. The possible values are:		
			00 = 1 DMA transfer		
			00 = 1 DMA transfers		
			10 = 3 DMA transfers		
			11 = 4 DMA transfers		
			00 = 5 DMA transfers		
		1 0000000	00 = 5 DIVIA transiers		
		•			
		•			
			11 = 1024 DMA transfers.		
		enables th	oller updates this field immediately prior to it entering the arbitration process. This the controller to store the number of outstanding DMA transfers that are necessary to the DMA cycle.		
[3]	next_useburst		the chnl_useburst_set [C] bit is set to a 1, when the controller is performing a scatter-gather and is completing a DMA cycle that uses the alternate data structure.		
		Note			
		s	mmediately prior to completion of the DMA cycle that the alternate data structure specifies, the controller sets the chnl_useburst_set [C] bit to 0 if the number of remaining transfers is less than 2 ^R . The setting of the next_useburst bit controls if the controller performs an additional modification of the chnl_useburst_set [C] bit.		
			ral scatter-gather DMA cycle then after the DMA cycle that uses the alternate data completes, either:		
		$0 =$ the controller does not change the value of the chnl_useburst_set [C] bit. If the chnl_useburst_set [C] bit is 0 then for all the remaining DMA cycles in the peripheral gather transaction, the controller responds to requests on $dma_req[]$ and $dma_sreq[]$ when it performs a DMA cycle that uses an alternate data structure.			
		cycles in tl	ntroller sets the chnl_useburst_set [C] bit to a 1. Therefore, for the remaining DMA he peripheral scatter-gather transaction, the controller only responds to requests on [], when it performs a DMA cycle that uses an alternate data structure.		
[2:0]	cycle_ctrl	The opera	ting mode of the DMA cycle. The modes are:		
		b001 Ba	op. Indicates that the data structure is invalid. ssic. The controller must receive a new request, prior to it entering the arbitration		
		b010 Au du	ocess, to enable the DMA cycle to complete. Ito-request. The controller automatically inserts a request for the appropriate channel iring the arbitration process. This means that the initial request is sufficient to enable a DMA cycle to complete.		
		b011 Pir the the da coi ho	ng-pong. The controller performs a DMA cycle using one of the data structures. After a DMA cycle completes, it performs a DMA cycle using the other data structure. After a DMA cycle completes and provided that the host processor has updated the origina ta structure, it performs a DMA cycle using the original data structure. The controller ntinues to perform DMA cycles until it either reads an invalid data structure or the last processor changes the cycle_ctrl bits to b001 or b010. See Section 8.4.2.3.4 (p.		
		46 b100 Me	emory scatter/gather. See Section 8.4.2.3.5 (p. 48).		
		va	hen the controller operates in memory scatter-gather mode, you must only use this lue in the primary data structure. emory scatter/gather. See Section 8.4.2.3.5 (p. 48).		
		Wi	hen the controller operates in memory scatter-gather mode, you must only use this lue in the alternate data structure.		
		b110 Pe	eripheral scatter/gather. See Section 8.4.2.3.6 (p. 50).		



Bit	Name	Description
		When the controller operates in peripheral scatter-gather mode, you must only use this value in the primary data structure. b111 Peripheral scatter/gather. See Section 8.4.2.3.6 (p. 50).
		When the controller operates in peripheral scatter-gather mode, you must only use this value in the alternate data structure.

At the start of a DMA cycle, or 2^R DMA transfer, the controller fetches the channel_cfg from system memory. After it performs 2^R , or N, transfers it stores the updated channel_cfg in system memory.

The controller does not support a dst size value that is different to the src size value. If it detects a mismatch in these values, it uses the src_size value for source and destination and when it next updates the n minus 1 field, it also sets the dst size field to the same as the src size field.

After the controller completes the N transfers it sets the cycle_ctrl field to b000, to indicate that the channel_cfg data is invalid. This prevents it from repeating the same DMA transfer.

8.4.3.4 Address calculation

To calculate the source address of a DMA transfer, the controller performs a left shift operation on the n_minus_1 value by a shift amount that src_inc specifies, and then subtracts the resulting value from the source data end pointer. Similarly, to calculate the destination address of a DMA transfer, it performs a left shift operation on the n_minus_1 value by a shift amount that dst_inc specifies, and then subtracts the resulting value from the destination end pointer.

Depending on the value of src inc and dst inc, the source address and destination address can be calculated using the equations:

 $src_inc = b00$ and $dst_inc = b00$

- source address = src_data_end_ptr n_minus_1
- src_inc = b01 and dst_inc = b01
- destination address = dst_data_end_ptr n_minus_1.
- source address = src_data_end_ptr (n_minus_1 << 1)
- $src_inc = b10$ and $dst_inc = b10$
- destination address = dst_data_end_ptr (n_minus_1 << 1).
- source address = src_data_end_ptr (n_minus_1 << 2) destination address = dst_data_end_ptr - (n_minus_1 << 2).
- src_inc = b11 and dst_inc = b11
- source address = src_data_end_ptr
- destination address = dst_data_end_ptr.

Table 8.10 (p. 58) lists the destination addresses for a DMA cycle of six words.

Table 8.10. DMA cycle of six words using a word increment

src_size = b10, dst_inc = b10, n_minus_1 = b101, cycle_ctrl = 1						
	End Pointer	Count	Difference ¹	Address		
	0x2AC	5	0x14	0x298		
	0x2AC	4	0x10	0x29C		
DMA transfers	0x2AC	3	0xC	0x2A0		
Divis Circulororo	0x2AC	2	0x8	0x2A4		
	0x2AC	1	0x4	0x2A8		
	0x2AC	0	0x0	0x2AC		
Final values of channel_cfg, after the DMA cycle						

^{&#}x27;This value is the result of count being shifted left by the value of dst_inc.



Table 8.11 (p. 59) lists the destination addresses for a DMA transfer of 12 bytes using a halfword increment.

Table 8.11. DMA cycle of 12 bytes using a halfword increment

	End Pointer 0x5E7	Count	Difference ¹	Address
	0x5E7			
		11	0x16	0x5D1
	0x5E7	10	0x14	0x5D3
DMA transfers	0x5E7	9	0x12	0x5D5
2 (0x5E7	8	0x10	0x5D7
	0x5E7	7	0xE	0x5D9
	0x5E7	6	0xC	0x5DB
	0x5E7	5	0xA	0x5DD
	0x5E7	4	0x8	0x5DF
Values of channel_cfg	after 2 ^R DMA transf	fers		
src_size = b00, dst_ind	c = b01, n_minus_1 =	= b011, cycl	e_ctrl = 1, R_pov	ver = b11
	End Pointer	Count	Difference	Address
	0x5E7	3	0x6	0x5E1
	0x5E7	2	0x4	0x5E3
DMA transfers	0x5E7	1	0x2	0x5E5
	0x5E7	0	0x0	0x5E7

¹This value is the result of count being shifted left by the value of dst_inc.

8.4.4 Interaction with the EMU

The DMA interacts with the Energy Management Unit (EMU) to allow transfers from , e.g., the LEUART to occur in EM2. The EMU can wake up the DMA sufficiently long to allow data transfers to occur. See section "DMA Support" in the LEUART documentation.

8.4.5 Interrupts

The PL230 dma_done[n:0] signals (one for each channel) as well as the dma_err signal, are available as interrupts to the Cortex-M3 core. They are combined into one interrupt vector, DMA_INT. If the interrupt for the DMA is enabled in the ARM Cortex-M3 core, an interrupt will be made if one or more of the interrupt flags in DMA_IF and their corresponding bits in DMA_IEN are set.

8.5 Examples

A basic example of how to program the DMA for transferring 42 bytes from the USART1 to memory location 0x20003420. Assumes that the channel 0 is currently disabled, and that the DMA_ALTCTRLBASE register has already been configured.

²After the controller completes the DMA cycle it invalidates the channel_cfg memory location by clearing the cycle_ctrl field.



Example 8.1. DMA Transfer

- 1. Configure the channel select for using USART1 with DMA channel 0
 - a. Write SOURCESEL=0b001101 and SIGSEL=XX to DMA CHCTRL0
- 2. Configure the primary channel descriptor for DMA channel 0
 - a. Write XX (read address of USART1) to src_data_end_ptr
 - b. Write 0x20003420 + 40 to dst_data_end_ptr c
 - c. Write these values to channel_cfg for channel 0:
 - i. dst_inc=b01 (destination halfword address increment)
 - ii. dst_size=b01 (halfword transfer size)
 - iii. src_inc=b11 (no address increment for source)
 - iv. src_size=01 (halfword transfer size)
 - v. dst_prot_ctrl=000 (no cache/buffer/privilege)
 - vi. src_prot_ctrl=000 (no cache/buffer/privilege)
 - vii.R_power=b0000 (arbitrate after each DMA transfer)
 - viiin_minus_1=d20 (transfer 21 halfwords)
 - ix. next_useburst=b0 (not applicable)
 - x. cycle_ctrl=b001 (basic operating mode)
- 3. Enable the DMA
 - a. Write EN=1 to DMA_CONFIG
- 4. Disable the single requests for channel 0 (i.e., do not react to data available, wait for buffer full)
 - a. Write DMA_CHUSEBURSTS[0]=1
- 5. Enable buffer-full requests for channel 0
 - a. Write DMA_CHREQMASKC[0]=1
- 6. Use the primary data structure for channel 0
 - a. Write DMA_CHALTC[0]=1
- 7. Enable channel 0
 - a. Write DMA_CHENS[0]=1



8.6 Register Map

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

Offset	Name	Туре	Description
0x000	DMA_STATUS	R	DMA Status Registers
0x004	DMA_CONFIG	W	DMA Configuration Register
0x008	DMA_CTRLBASE	RW	Channel Control Data Base Pointer Register
0x00C	DMA_ALTCTRLBASE	R	Channel Alternate Control Data Base Pointer Register
0x010	DMA_CHWAITSTATUS	R	Channel Wait on Request Status Register
0x014	DMA_CHSWREQ	W1	Channel Software Request Register
0x018	DMA_CHUSEBURSTS	RW1H	Channel Useburst Set Register
0x01C	DMA_CHUSEBURSTC	W1	Channel Useburst Clear Register
0x020	DMA_CHREQMASKS	RW1	Channel Request Mask Set Register
0x024	DMA_CHREQMASKC	W1	Channel Request Mask Clear Register
0x028	DMA_CHENS	RW1	Channel Enable Set Register
0x02C	DMA_CHENC	W1	Channel Enable Clear Register
0x030	DMA_CHALTS	RW1	Channel Alternate Set Register
0x034	DMA_CHALTC	W1	Channel Alternate Clear Register
0x038	DMA_CHPRIS	RW1	Channel Priority Set Register
0x03C	DMA_CHPRIC	W1	Channel Priority Clear Register
0x04C	DMA_ERRORC	RW	Bus Error Clear Register
0xE10	DMA_CHREQSTATUS	R	Channel Request Status
0xE18	DMA_CHSREQSTATUS	R	Channel Single Request Status
0x1000	DMA_IF	R	Interrupt Flag Register
0x1004	DMA_IFS	W1	Interrupt Flag Set Register
0x1008	DMA_IFC	W1	Interrupt Flag Clear Register
0x100C	DMA_IEN	RW	Interrupt Enable register
0x1100	DMA_CH0_CTRL	RW	Channel Control Register
0x1104	DMA_CH1_CTRL	RW	Channel Control Register
0x1108	DMA_CH2_CTRL	RW	Channel Control Register
0x110C	DMA_CH3_CTRL	RW	Channel Control Register
0x1110	DMA_CH4_CTRL	RW	Channel Control Register
0x1114	DMA_CH5_CTRL	RW	Channel Control Register
0x1118	DMA_CH6_CTRL	RW	Channel Control Register
0x111C	DMA_CH7_CTRL	RW	Channel Control Register



8.7 Register Description

8.7.1 DMA_STATUS - DMA Status Registers

Offset												,			Bi	t Pc	siti	on	-							-						
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	8	7	9	2	4	က	2	1	0
Reset														0×07													0x0					0
Access														~												-	ď					~
Name														CHNUM													STATE					Z

Bit	Name	Reset	Access	Description
31:21	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
20:16	CHNUM	0x07	R	Channel Number
	Number of availabl	e DMA channels minus	one.	
15:8	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:4	STATE	0x0	R	Control Current State
	State can be one o	f the following. Higher	/alues (11-15) are	e undefined.
	V 1		1_	

Value	Mode	Description
0	IDLE	Idle
1	RDCHCTRLDATA	Reading channel controller data
2	RDSRCENDPTR	Reading source data end pointer
3	RDDSTENDPTR	Reading destination data end pointer
4	RDSRCDATA	Reading source data
5	WRDSTDATA	Writing destination data
6	WAITREQCLR	Waiting for DMA request to clear
7	WRCHCTRLDATA	Writing channel controller data
8	STALLED	Stalled
9	DONE	Done
10	PERSCATTRANS	Peripheral scatter-gather transition

3:1 Reserved To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

0 EN 0 R DMA Enable Status

When this bit is 1, the DMA is enabled.

8.7.2 DMA_CONFIG - DMA Configuration Register

Offset															Bi	t Po	siti	on														
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	8	7	9	2	4	က	2	-	0
Reset			•	•						•	•												•				0					0
Access																											≯					>
Name																											CHPROT					N N

Bit	Name	Reset Access Description
31:6	Reserved	To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)
5	CHPROT	0 W Channel Protection Control



must be 0.

Bit	Name	Reset	Access	Description
				rivileged or not. When CHPROT = 1 then HPROT is HIGH and the access d the access is non-privileged.
4:1	Reserved	To ensure co	ompatibility with fo	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	EN	0	W	Enable DMA
	Set this bit to enab	le the DMA controller.		

8.7.3 DMA_CTRLBASE - Channel Control Data Base Pointer Register

	_																														
Offset															Bit	Posit	ion														
0x008	31	30	59	28	27	56	25	24	23	22	21	20	6 9	28	17	15	4	13	12	7	10	0	8	7	9	2	4	т	2	-	0
Reset																00000000x0				I							I				
Access			δ.																												
Name																CTRLBASE															
Bit	Na	me						Re	set			Α	ссе	ss		Desc	ripti	on													
31:0	СТ	RLBASE 0x00000000 RW Channel Control Data Base Pointer																													
		e base pointer for a location in system memory that holds the channel control data structure. This register must be written to point a location in system memory with the channel control data structure before the DMA can be used. Note that ctrl_base_ptr[8:0]																													

8.7.4 DMA_ALTCTRLBASE - Channel Alternate Control Data Base Pointer Register

Offset															Bit	t Po	siti	on														
0x00C	31	30	29	28	27	26	22	24	23	22	7	20	19	8	17	16	15	4	13	12	=	10	6	8	7	9	2	4	ო	2	-	0
Reset																08000000	00000000															
Access			α																													
Name																a a lator iv	ALICINEDASE															
Bit	Na	ıme						Re	set			Α	CCE	ess		De	scr	iptic	on													
31:0	AL ⁻	LTCTRLBASE 0x00000080 R Channel Alternate Control Data Base Pointer																														
	The	e base address of the alternate data structure. This register will read as DMA_CTRLBASE + 0x80.																														



8.7.5 DMA_CHWAITSTATUS - Channel Wait on Request Status Register

Offset					,			,	,						Bi	t Po	siti	on				,										
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	8	7	9	2	4	က	2	-	0
Reset			•								•	•						•			•		•		-	-	-	-	-	-	-	-
Access																									~	2	~	~	~	~	~	~
Name																									CH7WAITSTATUS	CH6WAITSTATUS	CH5WAITSTATUS	CH4WAITSTATUS	CH3WAITSTATUS	CH2WAITSTATUS	CH1WAITSTATUS	CHOWAITSTATUS

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure con	npatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	CH7WAITSTATUS	1	R	Channel 7 Wait on Request Status
	Status for wait on reques	st for channel 7.		
6	CH6WAITSTATUS	1	R	Channel 6 Wait on Request Status
	Status for wait on reques	st for channel 6.		
5	CH5WAITSTATUS	1	R	Channel 5 Wait on Request Status
	Status for wait on reques	st for channel 5.		
4	CH4WAITSTATUS	1	R	Channel 4 Wait on Request Status
	Status for wait on reques	st for channel 4.		
3	CH3WAITSTATUS	1	R	Channel 3 Wait on Request Status
	Status for wait on reques	st for channel 3.		
2	CH2WAITSTATUS	1	R	Channel 2 Wait on Request Status
	Status for wait on reques	st for channel 2.		
1	CH1WAITSTATUS	1	R	Channel 1 Wait on Request Status
	Status for wait on reques	st for channel 1.		
0	CH0WAITSTATUS	1	R	Channel 0 Wait on Request Status
	Status for wait on reques	st for channel 0.		

8.7.6 DMA_CHSWREQ - Channel Software Request Register

Offset															Bi	t Po	siti	on														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	-	10	0	80	7	9	2	4	က	2	-	0
Reset																•					•				0	0	0	0	0	0	0	0
Access																									W 1	W N	W	W	X	M	W	M
Name																									CH7SWREQ	CH6SWREQ	CH5SWREQ	CH4SWREQ	CH3SWREQ	CH2SWREQ	CH1SWREQ	CHOSWREQ

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	CH7SWREQ	0	W1	Channel 7 Software Request
	Write 1 to this bit to	generate a DMA requ	est for this channe	el.
6	CH6SWREQ	0	W1	Channel 6 Software Request
	Write 1 to this bit to	generate a DMA requ	est for this channe	el.



Bit	Name	Reset	Access	Description
5	CH5SWREQ	0	W1	Channel 5 Software Request
	Write 1 to this bit to	generate a DMA requ	est for this channe).
4	CH4SWREQ	0	W1	Channel 4 Software Request
	Write 1 to this bit to	generate a DMA requ	est for this channe	રો.
3	CH3SWREQ	0	W1	Channel 3 Software Request
	Write 1 to this bit to	generate a DMA requ	est for this channe).
2	CH2SWREQ	0	W1	Channel 2 Software Request
	Write 1 to this bit to	generate a DMA requ	est for this channe).
1	CH1SWREQ	0	W1	Channel 1 Software Request
	Write 1 to this bit to	generate a DMA requ	est for this channe	કો.
0	CH0SWREQ	0	W1	Channel 0 Software Request
	Write 1 to this bit to	generate a DMA requ	est for this channe	કો.

8.7.7 DMA_CHUSEBURSTS - Channel Useburst Set Register

Offset															Bi	t Po	siti	on														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	∞	7	9	2	4	က	2	-	0
Reset																									0	0	0	0	0	0	0	0
Access																									RW1H							
Name																									CH7USEBURSTS	CH6USEBURSTS	CH5USEBURSTS	CH4USEBURSTS	CH3USEBURSTS	CH2USEBURSTS	CH1USEBURSTS	CHOUSEBURSTS

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure comp	atibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	CH7USEBURSTS	0	RW1H	Channel 7 Useburst Set
	See description for chann	el 0.		
6	CH6USEBURSTS	0	RW1H	Channel 6 Useburst Set
	See description for chann	el 0.		
5	CH5USEBURSTS	0	RW1H	Channel 5 Useburst Set
	See description for chann	el 0.		
4	CH4USEBURSTS	0	RW1H	Channel 4 Useburst Set
	See description for chann	el 0.		
3	CH3USEBURSTS	0	RW1H	Channel 3 Useburst Set
	See description for chann	el 0.		
2	CH2USEBURSTS	0	RW1H	Channel 2 Useburst Set
	See description for chann	el 0.		
1	CH1USEBURSTS	0	RW1H	Channel 1 Useburst Set
	See description for chann	el 0.		
0	CH0USEBURSTS	0	RW1H	Channel 0 Useburst Set

Write to 1 to enable the useburst setting for this channel. Reading returns the useburst status. After the penultimate 2^R transfer completes, if the number of remaining transfers, N, is less than 2^R then the controller resets the chnl_useburst_set bit to 0. This enables you to complete the remaining transfers using dma_req[] or dma_sreq[]. In peripheral scatter-gather mode, if the next_useburst bit is set in channel_cfg then the controller sets the chnl_useburst_set[C] bit to a 1, when it completes the DMA cycle that uses the alternate data structure.



Bit	Name	Reset Acc	cess	Description
	Value	Mode	D	Description
	0	SINGLEANDBURST	С	Channel responds to both single and burst requests
	1	BURSTONLY	С	Channel responds to burst requests only

8.7.8 DMA_CHUSEBURSTC - Channel Useburst Clear Register

Offset															Bi	t Po	siti	on														
0x01C	33	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	ю	2	-	0
Reset																									0	0	0	0	0	0	0	0
Access																									W1	W1	M	W1	W1	W1	W1	W1
Name																									CH7USEBURSTC	CH6USEBURSTC	CH5USEBURSTC	CH4USEBURSTC	CH3USEBURSTC	CH2USEBURSTC	CH1USEBURSTC	CH0USEBURSTC

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure comp	patibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	CH7USEBURSTC	0	W1	Channel 7 Useburst Clear
	Write to 1 to disable usebu	ırst setting for this	channel.	
6	CH6USEBURSTC	0	W1	Channel 6 Useburst Clear
	Write to 1 to disable usebu	ırst setting for this	channel.	
5	CH5USEBURSTC	0	W1	Channel 5 Useburst Clear
	Write to 1 to disable usebu	ırst setting for this	channel.	
4	CH4USEBURSTC	0	W1	Channel 4 Useburst Clear
	Write to 1 to disable usebu	ırst setting for this	channel.	
3	CH3USEBURSTC	0	W1	Channel 3 Useburst Clear
	Write to 1 to disable usebu	ırst setting for this	channel.	
2	CH2USEBURSTC	0	W1	Channel 2 Useburst Clear
	Write to 1 to disable usebu	ırst setting for this	channel.	
1	CH1USEBURSTC	0	W1	Channel 1 Useburst Clear
	Write to 1 to disable usebu	irst setting for this	channel.	
0	CH0USEBURSTC	0	W1	Channel 0 Useburst Clear
	Write to 1 to disable usebu	ırst setting for this	channel.	

8.7.9 DMA_CHREQMASKS - Channel Request Mask Set Register

Offset															Bi	t Po	siti	on														
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	0	æ	7	9	2	4	ю	2	-	0
Reset															•			•					•		0	0	0	0	0	0	0	0
Access																									RW1							
Name																									CH7REQMASKS	CH6REQMASKS	CH5REQMASKS	CH4REQMASKS	CH3REQMASKS	CH2REQMASKS	CH1REQMASKS	CHOREQMASKS



Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	CH7REQMASKS	0	RW1	Channel 7 Request Mask Set
	Write to 1 to disable pe	ripheral requests fo	or this channel.	
6	CH6REQMASKS	0	RW1	Channel 6 Request Mask Set
	Write to 1 to disable pe	ripheral requests fo	or this channel.	
5	CH5REQMASKS	0	RW1	Channel 5 Request Mask Set
	Write to 1 to disable pe	ripheral requests fo	or this channel.	
4	CH4REQMASKS	0	RW1	Channel 4 Request Mask Set
	Write to 1 to disable pe	ripheral requests fo	or this channel.	
3	CH3REQMASKS	0	RW1	Channel 3 Request Mask Set
	Write to 1 to disable pe	ripheral requests fo	or this channel.	
2	CH2REQMASKS	0	RW1	Channel 2 Request Mask Set
	Write to 1 to disable pe	ripheral requests fo	or this channel.	
1	CH1REQMASKS	0	RW1	Channel 1 Request Mask Set
	Write to 1 to disable pe	ripheral requests fo	or this channel.	
0	CH0REQMASKS	0	RW1	Channel 0 Request Mask Set
	Write to 1 to disable pe	ripheral requests fo	or this channel.	

8.7.10 DMA_CHREQMASKC - Channel Request Mask Clear Register

Offset															Bi	t Po	siti	on														
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	-	10	6	80	7	9	2	4	က	2	-	0
Reset			•	•							•	•						•			•		•		0	0	0	0	0	0	0	0
Access																									W1	W1	M	W1	W1	W1	W1	W
Name																									CH7REQMASKC	CH6REQMASKC	CH5REQMASKC	CH4REQMASKC	CH3REQMASKC	CH2REQMASKC	CH1REQMASKC	CHOREQMASKC

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3,
7	CH7REQMASKC	0	W1	Channel 7 Request Mask Clear
	Write to 1 to enable per	ripheral requests fo	or this channel.	
6	CH6REQMASKC	0	W1	Channel 6 Request Mask Clear
	Write to 1 to enable per	ripheral requests fo	or this channel.	
5	CH5REQMASKC	0	W1	Channel 5 Request Mask Clear
	Write to 1 to enable pe	ripheral requests fo	or this channel.	
4	CH4REQMASKC	0	W1	Channel 4 Request Mask Clear
	Write to 1 to enable per	ripheral requests fo	or this channel.	
3	CH3REQMASKC	0	W1	Channel 3 Request Mask Clear
	Write to 1 to enable pe	ripheral requests fo	or this channel.	
2	CH2REQMASKC	0	W1	Channel 2 Request Mask Clear
	Write to 1 to enable pe	ripheral requests fo	or this channel.	
1	CH1REQMASKC	0	W1	Channel 1 Request Mask Clear
	Write to 1 to enable pe	ripheral requests fo	or this channel.	



Bit	Name	Reset	Access	Description
0	CH0REQMASKC	0	W1	Channel 0 Request Mask Clear
	Write to 1 to enable peri	pheral requests for	or this channel.	

8.7.11 DMA_CHENS - Channel Enable Set Register

Offset															Bi	t Po	siti	on														
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	-	10	6	œ	7	9	2	4	ю	2	-	0
Reset																									0	0	0	0	0	0	0	0
Access																									RW1							
Name																									CH7ENS	CH6ENS	CH5ENS	CH4ENS	CH3ENS	CHZENS	CH1ENS	CHOENS

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	CH7ENS	0	RW1	Channel 7 Enable Set
	Write to 1 to enable t	his channel. Reading	returns the enable	e status of the channel.
6	CH6ENS	0	RW1	Channel 6 Enable Set
	Write to 1 to enable t	his channel. Reading	returns the enable	e status of the channel.
5	CH5ENS	0	RW1	Channel 5 Enable Set
	Write to 1 to enable t	his channel. Reading	returns the enable	e status of the channel.
4	CH4ENS	0	RW1	Channel 4 Enable Set
	Write to 1 to enable t	his channel. Reading	returns the enable	e status of the channel.
3	CH3ENS	0	RW1	Channel 3 Enable Set
	Write to 1 to enable t	his channel. Reading	returns the enable	e status of the channel.
2	CH2ENS	0	RW1	Channel 2 Enable Set
	Write to 1 to enable t	his channel. Reading	returns the enable	e status of the channel.
1	CH1ENS	0	RW1	Channel 1 Enable Set
	Write to 1 to enable t	his channel. Reading	returns the enable	e status of the channel.
0	CH0ENS	0	RW1	Channel 0 Enable Set
	Write to 1 to enable t	his channel. Reading	returns the enable	e status of the channel.

8.7.12 DMA_CHENC - Channel Enable Clear Register

Offset				,				,							Bi	t Pc	siti	on					,	,							,	
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	0	8	7	9	2	4	ю	2	-	0
Reset					•											•							•		0	0	0	0	0	0	0	0
Access																									W	W1	W	W	W 1	W	W	W
Name																									CH7ENC	CH6ENC	CHSENC	CH4ENC	CH3ENC	CHZENC	CH1ENC	CHOENC

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)



Bit	Name	Reset	Access	Description
7	CH7ENC	0	W1	Channel 7 Enable Clear
	Write to 1 to disab	le this channel. See also	description for o	channel 0.
6	CH6ENC	0	W1	Channel 6 Enable Clear
	Write to 1 to disab	le this channel. See also	description for o	channel 0.
5	CH5ENC	0	W1	Channel 5 Enable Clear
	Write to 1 to disab	le this channel. See also	description for o	channel 0.
4	CH4ENC	0	W1	Channel 4 Enable Clear
	Write to 1 to disab	le this channel. See also	description for o	channel 0.
3	CH3ENC	0	W1	Channel 3 Enable Clear
	Write to 1 to disab	le this channel. See also	description for o	channel 0.
2	CH2ENC	0	W1	Channel 2 Enable Clear
	Write to 1 to disab	le this channel. See also	description for o	channel 0.
1	CH1ENC	0	W1	Channel 1 Enable Clear
	Write to 1 to disab	le this channel. See also	description for o	channel 0.
0	CH0ENC	0	W1	Channel 0 Enable Clear
	the DMA cycle, or	it reads a channel_cfg i	memory location	sables a channel, by setting the appropriate bit, when either it completes which has cycle_ctrl = b000, or an ERROR occurs on the AHB-Lite bus. a DMA_CHENS register.

8.7.13 DMA_CHALTS - Channel Alternate Set Register

Offset															Bi	t Po	siti	on														
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	1	10	6	∞	7	9	2	4	က	7	-	0
Reset																									0	0	0	0	0	0	0	0
Access																									RW1							
Name																									CH7ALTS	CH6ALTS	CH5ALTS	CH4ALTS	CH3ALTS	CH2ALTS	CH1ALTS	CH0ALTS

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	CH7ALTS	0	RW1	Channel 7 Alternate Structure Set
	Write to 1 to select the	alternate structure	for this channel.	
6	CH6ALTS	0	RW1	Channel 6 Alternate Structure Set
	Write to 1 to select the	alternate structure	for this channel.	
5	CH5ALTS	0	RW1	Channel 5 Alternate Structure Set
	Write to 1 to select the	alternate structure	for this channel.	
4	CH4ALTS	0	RW1	Channel 4 Alternate Structure Set
	Write to 1 to select the	alternate structure	for this channel.	
3	CH3ALTS	0	RW1	Channel 3 Alternate Structure Set
	Write to 1 to select the	alternate structure	for this channel.	
2	CH2ALTS	0	RW1	Channel 2 Alternate Structure Set
	Write to 1 to select the	alternate structure	for this channel.	
1	CH1ALTS	0	RW1	Channel 1 Alternate Structure Set
	Write to 1 to select the	alternate structure	for this channel.	



Bit	Name	Reset	Access	Description
0	CH0ALTS	0	RW1	Channel 0 Alternate Structure Set
	Write to 1 to select the alte	ernate structure for	this channel.	

8.7.14 DMA_CHALTC - Channel Alternate Clear Register

Offset															Bi	t Pc	siti	on														
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	80	7	9	2	4	ю	2	1	0
Reset											•				,			•			•		,		0	0	0	0	0	0	0	0
Access																									W	W1	W	W	W	W	W1	W
Name																									CH7ALTC	CH6ALTC	CH5ALTC	CH4ALTC	CH3ALTC	CH2ALTC	CH1ALTC	CHOALTC

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	CH7ALTC	0	W1	Channel 7 Alternate Clear
	Write to 1 to select	the primary structure for	or this channel.	
6	CH6ALTC	0	W1	Channel 6 Alternate Clear
	Write to 1 to select	the primary structure for	or this channel.	
5	CH5ALTC	0	W1	Channel 5 Alternate Clear
	Write to 1 to select	the primary structure for	or this channel.	
4	CH4ALTC	0	W1	Channel 4 Alternate Clear
	Write to 1 to select	the primary structure for	or this channel.	
3	CH3ALTC	0	W1	Channel 3 Alternate Clear
	Write to 1 to select	the primary structure for	or this channel.	
2	CH2ALTC	0	W1	Channel 2 Alternate Clear
	Write to 1 to select	the primary structure for	or this channel.	
1	CH1ALTC	0	W1	Channel 1 Alternate Clear
	Write to 1 to select	the primary structure for	or this channel.	
0	CH0ALTC	0	W1	Channel 0 Alternate Clear
	Write to 1 to select	the primary structure for	or this channel.	

8.7.15 DMA_CHPRIS - Channel Priority Set Register

Offset															Bi	t Po	siti	on														
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	-	10	6	∞	7	9	2	4	က	2	-	0
Reset											•	•						•							0	0	0	0	0	0	0	0
Access																									RW1							
Name																									CH7PRIS	CH6PRIS	CH5PRIS	CH4PRIS	CH3PRIS	CH2PRIS	CH1PRIS	CHOPRIS

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure com	patibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)



Bit Name Reset Access Description 7 CH7PRIS 0 RW1 Channel 7 High Priority Set Write to 1 to obtain high priority for this channel. Reading returns the channel priority status. 6 CH6PRIS 0 RW1 Channel 6 High Priority Set Write to 1 to obtain high priority for this channel. Reading returns the channel priority status. 5 CH5PRIS 0 RW1 Channel 5 High Priority Set Write to 1 to obtain high priority for this channel. Reading returns the channel priority status. 4 CH4PRIS 0 RW1 Channel 4 High Priority Set						
Write to 1 to obtain high priority for this channel. Reading returns the channel priority status. 6 CH6PRIS 0 RW1 Channel 6 High Priority Set Write to 1 to obtain high priority for this channel. Reading returns the channel priority status. 5 CH5PRIS 0 RW1 Channel 5 High Priority Set Write to 1 to obtain high priority for this channel. Reading returns the channel priority status.	Bit	Name	Reset	Access	Description	
6 CH6PRIS 0 RW1 Channel 6 High Priority Set Write to 1 to obtain high priority for this channel. Reading returns the channel priority status. 5 CH5PRIS 0 RW1 Channel 5 High Priority Set Write to 1 to obtain high priority for this channel. Reading returns the channel priority status.	7	CH7PRIS	0	RW1	Channel 7 High Priority Set	
Write to 1 to obtain high priority for this channel. Reading returns the channel priority status. 5 CH5PRIS 0 RW1 Channel 5 High Priority Set Write to 1 to obtain high priority for this channel. Reading returns the channel priority status.		Write to 1 to obtain h	nigh priority for this cha	annel. Reading re	sturns the channel priority status.	
5 CH5PRIS 0 RW1 Channel 5 High Priority Set Write to 1 to obtain high priority for this channel. Reading returns the channel priority status.	6	CH6PRIS	0	RW1	Channel 6 High Priority Set	
Write to 1 to obtain high priority for this channel. Reading returns the channel priority status.		Write to 1 to obtain h	nigh priority for this cha	annel. Reading re	sturns the channel priority status.	
	5	CH5PRIS	0	RW1	Channel 5 High Priority Set	
4 CH4PRIS 0 RW1 Channel 4 High Priority Set		Write to 1 to obtain h	nigh priority for this cha	annel. Reading re	sturns the channel priority status.	
	4	CH4PRIS	0	RW1	Channel 4 High Priority Set	
Write to 1 to obtain high priority for this channel. Reading returns the channel priority status.		Write to 1 to obtain h	nigh priority for this cha	annel. Reading re	sturns the channel priority status.	
3 CH3PRIS 0 RW1 Channel 3 High Priority Set	3	CH3PRIS	0	RW1	Channel 3 High Priority Set	
Write to 1 to obtain high priority for this channel. Reading returns the channel priority status.		Write to 1 to obtain h	nigh priority for this cha	annel. Reading re	sturns the channel priority status.	
2 CH2PRIS 0 RW1 Channel 2 High Priority Set	2	CH2PRIS	0	RW1	Channel 2 High Priority Set	
Write to 1 to obtain high priority for this channel. Reading returns the channel priority status.		Write to 1 to obtain h	nigh priority for this cha	annel. Reading re	sturns the channel priority status.	
1 CH1PRIS 0 RW1 Channel 1 High Priority Set	1	CH1PRIS	0	RW1	Channel 1 High Priority Set	
Write to 1 to obtain high priority for this channel. Reading returns the channel priority status.		Write to 1 to obtain h	nigh priority for this cha	annel. Reading re	sturns the channel priority status.	
0 CH0PRIS 0 RW1 Channel 0 High Priority Set	0	CH0PRIS	0	RW1	Channel 0 High Priority Set	
Write to 1 to obtain high priority for this channel. Reading returns the channel priority status.		Write to 1 to obtain h	nigh priority for this cha	annel. Reading re	sturns the channel priority status.	

8.7.16 DMA_CHPRIC - Channel Priority Clear Register

Offset															Bi	t Pc	siti	on	·												,	
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	-	10	6	8	7	9	2	4	က	2	-	0
Reset																•					•				0	0	0	0	0	0	0	0
Access																									W1	W N	W1	W V	×	X	W1	W
Name																									CH7PRIC	CH6PRIC	CH5PRIC	CH4PRIC	CH3PRIC	CH2PRIC	CH1PRIC	CHOPRIC

		<u> </u>		
Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	ompatibility with fu	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	CH7PRIC	0	W1	Channel 7 High Priority Clear
	Write to 1 to clear	high priority for this cha	innel.	
6	CH6PRIC	0	W1	Channel 6 High Priority Clear
	Write to 1 to clear	high priority for this cha	innel.	
5	CH5PRIC	0	W1	Channel 5 High Priority Clear
	Write to 1 to clear	high priority for this cha	innel.	
4	CH4PRIC	0	W1	Channel 4 High Priority Clear
	Write to 1 to clear	high priority for this cha	innel.	
3	CH3PRIC	0	W1	Channel 3 High Priority Clear
	Write to 1 to clear	high priority for this cha	innel.	
2	CH2PRIC	0	W1	Channel 2 High Priority Clear
	Write to 1 to clear	high priority for this cha	innel.	
1	CH1PRIC	0	W1	Channel 1 High Priority Clear
	Write to 1 to clear	high priority for this cha	innel.	
0	CH0PRIC	0	W1	Channel 0 High Priority Clear



Bit	Name	Reset	Access	Description							
Write to 1 to clear high priority for this channel.											

8.7.17 DMA_ERRORC - Bus Error Clear Register

Offset															Bi	t Pc	siti	on														
0x04C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	တ	ω	7	9	2	4	ю	2	-	0
Reset				,																												0
Access																																8 ≷
Name																																ERRORC

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure co	ompatibility with f	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	ERRORC	0	RW	Bus Error Clear
				g a 1 to this bit will clear the bit. If the error is deasserted at the same time eccedence and ERRORC remains asserted.

8.7.18 DMA_CHREQSTATUS - Channel Request Status

Offset															Bi	t Pc	siti	on														
0xE10	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	7	10	6	8	7	9	2	4	8	2	-	0
Reset			•								•				,										0	0	0	0	0	0	0	0
Access																									~	2	8	~	œ	œ	~	~
Name																									CH7REQSTATUS	CH6REQSTATUS	CH5REQSTATUS	CH4REQSTATUS	CH3REQSTATUS	CH2REQSTATUS	CH1REQSTATUS	CHOREQSTATUS

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	CH7REQSTATUS	0	R	Channel 7 Request Status
	When this bit is 1, it ind the DMA channel. The	icates that the per controller services	ripheral connected the request by pe	d as the input to this DMA channel is requesting the controller to service erforming the DMA cycle using 2^{R} DMA transfers.
6	CH6REQSTATUS	0	R	Channel 6 Request Status
	When this bit is 1, it ind the DMA channel. The	icates that the per controller services	ripheral connected the request by pe	d as the input to this DMA channel is requesting the controller to service erforming the DMA cycle using 2^{R} DMA transfers.
5	CH5REQSTATUS	0	R	Channel 5 Request Status
				d as the input to this DMA channel is requesting the controller to service erforming the DMA cycle using 2^R DMA transfers.
4	CH4REQSTATUS	0	R	Channel 4 Request Status
				d as the input to this DMA channel is requesting the controller to service erforming the DMA cycle using 2^R DMA transfers.
3	CH3REQSTATUS	0	R	Channel 3 Request Status



Bit	Name	Reset	Access	Description
	When this bit is 1, it indithe DMA channel. The c	cates that the per controller services	ipheral connected the request by per	as the input to this DMA channel is requesting the controller to service rforming the DMA cycle using $2^{\rm R}$ DMA transfers.
2	CH2REQSTATUS	0	R	Channel 2 Request Status
	When this bit is 1, it indithe DMA channel. The c	cates that the per controller services	ipheral connected the request by per	as the input to this DMA channel is requesting the controller to service rforming the DMA cycle using $2^{\rm R}$ DMA transfers.
1	CH1REQSTATUS	0	R	Channel 1 Request Status
	When this bit is 1, it indithe DMA channel. The c	cates that the per controller services	ipheral connected the request by per	as the input to this DMA channel is requesting the controller to service rforming the DMA cycle using $2^{\mbox{R}}$ DMA transfers.
0	CH0REQSTATUS	0	R	Channel 0 Request Status
	When this bit is 1, it indi	cates that the per controller services	ipheral connected the request by per	as the input to this DMA channel is requesting the controller to service rforming the DMA cycle using $2^{\rm R}$ DMA transfers.

8.7.19 DMA_CHSREQSTATUS - Channel Single Request Status

Offset															В	it F	Pos	ition															
0xE18	31	30	29	28	27	26	25	24	23	22	21	20	6	2 8	17	9	<u> </u>	υ <u>4</u>	:	5 5	i 2	_	10	တ	8	7	9	2	4	ю	7	-	0
Reset																										0	0	0	0	0	0	0	0
Access																										2	2	2	2	2	~	~	~
																									_								
Name																										CH7SREQSTATUS	CHESREGSTATUS	CH5SREQSTATUS	CH4SREQSTATUS	CH3SREQSTATUS	CH2SREQSTATUS	CH1SREQSTATUS	CHOSREQSTATUS
Bit	Naı	me						R	eset				Aco	cess	5	E	Des	cript	io	n						U	O	O	O	O	O	O	U
31:8	Res	serve	ed					To	o ens	ure	com	patik	bility	/ witl	h futu	ıre	dev	ices,	al	ways	wri	te k	oits t	o 0.	Мо	re int	orn	natic	n in	Sect	ion 2	2.1 (p	o. 3)
7	CH7	7SR	EQS	STAT	ΓUS			0				F	R			С	har	nel 7	7 8	Single	e Re	equ	ıest	Sta	itus								
											•							input g the l								•		_		ntroll	er to	ser	vice
6	CH6	6SR	EQS	TAT	rus			0				F	R			С	har	nel 6	S S	Single	e Re	equ	ıest	Sta	tus								
																		input the l												ntroll	er to	ser	vice
5	CH	5SR	EQS	TAT	rus			0				F	R			С	har	nel 5	5 S	Single	e Re	equ	ıest	Sta	tus								
											•							input the l								•		_		ntroll	er to	ser	vice
4	CH4	4SR	EQS	TAT	ΓUS			0				F	R			С	har	nel 4	ı s	Single	e Re	equ	ıest	Sta	tus								
																		input the l												ntroll	er to	ser	vice
3	CH	3SR	EQS	STAT	rus			0				F	R			С	har	nel 3	3 S	Single	e Re	equ	ıest	Sta	itus								
																		input the l												ntroll	er to	ser	vice
2	CH2	2SR	EQS	STAT	rus			0				F	R			С	har	nel 2	2 S	Single	e Re	equ	ıest	Sta	itus								
											•							input g the l								•		_		ntroll	er to	ser	vice
1	CH1	1SR	EQS	TAT	rus			0				F	R			С	har	nel 1	S	Single	e Re	equ	ıest	Sta	itus								
																		input g the l												ntroll	er to	ser	vice
0	CHO	OSR	EQS	TAT	rus			0				F	R			С	har	nel 0) S	Single	e Re	equ	ıest	Sta	itus								
	,																				. .												

the DMA channel. The controller services the request by performing the DMA cycle using single DMA transfers.

When this bit is 1, it indicates that the peripheral connected as the input to this DMA channel is requesting the controller to service



8.7.20 DMA_IF - Interrupt Flag Register

Offset															Bi	t Po	siti	on														
0x1000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	တ	8	7	9	2	4	ю	2	-	0
Reset	0		,	•					•						•		•								0	0	0	0	0	0	0	0
Access	~																								~	2	œ	~	~	~	~	~
Name	ERR																								CH7DONE	CH6DONE	CH5DONE	CH4DONE	CH3DONE	CH2DONE	CH1DONE	CHODONE

					H S	Š	Š	S	Ë.	3
Bit	Name	Reset	Access	Description						
31	ERR	0	R	DMA Error Interrupt Flag						
	This flag is set when	an error has occurre	d on the AHB bus							
30:8	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More	e inforn	natio	n in S	Section	on 2.	.1 (p. 3)
7	CH7DONE	0	R	DMA Channel 7 Complete Interrup	t Flag					
	Set when the DMA ch	nannel has completed	lits transfer. If the	channel is disabled, the flag is set when t	here is	a re	quest	t for t	he c	hannel.
6	CH6DONE	0	R	DMA Channel 6 Complete Interrup	t Flag					
	Set when the DMA ch	nannel has completed	lits transfer. If the	channel is disabled, the flag is set when t	here is	a re	quest	t for t	he c	hannel.
5	CH5DONE	0	R	DMA Channel 5 Complete Interrup	t Flag					
	Set when the DMA ch	nannel has completed	lits transfer. If the	channel is disabled, the flag is set when t	here is	a re	quest	t for t	he c	hannel.
4	CH4DONE	0	R	DMA Channel 4 Complete Interrup	t Flag					
	Set when the DMA ch	nannel has completed	lits transfer. If the	channel is disabled, the flag is set when t	here is	a re	quest	t for t	he c	hannel.
3	CH3DONE	0	R	DMA Channel 3 Complete Interrup	t Flag					
	Set when the DMA ch	nannel has completed	lits transfer. If the	channel is disabled, the flag is set when t	here is	a re	quest	t for t	he c	hannel.
2	CH2DONE	0	R	DMA Channel 2 Complete Interrup	t Flag					
	Set when the DMA ch	nannel has completed	lits transfer. If the	channel is disabled, the flag is set when t	here is	a re	quest	t for t	he c	hannel.
1	CH1DONE	0	R	DMA Channel 1 Complete Interrup	t Flag					
	Set when the DMA ch	nannel has completed	lits transfer. If the	channel is disabled, the flag is set when t	here is	a re	quest	t for t	he c	hannel.
0	CH0DONE	0	R	DMA Channel 0 Complete Interrup	t Flag					
	Set when the DMA ch	nannel has completed	lits transfer. If the	channel is disabled, the flag is set when t	here is	a re	quest	t for t	he c	hannel.

8.7.21 DMA_IFS - Interrupt Flag Set Register

Offset															Bi	t Pc	siti	on														
0x1004	31	30	29	28	27	56	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	8	7	9	2	4	ю	7	-	0
Reset	0				•				•			•					•			•			•		0	0	0	0	0	0	0	0
Access	×																								×	W1	×	W	×	×	×	× ×
Name	ERR																								CH7DONE	CH6DONE	CH5DONE	CH4DONE	CH3DONE	CH2DONE	CH1DONE	CHODONE

Bit	Name	Reset	Access	Description
31	ERR	0	W1	DMA Error Interrupt Flag Set
	Set to 1 to set DMA	error interrupt flag.		
30:8	Reserved	To ensure o	compatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	CH7DONE	0	W1	DMA Channel 7 Complete Interrupt Flag Set
	Write to 1 to set the	corresponding DMA	channel complete	interrupt flag.



Bit	Name	Reset	Access	Description
6	CH6DONE	0	W1	DMA Channel 6 Complete Interrupt Flag Set
	Write to 1 to set the	e corresponding DMA o	hannel complete	interrupt flag.
5	CH5DONE	0	W1	DMA Channel 5 Complete Interrupt Flag Set
	Write to 1 to set the	e corresponding DMA o	hannel complete	interrupt flag.
4	CH4DONE	0	W1	DMA Channel 4 Complete Interrupt Flag Set
	Write to 1 to set the	e corresponding DMA o	hannel complete	interrupt flag.
3	CH3DONE	0	W1	DMA Channel 3 Complete Interrupt Flag Set
	Write to 1 to set the	e corresponding DMA o	hannel complete	interrupt flag.
2	CH2DONE	0	W1	DMA Channel 2 Complete Interrupt Flag Set
	Write to 1 to set the	e corresponding DMA o	hannel complete	interrupt flag.
1	CH1DONE	0	W1	DMA Channel 1 Complete Interrupt Flag Set
	Write to 1 to set the	e corresponding DMA o	hannel complete	interrupt flag.
0	CH0DONE	0	W1	DMA Channel 0 Complete Interrupt Flag Set
	Write to 1 to set the	e corresponding DMA o	hannel complete	interrupt flag.

8.7.22 DMA_IFC - Interrupt Flag Clear Register

Offset															Bi	t Pc	siti	on														
0x1008	33	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	တ	ω	7	9	2	4	ю	7	-	0
Reset	0																								0	0	0	0	0	0	0	0
Access	×																								×	W1	×	×	×	×	×	×
Name	ERR																								CH7DONE	CH6DONE	CH5DONE	CH4DONE	CH3DONE	CH2DONE	CH1DONE	CHODONE

Bit	Name	Reset	Access	Description
31	ERR	0	W1	DMA Error Interrupt Flag Clear
0.		-		r happened, the Bus Error Clear Register must be used to clear the DMA.
30:8	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	CH7DONE	0	W1	DMA Channel 7 Complete Interrupt Flag Clear
	Write to 1 to clear t	the corresponding DMA	channel complet	e interrupt flag.
6	CH6DONE	0	W1	DMA Channel 6 Complete Interrupt Flag Clear
	Write to 1 to clear t	the corresponding DMA	channel complet	e interrupt flag.
5	CH5DONE	0	W1	DMA Channel 5 Complete Interrupt Flag Clear
	Write to 1 to clear t	the corresponding DMA	channel complet	e interrupt flag.
4	CH4DONE	0	W1	DMA Channel 4 Complete Interrupt Flag Clear
	Write to 1 to clear t	the corresponding DMA	channel complet	e interrupt flag.
3	CH3DONE	0	W1	DMA Channel 3 Complete Interrupt Flag Clear
	Write to 1 to clear t	the corresponding DMA	channel complet	e interrupt flag.
2	CH2DONE	0	W1	DMA Channel 2 Complete Interrupt Flag Clear
	Write to 1 to clear t	the corresponding DMA	channel complet	e interrupt flag.
1	CH1DONE	0	W1	DMA Channel 1 Complete Interrupt Flag Clear
	Write to 1 to clear t	the corresponding DMA	channel complet	e interrupt flag.
0	CH0DONE	0	W1	DMA Channel 0 Complete Interrupt Flag Clear
	Write to 1 to clear t	the corresponding DMA	channel complet	e interrupt flag.



8.7.23 DMA_IEN - Interrupt Enable register

Offset															Bi	t Po	siti	on														
0x100C	31	30	53	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	æ	7	9	2	4	ю	7	-	0
Reset	0				•		•									•	•		•						0	0	0	0	0	0	0	0
Access	R W																								W.W.	RW	RW	W.	W.	W.W.	RW W	RW
Name	ERR																								CH7DONE	CH6DONE	CH5DONE	CH4DONE	CH3DONE	CH2DONE	CH1DONE	CHODONE

Bit	Name	Reset	Access	Description
31	ERR	0	RW	DMA Error Interrupt Flag Enable
	Set this bit to enabl	le interrupt on AHB bus	error.	
30:8	Reserved	To ensure co.	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	CH7DONE	0	RW	DMA Channel 7 Complete Interrupt Enable
	Write to 1 to enable	e complete interrupt on	this DMA channe	el. Clear to disable the interrupt.
6	CH6DONE	0	RW	DMA Channel 6 Complete Interrupt Enable
	Write to 1 to enable	e complete interrupt on	this DMA channe	el. Clear to disable the interrupt.
5	CH5DONE	0	RW	DMA Channel 5 Complete Interrupt Enable
	Write to 1 to enable	e complete interrupt on	this DMA channe	el. Clear to disable the interrupt.
4	CH4DONE	0	RW	DMA Channel 4 Complete Interrupt Enable
	Write to 1 to enable	e complete interrupt on	this DMA channe	el. Clear to disable the interrupt.
3	CH3DONE	0	RW	DMA Channel 3 Complete Interrupt Enable
	Write to 1 to enable	e complete interrupt on	this DMA channe	el. Clear to disable the interrupt.
2	CH2DONE	0	RW	DMA Channel 2 Complete Interrupt Enable
	Write to 1 to enable	e complete interrupt on	this DMA channe	el. Clear to disable the interrupt.
1	CH1DONE	0	RW	DMA Channel 1 Complete Interrupt Enable
	Write to 1 to enable	e complete interrupt on	this DMA channe	el. Clear to disable the interrupt.
0	CH0DONE	0	RW	DMA Channel 0 Complete Interrupt Enable
	Write to 1 to enable	e complete interrupt on	this DMA channe	el. Clear to disable the interrupt.

8.7.24 DMA_CHx_CTRL - Channel Control Register

Offset															Bi	t Po	siti	on														
0x1100	31	30	29	28	27	26	22	24	23	22	21	20	19	9	17	16	15	41	13	12	7	10	6	80	7	9	2	4	က	2	-	0
Reset													OXO	8																ć) X	
Access										-			8	<u> </u>														-		2	<u>}</u>	
Name													SOURCESE	0000																I C	SIGSEL	

Bit	Name	Reset	Access	Description
31:22	Reserved	To ensure com	patibility with fut	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
21:16	SOURCESEL	0x00	RW	Source Select

Select input source to DMA channel.



Value	Mode	Description
0b000000	NONE	No source selected
0b001000	ADC0	Analog to Digital Converter 0
0b001010	DAC0	Digital to Analog Converter 0
0b001100	USART0	Universal Synchronous/Asynchronous Receiver/Transmitter 0
0b001101	USART1	Universal Synchronous/Asynchronous Receiver/Transmitter 1
0b001110	USART2	Universal Synchronous/Asynchronous Receiver/Transmitter 2
0b010000	LEUART0	Low Energy UART 0
0b010001	LEUART1	Low Energy UART 1
0b010100	I2C0	I2C 0
0b011000	TIMER0	Timer 0
0b011001	TIMER1	Timer 1
0b011010	TIMER2	Timer 2
0b101100	UART0	Universal Asynchronous Receiver/Transmitter 0
0b110000	MSC	
0b110001	AES	Advanced Encryption Standard Accelerator

3:0 SIGSEL To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

Signal Select

Select input signal to DMA channel.

Value	Mode	Description
SOURCESEL = 0b000000 (NONE)		
Obxxxx	OFF	Channel input selection is turned off
SOURCESEL = 0b001000 (ADC0)		
0b0000	ADC0SINGLE	ADCOSINGLE
0b0001	ADC0SCAN	ADCOSCAN
SOURCESEL = 0b001010 (DAC0)		
0b0000	DAC0CH0	DAC0CH0
0b0001	DAC0CH1	DAC0CH1
SOURCESEL = 0b001100 (USART0)		
0b0000	USART0RXDATAV	USART0RXDATAV REQ/SREQ
0b0001	USART0TXBL	USARTOTXBL REQ/SREQ
0b0010	USART0TXEMPTY	USARTOTXEMPTY
SOURCESEL = 0b001101 (USART1)		
0b0000	USART1RXDATAV	USART1RXDATAV REQ/SREQ
0b0001	USART1TXBL	USART1TXBL REQ/SREQ
0b0010	USART1TXEMPTY	USART1TXEMPTY
SOURCESEL = 0b001110 (USART2)		
0b0000	USART2RXDATAV	USART2RXDATAV REQ/SREQ
0b0001	USART2TXBL	USART2TXBL REQ/SREQ
0b0010	USART2TXEMPTY	USART2TXEMPTY
SOURCESEL = 0b010000 (LEUART0)		
0b0000	LEUART0RXDATAV	LEUART0RXDATAV
0b0001	LEUART0TXBL	LEUART0TXBL
0b0010	LEUART0TXEMPTY	LEUART0TXEMPTY
SOURCESEL = 0b010001 (LEUART1)		
0b0000	LEUART1RXDATAV	LEUART1RXDATAV
0b0001	LEUART1TXBL	LEUART1TXBL
0b0010	LEUART1TXEMPTY	LEUART1TXEMPTY
SOURCESEL = 0b010100 (I2C0)		
0b0000	I2C0RXDATAV	I2C0RXDATAV
0b0001	I2C0TXBL	I2C0TXBL

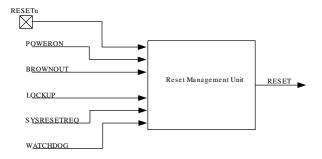


	Name R	eset Access Description	
	Value	Mode	Description
- 1	SOURCESEL = 0b01100 (TIMER0)	0	
Ì	0b0000	TIMER0UFOF	TIMER0UFOF
Ì	0b0001	TIMER0CC0	TIMER0CC0
Ī	0b0010	TIMER0CC1	TIMER0CC1
Ì	0b0011	TIMER0CC2	TIMER0CC2
- 1	SOURCESEL = 0b01100 (TIMER1)	1	
Ì	0b0000	TIMER1UFOF	TIMER1UFOF
Ī	0b0001	TIMER1CC0	TIMER1CC0
İ	0b0010	TIMER1CC1	TIMER1CC1
Ì	0b0011	TIMER1CC2	TIMER1CC2
- 1	SOURCESEL = 0b01101 (TIMER2)	0	
Ī	0b0000	TIMER2UFOF	TIMER2UFOF
İ	0b0001	TIMER2CC0	TIMER2CC0
Ĭ	0b0010	TIMER2CC1	TIMER2CC1
Ĭ	0b0011	TIMER2CC2	TIMER2CC2
ĺ	SOURCESEL = 0b101100 (UARTO	0)	
Ī	0b0000	UART0RXDATAV	UARTORXDATAV REQ/SREQ
İ	0b0001	UART0TXBL	UART0TXBL REQ/SREQ
ĺ	0b0010	UART0TXEMPTY	UART0TXEMPTY
Ì	SOURCESEL = 0b110000 (MSC)		
Ì	0b0000	MSCWDATA	MSCWDATA
Ī	SOURCESEL = 0b110001 (AES)		
Ì	0b0000	AESDATAWR	AESDATAWR
Ī	0b0001	AESXORDATAWR	AESXORDATAWR
Ì	0b0010	AESDATARD	AESDATARD
Ì	0b0011	AESKEYWR	AESKEYWR



9 RMU - Reset Management Unit





Quick Facts

What?

The RMU ensures correct reset operation. It is responsible for connecting the different reset sources to the reset lines of the EFM32G.

Why?

A correct reset sequence is needed to ensure safe and synchronous startup of the EFM32G. In the case of error situations such as power supply glitches or software crash, the RMU provides proper reset and startup of the EFM32G.

How?

The Power-on Reset and Brown-out Detector of the EFM32G provides power line monitoring with exceptionally low power consumption. The cause of the reset may be read from a register, thus providing software with information about the cause of the reset.

9.1 Introduction

The RMU is responsible for handling the reset functionality of the EFM32G.

9.2 Features

- Reset sources
 - Power-on Reset (POR)
 - Brown-out Detection (BOD)
 - RESETn pin reset
 - · Watchdog reset
 - Software triggered reset (SYSRESETREQ)
 - Core LOCKUP condition
- A software readable register indicates the cause of the last reset

9.3 Functional Description

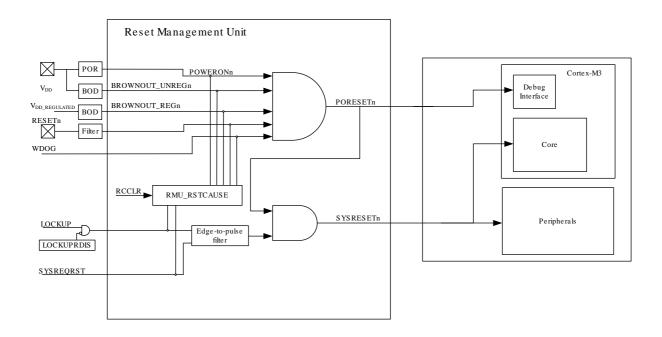
The RMU monitors each of the reset sources of the EFM32G. If one or more reset sources go active, the RMU applies reset to the EFM32G. When the reset sources go inactive the EFM32G starts up. At startup the EFM32G loads the stack pointer and program entry point from memory, and starts execution.

As seen in Figure 9.1 (p. 80) the Power-on Reset, Brown-out Detectors, Watchdog timeout and RESETn pin all reset the whole system including the Debug Interface. A Core Lockup condition or a System reset request from software resets the whole system except the Debug Interface.

Whenever a reset source is active, the corresponding bit in the RMU_RSTCAUSE register is set. At startup the program code may investigate this register in order to determine the cause of the reset. The register must be cleared by software.



Figure 9.1. RMU Reset Input Sources and Connections.



9.3.1 RMU_RSTCAUSE Register

The RMU_RSTCAUSE register indicates the reason for the last reset. The register should be cleared after the value has been read at startup. Otherwise the register may indicate multiple causes for the reset at next startup.

The following procedure must be done to clear RMU_RSTCAUSE:

- 1. Write a 1 to RCCLR in RMU CMD
- 2. Write a 1 to bit 0 in EMU_AUXCTRL
- 3. Write a 0 to bit 0 in EMU_AUXCTRL

RMU_RSTCAUSE should be interpreted according to Table 9.1 (p. 80). X bits are don't care. Notice that it is possible to have multiple reset causes. For example, an external reset and a watchdog reset may happen simultaneously.

Table 9.1. RMU Reset Cause Register Interpretation

Register Value	Cause
0bXXX XXX1	A Power-on Reset has been performed. X bits are don't care.
0b0XX XX10	A Brown-out has been detected on the unregulated power.
0bXX0 0100	A Brown-out has been detected on the regulated power.
0bXXX 1X00	An external reset has been applied.
0bXX1 XX00	A watchdog reset has occurred.
0bX10 0000	A lockup reset has occurred.
0b1X0 0000	A system request reset has occurred.

Note

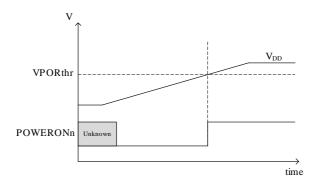
When exiting EM4 with external reset, both the BODREGRST and BODUNREGRST in RSTCAUSE might be set (i.e. are invalid)



9.3.2 Power-On Reset (POR)

The POR ensures that the EFM32G does not start up before the supply voltage V_{DD} has reached the threshold voltage VPORthr (see Device Datasheet Electrical Characteristics for details). Before the threshold voltage is reached, the EFM32G is kept in reset state. The operation of the POR is illustrated in Figure 9.2 (p. 81), with the active low POWERONn reset signal. The reason for the "unknown" region is that the corresponding supply voltage is too low for any reliable operation.

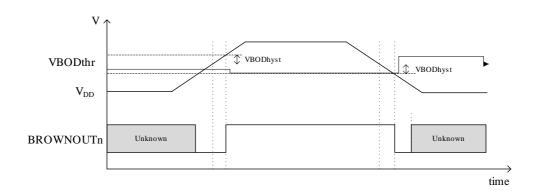
Figure 9.2. RMU Power-on Reset Operation



9.3.3 Brown-Out Detector Reset (BOD)

The EFM32G has 2 brownout detectors, one for the unregulated 3.0 V power and one for the internal 1.8 V power. The BODs are constantly monitoring the voltages. Whenever the voltage is below the VBODthr value (see Electrical Characteristics for details), the corresponding active low BROWNOUTn line is held low. The BODs also include hysteresis, which prevents instability in the corresponding BROWNOUTn line when the supply is crossing the VBODthr limit or the AVDD bods drops below decouple pin (DEC). The operation of the BOD is illustrated in Figure 9.3 (p. 81). The "unknown" regions are handled by the POR module.

Figure 9.3. RMU Brown-out Detector Operation



9.3.4 RESETn pin Reset

Forcing the RESETn pin low generates a reset of the EFM32G. The RESETn pin includes an on-chip pull-up resistor, 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 EFM32G.



9.3.5 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.

9.3.6 Lockup Reset

A Cortex-M3 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.

For more information about the Cortex-M3 lockup conditions see the ARMv7-M Architecture Reference Manual. The Lockup reset does not reset the Debug Interface. Set the LOCKUPRDIS bit in the RMU_CTRL register in order to disable this reset source.

9.3.7 System Reset Request

Software may initiate a reset (e.g. if it finds itself in a non-recoverable state). By writing to the SYSRESETREQ bit in the Application Interrupt and Reset Control Register (see the Cortex-M3 reference manual), a reset is issued. The SYSRESETREQ does not reset the Debug Interface.



9.4 Register Map

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

Offset	Name	Туре	Description
0x000	RMU_CTRL	RW	Control Register
0x004	RMU_RSTCAUSE	R	Reset Cause Register
0x008	RMU_CMD	W1	Command Register

9.5 Register Description

9.5.1 RMU_CTRL - Control Register

Offset															Bi	t Po	siti	on														
0x000	31	30	59	78	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	ი	80	7	9	2	4	က	7	-	0
Reset																					•						•					0
Access																																R
Name																																LOCKUPRDIS

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	LOCKUPRDIS	0	RW	Lockup Reset Disable
	Set this bit to disable th	ne LOCKUP signal	(from the Cortex)	from resetting the device.

9.5.2 RMU_RSTCAUSE - Reset Cause Register

Offset													,		Bi	t Po	siti	on														
0x004	31	30	59	28	27	56	25	24	23	22	21	20	19	18	17	16	15	4	13	12	11	10	6	∞	7	9	2	4	ю	2	-	0
Reset										•		•		•	•			•			•					0	0	0	0	0	0	0
Access																									-	2	~	~	~	~	~	~
Name																										SYSREQRST	LOCKUPRST	WDOGRST	EXTRST	BODREGRST	BODUNREGRST	PORST

Bit	Name	Reset	Access	Description
31:7	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
6	SYSREQRST	0	R	System Request Reset
	Set if a system requ to interpret this bit.	iest reset has been p	erformed. Must be	e cleared by software. Please see Table 9.1 (p. 80) for details on how
5	LOCKUPRST	0	R	LOCKUP Reset
	Set if a LOCKUP resthis bit.	et has been requeste	d. Must be cleared	d by software. Please see Table 9.1 (p. 80) for details on how to interpret
4	WDOGRST	0	R	Watchdog Reset
	Set if a watchdog res	set has been performe	ed. Must be cleared	d by software. Please see Table 9.1 (p. 80) for details on how to interpret



Bit	Name	Reset	Access	Description
3	EXTRST	0	R	External Pin Reset
	Set if an external pin interpret this bit.	reset has been per	formed. Must be o	cleared by software. Please see Table 9.1 (p. 80) for details on how to
2	BODREGRST	0	R	Brown Out Detector Regulated Domain Reset
	Set if a regulated dom for details on how to in		ctor reset has bee	n performed. Must be cleared by software. Please see Table 9.1 (p. 80)
1	BODUNREGRST	0	R	Brown Out Detector Unregulated Domain Reset
	Set if a unregulated do 80) for details on how			been performed. Must be cleared by software. Please see Table 9.1 (p.
0	PORST	0	R	Power On Reset

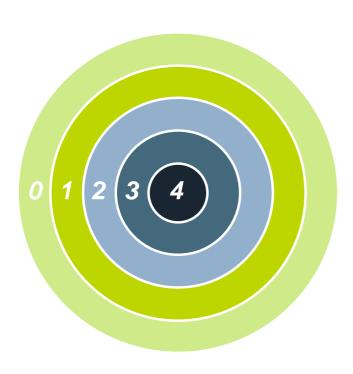
9.5.3 RMU_CMD - Command Register

Offset															Bi	t Po	siti	on														
0x008	31	8	53	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	80	7	9	2	4	က	2	-	0
Reset				•							•	•						•					•	•							•	0
Access																																*
Name																																RCCLR

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	RCCLR	0	W1	Reset Cause Clear
	Set this bit to clear the I EMU_AUXCTRL register t			bits in the RMU_RSTCAUSE register. Use the HRCCLR bit in the



10 EMU - Energy Management Unit



Quick Facts

What?

The EMU (Energy Management Unit) handles the different low energy modes in the EFM32G microcontrollers.

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 (2 µs from EM2 and EM3), applications can dynamically minimize energy consumption during program execution.

10.1 Introduction

The Energy Management Unit (EMU) manages all the low energy modes (EM) in EFM32G microcontrollers. Each energy mode manages if the CPU and the various peripherals are available. The energy modes range from EM0 to EM4, where EM0, also called run mode, enables the CPU and all peripherals. The lowest recoverable energy mode, EM3, disables the CPU and most peripherals while maintaining wake-up and RAM functionality. EM4 disables everything except the POR and pin reset.

The various energy modes differ in:

- · Energy consumption
- CPU activity
- · Reaction time
- Wake-up triggers
- · Active peripherals
- Available clock sources

Low energy modes EM1 to EM4 are enabled through the application software. In EM1-EM3, a range of wake-up triggers return the microcontroller back to EM0. EM4 can only return to EM0 by power on reset or external pin reset.

The EMU can also be used to turn off the power to unused SRAM blocks.

10.2 Features

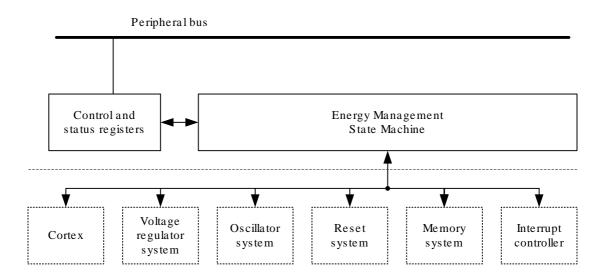
- Energy Mode control from software
- Flexible wakeup from low energy modes
- · Low wakeup time



10.3 Functional Description

The Energy Management Unit (EMU) is responsible for managing the wide range of energy modes available in EFM32G. An overview of the EMU module is shown in Figure 10.1 (p. 86).

Figure 10.1. EMU Overview



The EMU is available as a peripheral on the peripheral bus. The energy management state machine is triggered from the Cortex-M3 and controls the internal voltage regulators, oscillators, memories and interrupt systems in the low energy modes. Events from the interrupt or reset systems can in turn cause the energy management state machine to return to its active state. This is further described in the following sections.

10.3.1 Energy Modes

There are five main energy modes available in EFM32G, called Energy Mode 0 (EM0) through Energy Mode 4 (EM4). EM0, also called the active mode, is the energy mode in which any peripheral function can be enabled and the Cortex-M3 core is executing instructions. EM1 through EM4, also called low energy modes, provide a selection of reduced peripheral functionality that also lead to reduced energy consumption, as described below.

Figure 10.2 (p. 87) shows the transitions between different energy modes. After reset the EMU will always start in EM0. A transition from EM0 to another energy mode is always initiated by software. EM0 is the highest activity mode, in which all functionality is available. EM0 is therefore also the mode with highest energy consumption.

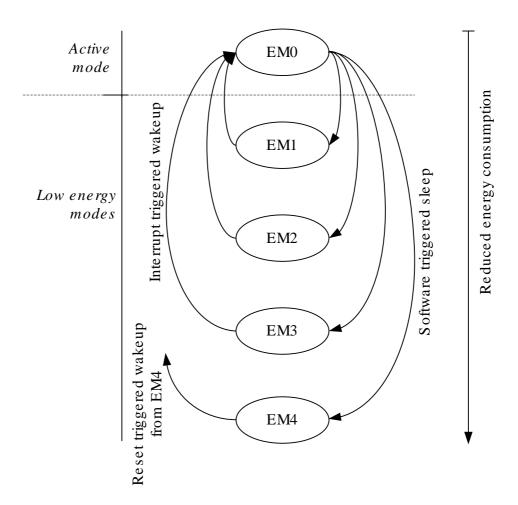
The low energy modes EM1 through EM4 result in less functionality being available, and therefore also reduced energy consumption. The Cortex-M3 is not executing instructions in any low energy mode. Each low energy mode provides different energy consumptions associated with it, for example because a different set of peripherals are enabled or because these peripherals are configured differently.

A transition from EM0 to a low energy mode can only be triggered by software.

A transition from EM1 – EM3 to EM0 can be triggered by an enabled interrupt or event. In addition, a chip reset will return the device to EM0.A transition from EM4 can only be triggered by a pin reset or power-on reset.



Figure 10.2. EMU Energy Mode Transitions



No direct transitions between EM1, EM2 or EM3 are available, as can also be seen from Figure 10.2 (p. 87). Instead, a wakeup will transition back to EM0, in which software can enter any other low energy mode. An overview of the supported energy modes and the functionality available in each mode is shown in Table 10.1 (p. 88). Most peripheral functionality indicated as "On" in a particular energy mode can also be turned off from software in order to save further energy.



Table 10.1. EMU Energy Mode Overview

	EM0 ¹	EM1 ²	EM2 ²	EM3 ²	EM4 ²
Wakeup time to EM0	-	-	2 µs	2 µs	160 µs
MCU clock tree	On	-	-	-	-
High frequency peripheral clock trees	On	On	-	-	-
Core voltage regulator	On	On	-	-	-
High frequency oscillator	On	On	-	-	-
I ² C full functionality	On	On	-	-	-
Low frequency peripheral clock trees	On	On	On	-	-
Low frequency oscillator	On	On	On	-	-
Real Time Counter	On	On	On	-	-
LCD	On	On	On	-	-
LEUART	On	On	On	-	-
LETIMER	On	On	On	-	-
PCNT	On	On	On	On	-
ACMP	On	On	On	On	-
I ² C receive address recognition	On	On	On	On	-
Watchdog	On	On	On	On ³	-
Pin interrupts	On	On	On	On	-
RAM voltage regulator/RAM retention	On	On	On	On	-
Brown Out Reset	On	On	On	On	-
Power On Reset	On	On	On	On	On
Pin Reset	On	On	On	On	On

¹Energy Mode 0/Active Mode

The different Energy Modes are summarized in the following sections.

10.3.1.1 EM0

- The high frequency oscillator is active
- High frequency clock trees are active
- All peripheral functionality is available

10.3.1.2 EM1

- The high frequency oscillator is active
- · MCU clock tree is inactive
- High frequency peripheral clock trees are active
- · All peripheral functionality is available

10.3.1.3 EM2

- The high frequency oscillator is inactive
- The high frequency peripheral and MCU clock trees are inactive

²Energy Mode 1/2/3/4

³When the 1 kHz ULFRCO is selected



- · The low frequency oscillator and clock trees are active
- Low frequency peripheral functionality is available
- Wakeup through peripheral interrupt or asynchronous pin interrupt
- RAM and register values are preserved

10.3.1.4 EM3

- · Both high and low frequency oscillators and clock trees are inactive
- Wakeup through asynchronous pin interrupts, I²C address recognition or ACMP edge interrupt
- Watchdog available when ULFRCO (1 kHz clock) has been selected
- · All other peripheral functionality is disabled
- · RAM and register values are preserved

10.3.1.5 EM4

- · All oscillators and regulators are inactive
- RAM and register values are not preserved
- Wakeup from external pin reset

10.3.2 Entering a Low Energy Mode

A low energy mode is entered by first configuring the desired Energy Mode through the EMU_CTRL register and the SLEEPDEEP bit in the Cortex-M3 System Control Register, see Table 10.2 (p. 89). A Wait For Interrupt (WFI) or Wait For Event (WFE) instruction from the Cortex-M3 triggers the transition into a low energy mode.

The transition into a low energy mode can optionally be delayed until the lowest priority Interrupt Service Routine (ISR) is exited, if the SLEEPONEXIT bit in the Cortex-M3 System Control Register is set.

Entering the lowest energy mode, EM4, is done by writing a sequence to the EM4CTRL bitfield in the EMU_CTRL register. Writing a zero to the EM4CTRL bitfield will restart the power sequence. EM2BLOCK prevents the EMU to enter EM2 or lower, and it will instead enter EM1.

EM3 is equal to EM2, except that the LFACLK/LFBCLK are disabled in EM3. The LFACLK/LFBCLK must be disabled by the user before entering low energy mode.

The EMVREG bit in EMU_CTRL can be used to prevent the voltage regulator from being turned off in low energy modes. The device will then essentially stay in EM1 (with HF oscillators disabled) when entering a low energy mode. Note that if a DMA transfer is initiated in this mode, the HF-oscillators will start and remain enabled until the device is woken up from an EM2 interrupt.

Table 10.2. EMU Entering a Low Energy Mode

Low Energy Mode	EM4CTRL	EMVREG	EM2BLOCK	SLEEPDEEP	Cortex-M3 Instruction
EM1	0	x	x	0	WFI or WFE
EM2	0	0	0	1	WFI or WFE
EM4	Write sequence: 2, 3, 2, 3, 2, 3, 2, 3, 2	х	х	х	х

('x' means don't care)

10.3.3 Leaving a Low Energy Mode

In each low energy mode a selection of peripheral units are available, and software can either enable or disable the functionality. Enabled interrupts that can cause wakeup from a low energy mode are shown



in Table 10.3 (p. 90). The wakeup triggers always return the EFM32 to EM0. Additionally, any reset source will return to EM0.

Table 10.3. EMU Wakeup Triggers from Low Energy Modes

Peripheral	Wakeup Trigger	EM0 ¹	EM1 ²	EM2 ²	EM3 ²	EM4 ²
RTC	Any enabled interrupt	-	Yes	Yes	-	-
USART	Receive / transmit	-	Yes	-	-	-
UART	Receive / transmit	-	Yes	-	-	-
LEUART	Receive / transmit	-	Yes	Yes	-	-
I ² C	Any enabled interrupt	-	Yes	-	-	-
I ² C	Receive address recognition	-	Yes	Yes	Yes	-
TIMER	Any enabled interrupt	-	Yes	-	-	-
LETIMER	Any enabled interrupt	-	Yes	Yes	-	-
СМИ	Any enabled interrupt	-	Yes	-	-	-
DMA	Any enabled interrupt	-	Yes	-	-	-
MSC	Any enabled interrupt	-	Yes	-	-	-
DAC	Any enabled interrupt	-	Yes	-	-	-
ADC	Any enabled interrupt	-	Yes	-	-	-
AES	Any enabled interrupt	-	Yes	-	-	-
PCNT	Any enabled interrupt	-	Yes	Yes	Yes ³	-
LCD	Any enabled interrupt	-	Yes	Yes	-	-
ACMP	Any enabled edge interrupt	-	Yes	Yes	Yes	-
VCMP	Any enabled edge interrupt	-	Yes	Yes	Yes	-
Pin interrupts	Asynchronous	-	Yes	Yes	Yes	-
Pin	Reset	-	Yes	Yes	Yes	Yes
Power	Cycle Off/On		Yes	Yes	Yes	Yes

¹Energy Mode 0/Active Mode

10.3.4 Powering off SRAM blocks

The SRAM blocks can be individually disabled using the POWERDOWN bitfield in the EMU_MEMCTRL register. To disable a block means that the power source is removed from the entire block, which will conserve energy. Once a block has been disabled it can only be enabled by reset.

All the blocks can be turned off except the first one.

²Energy mode 1/2/3/4

³When using an external clock



10.4 Register Map

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

Offset	Name	Туре	Description
0x000	EMU_CTRL	RW	Control Register
0x004	EMU_MEMCTRL	RW	Memory Control Register
0x008	EMU_LOCK	RW	Configuration Lock Register
0x024	EMU_AUXCTRL	RW	Auxiliary Control Register

10.5 Register Description

10.5.1 EMU_CTRL - Control Register

Offset		Bit Position																														
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	ю	2	-	0
Reset																							•	•					2	2	0	0
Access																													Ņ	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	RW	RW
Name																													TANA	5	EM2BLOCK	EMVREG

Bit	Name	Reset	Access	Description
31:4	Reserved	To ensure co	mpatibility with fu	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
3:2	EM4CTRL	0x0	RW	Energy Mode 4 Control
		sed to enter Energy Mode entered when the EM4 s		device only wakes up from an external pin reset or from a power cycle. en to this bitfield.
1	EM2BLOCK	0	RW	Energy Mode 2 Block
	This bit is used to	prevent the MCU to ente	er Energy Mode 2	2 or lower.
0	EMVREG	0	RW	Energy Mode Voltage Regulator Control
	Control the voltag	e regulator in low energy	modes 2 and 3.	
	Value	Mode	Des	scription
	0	REDUCED	Red	duced voltage regulator drive strength in EM2 and EM3.
	1	FULL	Full	voltage regulator drive strength in EM2 and EM3.

10.5.2 EMU_MEMCTRL - Memory Control Register

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	œ	7	9	2	4	3	2	_	0
Reset				•				•									•				-		•	•				•		·	0x0	
Access																RW																
Name																															POWERDOWN	

Bit	Name	Reset	Access	Description
31:3	Reserved	To ensure compa	atibility with futu	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)



Bit	Name	Reset	Access	Description
2:0	POWERDOWN	0x0	RW	RAM block power-down
	up after the reset.	Block 0 (address range 0x2	20000000-0x2	owered down, it cannot be powered up again. The block will be powered 0007FFF) may never be powered down.
	Value	Mode	Des	cription
	4	BLK3	Pow	rer down RAM block 3 (address range 0x20018000-0x2001FFFF).
	6	BLK23	Pow	rer down RAM blocks 2-3 (address range 0x20010000-0x2001FFFF).
	7	BLK123	Pow	rer down RAM blocks 1-3 (address range 0x20008000-0x2001FFFF).

10.5.3 EMU_LOCK - Configuration Lock Register

Offset															Bi	t Po	siti	on														
0x008	33	8	29	28	27	26	25	24	23	22	21	70	19	18	17	16	15	14	13	12	1	10	6	80	7	9	2	4	က	2	-	0
Reset																								000000								
Access																								Ž	<u>}</u>							
Name) 	FOCUME							

15:0	LOCKKEY	0x0000	RW	Configuration Lock Key
31:16	Reserved	To ensure compa	atibility with fu	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
Bit	Name	Reset	Access	Description

Write any other value than the unlock code to lock all EMU registers, except the interrupt registers, from editing. Write the unlock code to unlock. When reading the register, bit 0 is set when the lock is enabled.

Mode	Value	Description
Read Operation		
UNLOCKED	0	EMU registers are unlocked.
LOCKED	1	EMU registers are locked.
Write Operation		
LOCK	0	Lock EMU registers.
UNLOCK	0xADE8	Unlock EMU registers.

10.5.4 EMU_AUXCTRL - Auxiliary Control Register

Offset															Bi	t Pc	siti	on								-						
0x024	31	30	53	28	27	56	52	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	ი	80	7	9	2	4	3	2	-	0
Reset																																0
Access																																R W
Name																																HRCCLR

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure com	patibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	HRCCLR	0	RW	Hard Reset Cause Clear



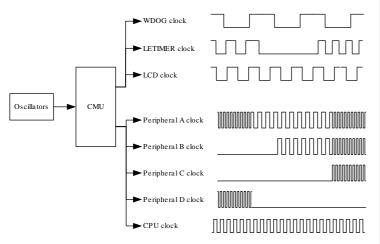
Bit Name Reset Access Description

Write to 1 and then 0 to clear the POR, BOD and WDOG reset cause register bits. See also the Reset Management Unit (RMU).



11 CMU - Clock Management Unit





Quick Facts

What?

The CMU controls oscillators and clocks. EFM32G supports several different oscillators with minimized power consumption and short start-up time. An additional separate RC oscillator is used for flash programming and debug trace. The CMU also 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 can configure different clock sources, enable/disable clocks to peripherals on an individual basis and set the prescaler for the different clocks. The short oscillator start-up times makes duty-cycling between active mode and the different low energy modes (EM2-EM4) very efficient. The calibration feature ensures high accuracy RC oscillators. Several interrupts are available to avoid CPU polling of flags.

11.1 Introduction

The Clock Management Unit (CMU) is responsible for controlling the oscillators and clocks on-board the EFM32G. The CMU provides the capability to turn on and off the clock on an individual basis to all peripheral modules in addition to enable/disable and configure the available oscillators. The high degree of flexibility enables software to minimize energy consumption in any specific application by not wasting power on peripherals and oscillators that are inactive.

11.2 Features

- Multiple clock sources available:
 - 1-28 MHz High Frequency RC Oscillator (HFRCO)
 - 4-32 MHz High Frequency Crystal Oscillator (HFXO)
 - 32768 Hz Low Frequency RC Oscillator (LFRCO)
 - 32768 Hz Low Frequency Crystal Oscillator (LFXO)
 - 1000 Hz Ultra Low Frequency RC Oscillator (ULFRCO)
- Low power oscillators
- · Low start-up times
- Separate prescaler for High Frequency Core Clocks (HFCORECLK) and Peripheral Clocks (HFPERCLK)



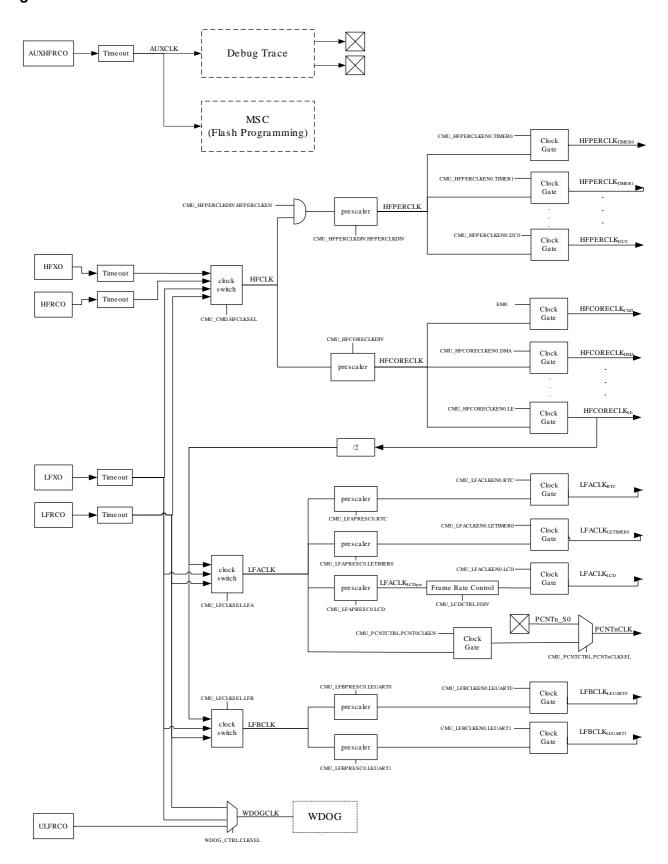
- Individual clock prescaler selection for each Low Energy Peripheral
- Clock Gating on an individual basis to core modules and all peripherals
- Selectable clocks can be output on two pins for use externally.
- Auxiliary 14 MHz RC oscillator (AUXHFRCO) for flash programming, and debug trace.

11.3 Functional Description

An overview of the CMU is shown in Figure 11.1 (p. 96). The number of peripheral modules that are connected to the different clocks varies from device to device.



Figure 11.1. CMU Overview



11.3.1 System Clocks

11.3.1.1 HFCLK - High Frequency Clock

HFCLK is the selected High Frequency Clock. This clock is used by the CMU and drives the two prescalers that generate HFCORECLK and HFPERCLK. The HFCLK can be driven by a high-frequency



oscillator (HFRCO or HFXO) or one of the low-frequency oscillators (LFRCO or LFXO). By default the HFRCO is selected. In most applications, one of the high frequency oscillators will be the preferred choice. To change the selected HFCLK write to HFCLKSEL in CMU_CMD. The HFCLK is running in EM0 and EM1.

11.3.1.2 HFCORECLK - High Frequency Core Clock

HFCORECLK is a prescaled version of HFCLK. This clock drives the Core Modules, which consists of the CPU and modules that are tightly coupled to the CPU, e.g. MSC, DMA etc. This also includes the interface to the Low Energy Peripherals. 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 clock enable bit for the specific module in CMU_HFCORECLKENO. The frequency of HFCORECLK is set using the CMU_HFCORECLKDIV register. The setting can be changed dynamically and the new setting takes effect immediately.

Note

Note that if HFPERCLK runs faster than HFCORECLK, the number of clock cycles for each bus-access to peripheral modules will increase with the ratio between the clocks. Please refer to Section 5.2.3.2 (p. 18) for more details.

11.3.1.3 HFPERCLK - High Frequency Peripheral Clock

Like HFCORECLK, HFPERCLK can also be a prescaled version of HFCLK. This clock drives the High-Frequency Peripherals. All the peripherals that are driven by this clock can be clock gated completely when not in use. This is done by clearing the clock enable bit for the specific peripheral in CMU_HFPERCLKENO. The frequency of HFPERCLK is set using the CMU_HFPERCLKDIV register. The setting can be changed dynamically and the new setting takes effect immediately.

Note

Note that if HFPERCLK runs faster than HFCORECLK, the number of clock cycles for each bus-access to peripheral modules will increase with the ratio between the clocks. E.g. if a bus-access normally takes three cycles, it will take 9 cycles if HFPERCLK runs three times as fast as the HFCORECLK.

11.3.1.4 LFACLK - Low Frequency A Clock

LFACLK is the selected clock for the Low Energy A Peripherals. There are three selectable sources for LFACLK: LFRCO, LFXO and HFCORECLK_{LE}/2. In addition, the LFACLK can be disabled. From reset, the LFACLK source is set to LFRCO. However, note that the LFRCO is disabled from reset. The selection is configured using the LFA field in CMU_LFCLKSEL. The HFCORECLK_{LE}/2 setting allows the Low Energy A Peripherals to be used as high-frequency peripherals.

Note

If HFCORECLK/2 is selected as LFACLK, the clock will stop in EM2/3.

Each Low Energy Peripheral that is clocked by LFACLK has its own prescaler setting and enable bit. The prescaler settings are configured using CMU_LFAPRESC0 and the clock enable bits can be found in CMU_LFACLKEN0. Notice that the LCD has an additional high resolution prescaler for Frame Rate Control, configured by FDIV in CMU_LCDCTRL. When operating in oversampling mode, the pulse counters are clocked by LFACLK. This is configured for each pulse counter (n) individually by setting PCNTnCLKSEL in CMU_PCNTCTRL.

11.3.1.5 LFBCLK - Low Frequency B Clock

LFBCLK is the selected clock for the Low Energy B Peripherals. There are three selectable sources for LFBCLK: LFRCO, LFXO and HFCORECLK_{LE}/2. In addition, the LFBCLK can be disabled. From reset, the LFBCLK source is set to LFRCO. However, note that the LFRCO is disabled from reset. The selection is configured using the LFB field in CMU_LFCLKSEL. The HFCORECLK_{LE}/2 setting allows the Low Energy B Peripherals to be used as high-frequency peripherals.

Note

If HFCORECLK/2 is selected as LFBCLK, the clock will stop in EM2/3.



Each Low Energy Peripheral that is clocked by LFBCLK has its own prescaler setting and enable bit. The prescaler settings are configured using CMU_LFBPRESC0 and the clock enable bits can be found in CMU_LFBCLKEN0.

11.3.1.6 PCNTnCLK - Pulse Counter n Clock

Each available pulse counter is driven by its own clock, PCNTnCLK where n is the pulse counter instance number. Each pulse counter can be configured to use an external pin (PCNTn_S0) or LFACLK as PCNTnCLK.

11.3.1.7 WDOGCLK - Watchdog Timer Clock

The Watchdog Timer (WDOG) can be configured to use one of three different clock sources: LFRCO, LFXO or ULFRCO. ULFRCO (Ultra Low Frequency RC Oscillator) is a separate 1 kHz RC oscillator that also runs in EM3.

11.3.1.8 AUXCLK - Auxiliary Clock

AUXCLK is a 14 MHz clock driven by a separate RC oscillator, AUXHFRCO. This clock is used for flash programming and Serial Wire Output (SWO). During flash programming, this clock will be active. If the AUXHFRCO has not been enabled explicitly by software, the MSC module will automatically start and stop it. The AUXHFRCO is enabled by writing a 1 to AUXHFRCOEN in CMU_OSCENCMD. This explicit enabling is required when SWO is used.

11.3.2 Oscillator Selection

11.3.2.1 Start-up Time

The different oscillators have different start-up times. For the RC oscillators, the start-up time is fixed, but both the LFXO and the HFXO have configurable start-up time. At the end of the start-up time a ready flag is set to indicated that the start-up time has exceeded and that the clock is available. The low start-up time values can be used for an external clock source of already high quality, while the higher start-up times should be used when the clock signal is coming directly from a crystal. The startup time for HFXO and LFXO can be set by configuring the HFXOTIMEOUT and LFXOTIMEOUT bitfields, respectively. Both bitfields are located in CMU_CTRL. For HFXO it is also possible to enable a glitch detection filter by setting HFXOGLITCHDETEN in CMU_CTRL. The glitch detector will reset the start-up counter if a glitch is detected, making the start-up process start over again.

There are individual bits for each oscillator indicating the status of the oscillator:

- ENABLED Indicates that the oscillator is enabled
- READY Start-up time is exceeded
- SELECTED Start-up time is exceeded and oscillator is chosen as clock source

These status bits are located in the CMU_STATUS register.

11.3.2.2 Switching Clock Source

The HFRCO oscillator is a low energy oscillator with extremely short wake-up time. Therefore, this oscillator is always chosen by hardware as the clock source for HFCLK when the device starts up (e.g. after reset and after waking up from EM2 and EM3). After reset, the HFRCO frequency is 14 MHz.

Software can switch between the different clock sources at run-time. E.g., when the HFRCO is the clock source, software can switch to HFXO by writing the field HFCLKSEL in the CMU_CMD command register. See Figure 11.2 (p. 99) for a description of the sequence of events for this specific operation.

Note

It is important first to enable the HFXO since switching to a disabled oscillator will effectively stop HFCLK and only a reset can recover the system.



During the start-up period HFCLK will stop since the oscillator driving it is not ready. This effectively stalls the Core Modules and the High-Frequency Peripherals. It is possible to avoid this by first enabling the HFXO and then wait for the oscillator to become ready before switching the clock source. This way, the system continues to run on the HFRCO until the HFXO has timed out and provides a reliable clock. This sequence of events is shown in Figure 11.3 (p. 99) .

A separate flag is set when the oscillator is ready. This flag can also be configured to generate an interrupt.

Figure 11.2. CMU Switching from HFRCO to HFXO before HFXO is ready

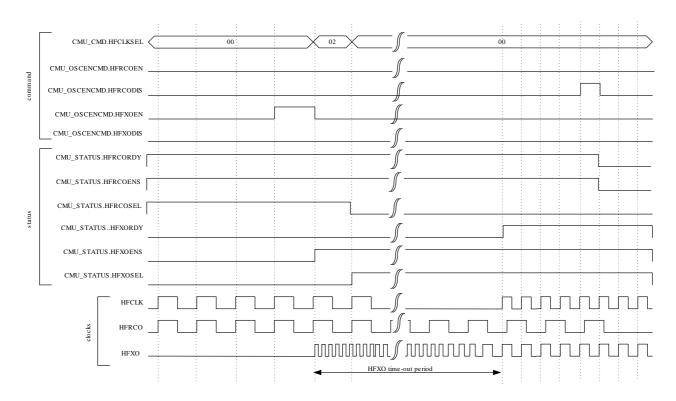
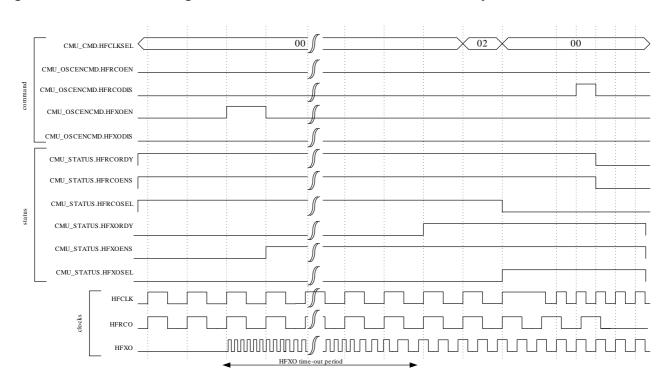


Figure 11.3. CMU Switching from HFRCO to HFXO after HFXO is ready





Switching clock source for LFACLK and LFBCLK is done by setting the LFA and LFB fields in CMU_LFCLKSEL. To ensure no stalls in the Low Energy Peripherals, the clock source should be ready before switching to it.

Note

To save energy, remember to turn off all oscillators not in use.

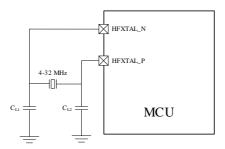
11.3.3 Oscillator Configuration

11.3.3.1 HFXO and LFXO

The crystal oscillators are by default configured to ensure safe startup and operation of the most common crystals. In order to optimize startup margin, startup time and power consumption for a given crystal, it is possible to adjust the gain in the oscillator. HFXO gain can be increased by setting HFXOBOOST field in CMU_CTRL, LFXO gain can be increased by setting LFXOBOOST field in CMU_CTRL. It is important that the boost settings, along with the crystal load capacitors are matched to the crystals in use. Correct values for these parameters can be found using the energyAware Designer.

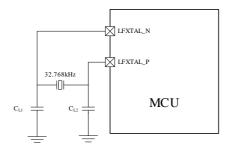
The HFXO crystal is connected to the HFXTAL_N/HFXTAL_P pins as shown in Figure 11.4 (p. 100)

Figure 11.4. HFXO Pin Connection



Similarly, the LFXO crystal is connected to the LFXTAL_N/LFXTAL_P pins as shown in Figure 11.5 (p. 100)

Figure 11.5. LFXO Pin Connection



It is possible to connect an external clock source to HFXTAL_N/LFXTAL_N pin of the HFXO or LFXO oscillator. By configuring the HFXOMODE/LFXOMODE fields in CMU_CTRL, the HFXO/LFXO can be bypassed.

11.3.3.2 HFRCO, LFRCO and AUXHFRCO

The HFRCO can be set to one of several different frequency bands from 1 MHz to 28 MHz by setting the BAND field in CMU_HFRCOCTRL. The HFRCO frequency bands are calibrated during production test, and the production tested calibration values can be read from the Device Information (DI) page. The DI page contains a separate tuning value for each frequency band. During reset HFRCO tuning value is



set to the production calibrated value for the 14 MHz band, which is the default frequency band. When changing to a different HFRCO band, make sure to also update the tuning value.

The LFRCO and AUXHFRCO are also calibrated in production and their TUNING value is set to the correct value during reset.

11.3.3.3 RC oscillator calibration

It is possible to calibrate the HFRCO and LFRCO to achieve higher accuracy (see the device datasheets for details on accuracy). The frequency is adjusted by changing the TUNING fields in CMU_HFRCOCTRL/CMU_LFRCOCTRL. Changing to a higher value will result in a higher frequency. Please refer to the datasheet for stepsize details.

The CMU has built-in HW support to efficiently calibrate the RC oscillators at run-time, see Figure 11.6 (p. 101) The concept is to select a reference and compare the RC frequency with the reference frequency. When the calibration circuit is started, one down-counter running on HFCLK and one up-counter running on a selectable reference clock are started simultaneously. The down-counter counts for CMU_CALCNT +1 cycles. When the down-counter has reached 0, both counters are stopped and software can read out the reference counter value (CALCLK counter) and compare with the start value of the down-counter. Then it is easy to find the ratio between the reference and the oscillator subject to the calibration. With this HW support, it is simple to write efficient calibration algorithms in software.

Figure 11.6. HW-support for RC Oscillator Calibration

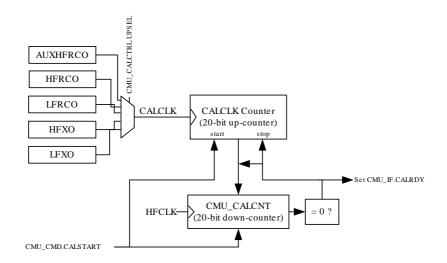
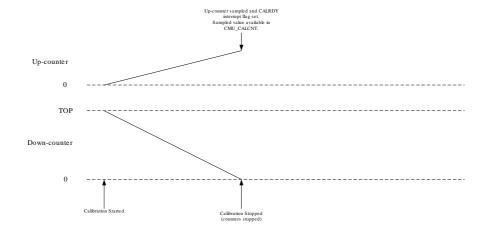


Figure 11.7. Single Calibration (CONT=0)





11.3.4 Output Clock on a Pin

It is possible to configure the CMU to output clocks on two pins. This clock selection is done using CLKOUTSEL0 and CLKOUTSEL1 fields in CMU_CTRL. The output pins must be configured in the CMU_ROUTE register.

- LFRCO or LFXO can be output on one pin (CMU_OUT1)
- HFRCO, HFXO, HFCLK/2, HFCLK/4, HFCLK/8, HFCLK/16 or ULFRCO can be output on another pin (CMU_OUT0)

Note that HFXO and HFRCO clock outputs to pin can be unstable after startup and should not be output on a pin before HFXORDY/HFRCORDY is set high in CMU_STATUS.

11.3.5 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.



11.4 Register Map

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

Offset	Name	Туре	Description
0x000	CMU_CTRL	RW	CMU Control Register
0x004	CMU_HFCORECLKDIV	RW	High Frequency Core Clock Division Register
0x008	CMU_HFPERCLKDIV	RW	High Frequency Peripheral Clock Division Register
0x00C	CMU_HFRCOCTRL	RW	HFRCO Control Register
0x010	CMU_LFRCOCTRL	RW	LFRCO Control Register
0x014	CMU_AUXHFRCOCTRL	RW	AUXHFRCO Control Register
0x018	CMU_CALCTRL	RW	Calibration Control Register
0x01C	CMU_CALCNT	RWH	Calibration Counter Register
0x020	CMU_OSCENCMD	W1	Oscillator Enable/Disable Command Register
0x024	CMU_CMD	W1	Command Register
0x028	CMU_LFCLKSEL	RW	Low Frequency Clock Select Register
0x02C	CMU_STATUS	R	Status Register
0x030	CMU_IF	R	Interrupt Flag Register
0x034	CMU_IFS	W1	Interrupt Flag Set Register
0x038	CMU_IFC	W1	Interrupt Flag Clear Register
0x03C	CMU_IEN	RW	Interrupt Enable Register
0x040	CMU_HFCORECLKEN0	RW	High Frequency Core Clock Enable Register 0
0x044	CMU_HFPERCLKEN0	RW	High Frequency Peripheral Clock Enable Register 0
0x050	CMU_SYNCBUSY	R	Synchronization Busy Register
0x054	CMU_FREEZE	RW	Freeze Register
0x058	CMU_LFACLKEN0	RW	Low Frequency A Clock Enable Register 0 (Async Reg)
0x060	CMU_LFBCLKEN0	RW	Low Frequency B Clock Enable Register 0 (Async Reg)
0x068	CMU_LFAPRESC0	RW	Low Frequency A Prescaler Register 0 (Async Reg)
0x070	CMU_LFBPRESC0	RW	Low Frequency B Prescaler Register 0 (Async Reg)
0x078	CMU_PCNTCTRL	RW	PCNT Control Register
0x07C	CMU_LCDCTRL	RW	LCD Control Register
0x080	CMU_ROUTE	RW	I/O Routing Register
0x084	CMU_LOCK	RW	Configuration Lock Register



11.5 Register Description

11.5.1 CMU_CTRL - CMU Control Register

Offset														Bi	t F	Positi	on										
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	17	9	15	4	13	12	7	9	∞	7	9 2	4	8 2	- 0
Reset									0		0×0		0x3	0				-	OXO		0x3		0	0×1		0x3	0x0
Access									W.		-W W		RW O	Z.			_	Z.	- MA		- KW		M M	W.		WW.	RW (
									ш.				<u> </u>	<u> </u>						•	<u> </u>		LE.			<u> </u>	
Name									CLKOUTSEL1		CLKOUTSEL0		LFXOTIMEOUT	LFXOBUFCUR				LFXOBOOST	FXOMODE	i i	HFXOTIMEOUT		HFXOGLITCHDETEN	HFXOBUFCUR		HFXOBOOST	HFXOMODE
Bit	N	ame						Re	set			A	ccess	5	Ī	Descr	iptio	on									
31:24	R	eserv	red					То	ensi	ure (comp	oatib	ility witl	h futu	ıre	device	es, a	lwa	ys w	rite l	oits to 0.	Mor	e inf	ormatio	n in .	Section 2	.1 (p. 3)
23		LKOL	JTSE	 L1				0					W			lock (. ,
					ck c	outpu	ıt mı		exer	. To	actu			on th			_				in CMU	_RC	UTE	Ε.			
	_	alue				Mode								Descr													
	0	aiuc			_	_FRC									_	(directly	/ from	n os	cillato	or).							
	1				_	_FXC										irectly f											
22:20		LKOL	ITSE	-1.0				0x0)			R	W		_	lock (Jutn	uit (مام؟	ct O							
22.20	Controls the clock output multiplexer. To actually output on the pin, set CLKOUT0PEN in CMU_ROUT									UTI	E.																
	V	alue			ı	Mode							I	Descr	ipti	ion											
	0				ŀ	HFRO	CO						I	HFRC	Ю	(directly	y fron	n os	cillate	or).							
	1				ŀ	HFXC)						ı	HFXO) (d	directly	from	osci	llator).							
	2				_	HFCL								HFCL													
	3				-	HFCL								HFCL													
	4				-	HFCL								HFCL													
	5				-	HFCL								HFCL													
	6					JLFR	CO							ULFR	CC) (direc	tly fro	m o	scilla	itor).							
19:18		XOT onfigi				t-up	dela	0x3 y for		O.		R	W		L	FXO 1	Γime	out									
	V	alue			ı	Mode	!						ı	Descr	ipti	ion											
	0				8	BCYC	LES	;					-	Timeo	out	period	of 8 c	cycle	es.								
	1				•	1KCY	'CLE	S					-	Timeo	out	period	of 10	24 c	ycles	S.							
	2				·	16KC	YCL	ES					-	Timeo	ut	period	of 16	384	cycle	es.							
	3					32KC	YCL	ES					-	Timeo	ut	period	of 32	768	cycle	es.							
17		XOE						0					W			FXO E											
	Tr	nis va	lue l	nas l	peei	n upo	date	d to	the o	corre	ect le	evel	during (calibr	ati	ion an	d sho	oulc	not	be o	changed	-					
16:14	R	eserv	red					То	ensi	ure (comp	oatib	ility witl	h futu	ire	device	es, a	lwa _.	ys w	rite l	oits to 0.	Mor	e inf	ormatio	n in .	Section 2	.1 (p. 3)
13	LF	XOE	300	ST				1				R	W		L	FXO S	Start	-up	Вос	ost (Current						
	Ad	djusts	star	rt-up	boo	ost c	urre	nt fo	r LF	XO.																	
	V	alue			1	Mode	!						ı	Descr	ipti	ion											
	0				-	70PC		•						70 %.	•												
	1				1	100P	CEN	Т						100 %	, o.												
12:11	LF	XON	/IODI	E				0x0)			R	W		L	FXO I	Node)									

Set this to configure the external source for the LFXO. The oscillator setting takes effect when 1 is written to LFXOEN in CMU_OSCENCMD. The oscillator setting is reset to default when 1 is written to LFXODIS in CMU_OSCENCMD.



Bit	Name	Reset	Acce	ess Description
	Value	Mode		Description
	0	XTAL		32.768 kHz crystal oscillator.
	1	BUFEXTCLK		An AC coupled buffer is coupled in series with LFXTAL_N pin, suitable for external sinus wave (32.768 kHz).
	2	DIGEXTCLK		Digital external clock on LFXTAL_N pin. Oscillator is effectively bypassed.
10:9	HFXOTIMEOU	JT 0x3	RW	HFXO Timeout
	Configures the	start-up delay for HFXO.		
	Value	Mode		Description
	0	8CYCLES		Timeout period of 8 cycles.
	1	256CYCLES		Timeout period of 256 cycles.
	2	1KCYCLES		Timeout period of 1024 cycles.
	3	16KCYCLES		Timeout period of 16384 cycles.
8	Reserved	To ensure c	ompatibility v	with future devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	HFXOGLITCH	DETEN 0	RW	HFXO Glitch Detector Enable
0.5	ripple-counter	effectively increasing the s	tart-up time.	long as the start-up ripple-counter is counting. A detected glitch will reset the Once the ripple-counter has timed-out, glitches will not be detected.
6:5	HFXOBUFCUI	-	RW	HFXO Boost Buffer Current
	This value has	been set during calibration	n and should	not be changed.
4	Reserved	To ensure c	ompatibility v	with future devices, always write bits to 0. More information in Section 2.1 (p. 3)
3:2	HFXOBOOST	0x3	RW	HFXO Start-up Boost Current
	Used to adjust	start-up boost current for h	HFXO.	
	Value	Mode		Description
	0	50PCENT		50 %.
	1	70PCENT		70 %.
	2	80PCENT		80 %.
	3	100PCENT		100 % (default).
1:0	HFXOMODE	0x0	RW	HFXO Mode
				IFXO. The oscillator setting takes effect when 1 is written to HFXOEN in default when 1 is written to HFXODIS in CMU_OSCENCMD.
	Value	Mode		Description
	0	XTAL		4-32 MHz crystal oscillator.
	1	BUFEXTCLK		An AC coupled buffer is coupled in series with HFXTAL_N, suitable for external sine wave (4-32 MHz). The sine wave should have a minimum of 200 mV peak to peak.
	2	DIGEXTCLK		Digital external clock on HFXTAL_N pin. Oscillator is effectively bypassed.

11.5.2 CMU_HFCORECLKDIV - High Frequency Core Clock Division Register

Offset															Bi	t Po	siti	on														
0x004	31	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	0	∞	7	9	2	4	ю	7	-	0
Reset								,															•			•	,			2	3	
Access																														///		
Name																														- VIGN DEGO		



	<u> </u>	<u> </u>		
Bit	Name	Reset	Access	s Description
31:4	Reserved	To ensure c	compatibility with	h future devices, always write bits to 0. More information in Section 2.1 (p. 3)
3:0	HFCORECLKDIV	0x0	RW	HFCORECLK Divider
	Specifies the cloc	k divider for HFCOREC	CLK.	
	Value	Mode	1	Description
	0	HFCLK	ŀ	HFCORECLK = HFCLK.
	1	HFCLK2	ŀ	HFCORECLK = HFCLK/2.
	2	HFCLK4	ŀ	HFCORECLK = HFCLK/4.
	3	HFCLK8	ŀ	HFCORECLK = HFCLK/8.
	4	HFCLK16	ŀ	HFCORECLK = HFCLK/16.
	5	HFCLK32	ŀ	HFCORECLK = HFCLK/32.
	6	HFCLK64	ŀ	HFCORECLK = HFCLK/64.
	7	HFCLK128	ŀ	HFCORECLK = HFCLK/128.
	8	HFCLK256	ŀ	HFCORECLK = HFCLK/256.
	9	HFCLK512	ŀ	HFCORECLK = HFCLK/512.

11.5.3 CMU_HFPERCLKDIV - High Frequency Peripheral Clock Division Register

Offset															Bi	t Po	siti	on														
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	8	7	9	2	4	ю	2	1	0
Reset																								-						2	3	
Access																								RW							<u> </u>	
Name																								HFPERCLKEN						NOW ISOLATED		

Bit	Name	Reset	Access	Description
31:9	Reserved	To ensure o	compatibility with f	future devices, always write bits to 0. More information in Section 2.1 (p. 3)
8	HFPERCLKEN	1	RW	HFPERCLK Enable
	Set to enable the	HFPERCLK.		
7:4	Reserved	To ensure o	compatibility with t	future devices, always write bits to 0. More information in Section 2.1 (p. 3)
3:0	HFPERCLKDIV	0x0	RW	HFPERCLK Divider
	Specifies the close	ck divider for the HFPER	RCLK.	
	Value	Mode	De	escription
	0	HFCLK	HF	PERCLK = HFCLK.
	1	HFCLK2	HF	PERCLK = HFCLK/2.
	2	HFCLK4	HF	PERCLK = HFCLK/4.
	3	HFCLK8	HF	PERCLK = HFCLK/8.
	4	HFCLK16	HE	PERCLK = HFCLK/16.
	5	HFCLK32	HF	PERCLK = HFCLK/32.
	6	HFCLK64	HE	PERCLK = HFCLK/64.
	7	HFCLK128	HF	PERCLK = HFCLK/128.
	8	HFCLK256	HF	PERCLK = HFCLK/256.
	9	HFCLK512	HF	PERCLK = HFCLK/512.



7:0

11.5.4 CMU_HFRCOCTRL - HFRCO Control Register

Offset															Bi	t Pc	siti	on														
0x00C	31	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	41	13	12	=	10	6	∞	7	9	2	4	က	2	-	0
Reset																		0x00					0x3						0x80			
Access																		R₩					R W						S ≷			
Name																		SUDELAY					BAND					i	9 NINOL			

Bit	Name	Reset	Access	Description
31:17	Reserved	To ensure comp	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
16:12	SUDELAY	0x00	RW	HFRCO Start-up Delay
	Always write this field to 0.			
11	Reserved	To ensure comp	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
10:8	BAND	0x3	RW	HFRCO Band Select

Write this field to set the frequency band in which the HFRCO is to operate. When changing this setting there will be no glitches on the HFRCO output, hence it is safe to change this setting even while the system is running on the HFRCO. To ensure an accurate frequency, the HFTUNING value should also be written when changing the frequency band. The calibrated tuning value for the different bands can be read from the Device Information page.

Value	Mode	Description
0	1MHZ	1 MHz band. NOTE: Also set the TUNING value (bits 7:0) when changing band.
1	7MHZ	7 MHz band. NOTE: Also set the TUNING value (bits 7:0) when changing band.
2	11MHZ	11 MHz band. NOTE: Also set the TUNING value (bits 7:0) when changing band.
3	14MHZ	14 MHz band. NOTE: Also set the TUNING value (bits 7:0) when changing band.
4	21MHZ	21 MHz band. NOTE: Also set the TUNING value (bits 7:0) when changing band.
5	28MHZ	28 MHz band. NOTE: Also set the TUNING value (bits 7:0) when changing band.

TUNING 0x80 RW **HFRCO Tuning Value**

Writing this field adjusts the HFRCO frequency (the higher value, the higher frequency). This field is updated with the production calibrated value for the 14 MHz band during reset, and the reset value might therefore vary between devices.

11.5.5 CMU_LFRCOCTRL - LFRCO Control Register

Offset	Bit Position						
0x010	1 1 <th>0 2 4 8 7 - 0</th>	0 2 4 8 7 - 0					
Reset		0x40					
Access		- WA					
Name		HUNING H					

Bit	Name	Reset	Access	Description							
31:7	Reserved	To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)									
6:0	TUNING	0x40 RW LFRCO Tuning Value									
	Writing this field adjusts the LFRCO frequency (the higher value, the higher frequency). This field is updated with the production calibrated value during reset, and the reset value might therefore vary between devices.										



11.5.6 CMU_AUXHFRCOCTRL - AUXHFRCO 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	1	10	6	80	7	9	2	4	က	2	-	0
Reset																													0x80			
Access																													S. N			
Name																													DNINOL			

Bit	Name	Reset	Access	Description							
31:8	Reserved	To ensure comp	To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)								
7:0	TUNING	0x80	RW	AUXHFRCO Tuning Value							
	Writing this field adjusts the AUXHFRCO frequency (the higher value, the higher frequency). This field is updated with the production calibrated value during reset, and the reset value might therefore vary between devices.										

11.5.7 CMU_CALCTRL - Calibration Control Register

Offset	Bit Position	
0x018	31	0 1 2
Reset		0×0
Access		R W
Name		UPSEL

Bit	Name	Reset	Access	Description				
31:3	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)				
2:0	UPSEL	0x0	RW	Calibration Up-counter Select				
	Selects clock s	ource for the calibration up	-counter.					
	Value	Mode	Des	cription				
	0	HFXO	Sele	ect HFXO as up-counter.				
	1	LFXO	Sele	ect LFXO as up-counter.				
	2	HFRCO	Select HFRCO as up-counter.					
	3	LFRCO	Sele	ect LFRCO as up-counter.				
	4	AUXHFRCO	Sele	ect AUXHFRCO as up-counter.				



11.5.8 CMU_CALCNT - Calibration Counter Register

Offset														Bit P	ositi	on													
0x01C	31	30	29	28	27	26	27 27	23	22	21	20	6 4	5 5	16	15	4	13	7	10	6	8	7	9	2	4	က	7	-	0
Reset																				0000000									
Access																				RWH									
Name																				CALCNT									

Bit	Name	Reset	Access	Description
31:20	Reserved	To ensure comp	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
19:0	CALCNT	0x00000	RWH	Calibration Counter
	Write top value before calib	oration. Read calibr	ation result fro	m this register when Calibration Ready flag has been set.

11.5.9 CMU_OSCENCMD - Oscillator Enable/Disable Command Register

Offset															Bi	t Po	siti	on														
0x020	31	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	∞	7	9	2	4	ю	7	-	0
Reset																						,	0	0	0	0	0	0	0	0	0	0
Access		-																				-	×	×	×	W1	×	×	×	×	×	W
Name																							LFXODIS	LFXOEN	LFRCODIS	LFRCOEN	AUXHFRCODIS	AUXHFRCOEN	HFXODIS	HFXOEN	HFRCODIS	HFRCOEN

Bit	Name	Reset	Access	Description
31:10	Reserved	To ensure comp	atibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
9	LFXODIS	0	W1	LFXO Disable
	Disables the LFXO. LFXC	EN has higher prior	rity if written si	multaneously.
8	LFXOEN	0	W1	LFXO Enable
	Enables the LFXO.			
7	LFRCODIS	0	W1	LFRCO Disable
	Disables the LFRCO. LFR	RCOEN has higher p	oriority if writter	n simultaneously.
6	LFRCOEN	0	W1	LFRCO Enable
	Enables the LFRCO.			
5	AUXHFRCODIS	0	W1	AUXHFRCO Disable
	Disables the AUXHFRCO a flash erase/write operation		as higher priori	ty if written simultaneously. WARNING: Do not disable this clock during
4	AUXHFRCOEN	0	W1	AUXHFRCO Enable
	Enables the AUXHFRCO.			
3	HFXODIS	0	W1	HFXO Disable



Bit	Name	Reset	Access	Description
	Disables the HFXO is selected as the se		priority if written	simultaneously. WARNING: Do not disable the HFRXO if this oscillator
2	HFXOEN	0	W1	HFXO Enable
	Enables the HFXO.			
1	HFRCODIS	0	W1	HFRCO Disable
	Disables the HFRCO is selected as the se		ner priority if writte	n simultaneously. WARNING: Do not disable the HFRCO if this oscillator
0	HFRCOEN	0	W1	HFRCO Enable
	Enables the HFRCO) .		

11.5.10 CMU_CMD - Command Register

Offset															Bi	t Pc	siti	on														
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	7	10	6	8	7	9	2	4	က	2	-	0
Reset																													0		0x0	
Access																													M		W	
Name																													CALSTART		HFCLKSEL	

Bit	Name	Reset	Access	Description
31:4	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
3	CALSTART	0	W1	Calibration Start
	Starts the calibra	ation, effectively loading the	into the down-counter and start decrementing.	
2:0	HFCLKSEL	0x0	W1	HFCLK Select
2:0	Selects the clock		e that selecting an	oscillator that is disabled will cause the system clock to stop. Check the
2:0	Selects the clock	k source for HFCLK. Note	e that selecting an is ready before sw	oscillator that is disabled will cause the system clock to stop. Check the
2:0	Selects the clock status register a	k source for HFCLK. Note nd confirm that oscillator	e that selecting an is ready before sw	oscillator that is disabled will cause the system clock to stop. Check the itching.
2:0	Selects the clock status register a	k source for HFCLK. Note nd confirm that oscillator	e that selecting an is ready before sw Desc	oscillator that is disabled will cause the system clock to stop. Check the ritching.
2:0	Selects the clock status register a	k source for HFCLK. Note nd confirm that oscillator Mode HFRCO	e that selecting an is ready before sw Desc Sele Sele	oscillator that is disabled will cause the system clock to stop. Check the ritching. cription crit HFRCO as HFCLK.

11.5.11 CMU_LFCLKSEL - Low Frequency Clock Select Register

Offset															Bi	t Pc	siti	on														
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	7	10	6	8	7	9	2	4	က	2	-	0
Reset																													5	5	5	Š
Access																													/\	2	//	<u> </u>
Name																													<u>a</u>	<u>.</u>	<u>د</u>	

Bit	Name	Reset	Access	Description
31:4	Reserved	To ensure compa	atibility with futo	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)



Bit	Name	Reset	Access	Description	
3:2	LFB	0x1	RW	Clock Select for LFB	
	Selects the cloc	ck source for LFBCLK.			
	LFB	LFBE		Mode	Description
	0	0		Disabled	LFBCLK is disabled
	1	0		LFRCO	LFRCO selected as LFBCLK
	2	0		LFXO	LFXO selected as LFBCLK
	3	0		HFCORECLKLEDIV2	HFCORECLK _{LE} divided by two is selected as LFBCLK
	0	1		ULFRCO	ULFRCO selected as LFBCLK
1:0	LFA	0x1	RW	Clock Select for LFA	
	Selects the cloc	ck source for LFACLK.			
	LFA	LFAE		Mode	Description
	0	0		Disabled	LFACLK is disabled
	1	0		LFRCO	LFRCO selected as LFACLK
	2	0		LFXO	LFXO selected as LFACLK
	3	0		HFCORECLKLEDIV2	HFCORECLK _{LE} divided by two is selected as LFACLK
	0	1		ULFRCO	ULFRCO selected as LFACLK

11.5.12 CMU_STATUS - Status Register

Offset															Bi	t Po	siti	on														
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	œ	7	9	2	4	က	2	-	0
Reset				,						•			•					0	0	0	0	-	0	0	0	0	0	0	0	0	-	-
Access																		~	2	~	~	~	2	œ	~	~	8	~	~	~	~	œ
Name																		CALBSY	LFXOSEL	LFRCOSEL	HFXOSEL	HFRCOSEL	LFXORDY	LFXOENS	LFRCORDY	LFRCOENS	AUXHFRCORDY	AUXHFRCOENS	HFXORDY	HFXOENS	HFRCORDY	HFRCOENS

Bit	Name	Reset	Access	Description
31:15	Reserved	To ensure com	npatibility with fu	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
14	CALBSY	0	R	Calibration Busy
	Calibration is on-going.			
13	LFXOSEL	0	R	LFXO Selected
	LFXO is selected as HF0	CLK clock source.		
12	LFRCOSEL	0	R	LFRCO Selected
	LFRCO is selected as H	FCLK clock source.		
11	HFXOSEL	0	R	HFXO Selected
	HFXO is selected as HF	CLK clock source.		
10	HFRCOSEL	1	R	HFRCO Selected
	HFRCO is selected as H	FCLK clock source		
9	LFXORDY	0	R	LFXO Ready
	LFXO is enabled and sta	art-up time has exce	eeded.	
8	LFXOENS	0	R	LFXO Enable Status
	LFXO is enabled.			
7	LFRCORDY	0	R	LFRCO Ready



Bit	Name	Reset	Access	Description
Dit				Description
	LFRCO is enabled and	start-up time has	exceeded.	
6	LFRCOENS	0	R	LFRCO Enable Status
	LFRCO is enabled.			
5	AUXHFRCORDY	0	R	AUXHFRCO Ready
	AUXHFRCO is enabled	d and start-up time	has exceeded.	
4	AUXHFRCOENS	0	R	AUXHFRCO Enable Status
	AUXHFRCO is enabled	d.		
3	HFXORDY	0	R	HFXO Ready
	HFXO is enabled and s	start-up time has ex	ceeded.	
2	HFXOENS	0	R	HFXO Enable Status
	HFXO is enabled.			
1	HFRCORDY	1	R	HFRCO Ready
	HFRCO is enabled and	I start-up time has	exceeded.	
0	HFRCOENS	1	R	HFRCO Enable Status
	HFRCO is enabled.			
	,			

11.5.13 CMU_IF - Interrupt Flag Register

Offset															Bi	t Po	siti	on														
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	∞	7	9	2	4	က	2	-	0
Reset				•					•						•									•			0	0	0	0	0	-
Access																											~	~	~	~	~	~
Name																											CALRDY	AUXHFRCORDY	LFXORDY	LFRCORDY	HFXORDY	HFRCORDY

Bit	Name	Reset	Access	Description
31:6	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
5	CALRDY	0	R	Calibration Ready Interrupt Flag
	Set when calibration is	completed.		
4	AUXHFRCORDY	0	R	AUXHFRCO Ready Interrupt Flag
	Set when AUXHFRCC	is ready (start-up	time exceeded).	
3	LFXORDY	0	R	LFXO Ready Interrupt Flag
	Set when LFXO is rea	dy (start-up time ex	cceeded).	
2	LFRCORDY	0	R	LFRCO Ready Interrupt Flag
	Set when LFRCO is re	ady (start-up time	exceeded).	
1	HFXORDY	0	R	HFXO Ready Interrupt Flag
	Set when HFXO is rea	dy (start-up time ex	xceeded).	
0	HFRCORDY	1	R	HFRCO Ready Interrupt Flag
	Set when HFRCO is re	eady (start-up time	exceeded).	



11.5.14 CMU_IFS - Interrupt Flag Set Register

Offset															Bi	t Po	siti	on														
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	11	10	တ	∞	7	9	2	4	က	2	-	0
Reset				•							-					•					-						0	0	0	0	0	0
Access																											W	W 1	W1	W	W	W1
Name																											CALRDY	AUXHFRCORDY	LFXORDY	LFRCORDY	HFXORDY	HFRCORDY

Bit	Name	Reset	Access	Description
31:6	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
5	CALRDY	0	W1	Calibration Ready Interrupt Flag Set
	Write to 1 to set the Ca	libration Ready(co	mpleted) Interrupt	Flag.
4	AUXHFRCORDY	0	W1	AUXHFRCO Ready Interrupt Flag Set
	Write to 1 to set the AL	JXHFRCO Ready I	nterrupt Flag.	
3	LFXORDY	0	W1	LFXO Ready Interrupt Flag Set
	Write to 1 to set the LF	XO Ready Interrup	t Flag.	
2	LFRCORDY	0	W1	LFRCO Ready Interrupt Flag Set
	Write to 1 to set the LF	RCO Ready Interru	upt Flag.	
1	HFXORDY	0	W1	HFXO Ready Interrupt Flag Set
	Write to 1 to set the HF	XO Ready Interrup	ot Flag.	
0	HFRCORDY	0	W1	HFRCO Ready Interrupt Flag Set
	Write to 1 to set the HF	RCO Ready Interr	upt Flag.	

11.5.15 CMU_IFC - Interrupt Flag Clear Register

Offset															Bi	it Po	ositi	on														
0x038	33	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	0	∞	7	9	2	4	က	7	-	0
Reset																											0	0	0	0	0	0
Access																											W	×	×	M	W	W
Name																											CALRDY	AUXHFRCORDY	LFXORDY	LFRCORDY	HFXORDY	HFRCORDY

Bit	Name	Reset	Access	Description
31:6	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
5	CALRDY	0	W1	Calibration Ready Interrupt Flag Clear
	Write to 1 to clear the 0	Calibration Ready I	nterrupt Flag.	
4	AUXHFRCORDY	0	W1	AUXHFRCO Ready Interrupt Flag Clear
	Write to 1 to clear the	AUXHFRCO Ready	y Interrupt Flag.	
3	LFXORDY	0	W1	LFXO Ready Interrupt Flag Clear
	Write to 1 to clear the L	FXO Ready Interr	upt Flag.	



Bit	Name	Reset	Access	Description
2	LFRCORDY	0	W1	LFRCO Ready Interrupt Flag Clear
	Write to 1 to clear th	e LFRCO Ready Inte	rrupt Flag.	
1	HFXORDY	0	W1	HFXO Ready Interrupt Flag Clear
	Write to 1 to clear th	e HFXO Ready Interr	upt Flag.	
0	HFRCORDY	0	W1	HFRCO Ready Interrupt Flag Clear
	Write to 1 to clear th	e HFRCO Ready Inte	rrupt Flag.	

11.5.16 CMU_IEN - Interrupt Enable Register

Offset															Bi	t Po	siti	on														
0x03C	31	30	59	78	27	26	52	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	80	7	9	2	4	က	7	-	0
Reset																											0	0	0	0	0	0
Access																											RW	ΑW	ΑW	ΑW	N N	W.W.
Name																											CALRDY	AUXHFRCORDY	LFXORDY	LFRCORDY	HFXORDY	HFRCORDY

	Reset	Access	Description
Reserved	To ensure compa	tibility with futu	re devices, always write bits to 0. More information in Section 2.1 (p. 3)
ALRDY	0	RW	Calibration Ready Interrupt Enable
et to enable the Calibration	Ready Interrupt.		
UXHFRCORDY	0	RW	AUXHFRCO Ready Interrupt Enable
et to enable the AUXHFRC	O Ready Interrupt.		
FXORDY	0	RW	LFXO Ready Interrupt Enable
et to enable the LFXO Rea	dy Interrupt.		
FRCORDY	0	RW	LFRCO Ready Interrupt Enable
et to enable the LFRCO Re	eady Interrupt.		
IFXORDY	0	RW	HFXO Ready Interrupt Enable
et to enable the HFXO Rea	dy Interrupt.		
IFRCORDY	0	RW	HFRCO Ready Interrupt Enable
et to enable the HFRCO Re	eady Interrupt.		
F	UXHFRCORDY et to enable the AUXHFRC FXORDY et to enable the LFXO Rea FRCORDY et to enable the LFRCO Re FXORDY et to enable the HFXO Rea FRCORDY	et to enable the AUXHFRCO Ready Interrupt. FXORDY ot to enable the LFXO Ready Interrupt. FRCORDY ot to enable the LFRCO Ready Interrupt. FXORDY ot to enable the HFXO Ready Interrupt.	DIXHFRCORDY 0 RW et to enable the AUXHFRCO Ready Interrupt. FXORDY 0 RW et to enable the LFXO Ready Interrupt. FRCORDY 0 RW et to enable the LFRCO Ready Interrupt. FXORDY 0 RW et to enable the LFRCO Ready Interrupt. FXORDY 0 RW et to enable the HFXO Ready Interrupt. FRCORDY 0 RW

11.5.17 CMU_HFCORECLKEN0 - High Frequency Core Clock Enable Register 0

Offset															Bi	t Pc	siti	on														
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	œ	7	9	2	4	3	2	-	0
Reset					•																•								0	0	0	0
Access																													RW	RW	RW	RW
Name																													EBI	E E	DMA	AES



Bit	Name	Reset	Access	Description
31:4	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
3	EBI	0	RW	External Bus Interface Clock Enable
	Set to enable the o	clock for EBI.		
2	LE	0	RW	Low Energy Peripheral Interface Clock Enable
	Set to enable the o	clock for LE. Interface u	sed for bus acces	s to Low Energy peripherals.
1	DMA	0	RW	Direct Memory Access Controller Clock Enable
	Set to enable the o	clock for DMA.		
0	AES	0	RW	Advanced Encryption Standard Accelerator Clock Enable
	Set to enable the o	clock for AES.		

11.5.18 CMU_HFPERCLKEN0 - High Frequency Peripheral Clock Enable Register 0

Offset															Bi	t Po	siti	on														
0x044	31	30	29	28	27	56	25	24	23	22	21	20	19	18	17	16	15	4	13	12	11	10	0	æ	7	9	2	4	က	2	-	0
Reset									•						•		0	0	0	0	0	0		0	0	0	0	0	0	0	0	0
Access																	R W	RW W	RW	R W	RW W	R W		R W	RW	RW	R W	R W	R W	R W	RW	W.
Name																	12C0	ADC0	VCMP	GPIO	DAC0	PRS		ACMP1	ACMP0	TIMER2	TIMER1	TIMERO	UARTO	USART2	USART1	USARTO

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15	I2C0	0	RW	I2C 0 Clock Enable
	Set to enable the clock to	for I2C0.		
14	ADC0	0	RW	Analog to Digital Converter 0 Clock Enable
	Set to enable the clock to	for ADC0.		
13	VCMP	0	RW	Voltage Comparator Clock Enable
	Set to enable the clock to	for VCMP.		
12	GPIO	0	RW	General purpose Input/Output Clock Enable
	Set to enable the clock t	for GPIO.		
11	DAC0	0	RW	Digital to Analog Converter 0 Clock Enable
	Set to enable the clock t	for DAC0.		
10	PRS	0	RW	Peripheral Reflex System Clock Enable
	Set to enable the clock to	for PRS.		
9	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
8	ACMP1	0	RW	Analog Comparator 1 Clock Enable
	Set to enable the clock t	for ACMP1.		
7	ACMP0	0	RW	Analog Comparator 0 Clock Enable
	Set to enable the clock t	for ACMP0.		
6	TIMER2	0	RW	Timer 2 Clock Enable
	Set to enable the clock t	for TIMER2.		
5	TIMER1	0	RW	Timer 1 Clock Enable
	Set to enable the clock t	for TIMER1.		



		·		
Bit	Name	Reset	Access	Description
4	TIMER0	0	RW	Timer 0 Clock Enable
	Set to enable the	clock for TIMER0.		
3	UART0	0	RW	Universal Asynchronous Receiver/Transmitter 0 Clock Enable
	Set to enable the	clock for UART0.		
2	USART2	0	RW	Universal Synchronous/Asynchronous Receiver/Transmitter 2 Clock Enable
	Set to enable the	clock for USART2.		
1	USART1	0	RW	Universal Synchronous/Asynchronous Receiver/Transmitter 1 Clock Enable
	Set to enable the	clock for USART1.		
0	USART0	0	RW	Universal Synchronous/Asynchronous Receiver/Transmitter 0 Clock Enable
	Set to enable the	clock for USART0.		

11.5.19 CMU_SYNCBUSY - Synchronization Busy Register

Offset															Bi	t Po	siti	on														
0x050	33	30	53	78	27	56	52	24	23	22	21	20	19	18	17	16	15	4	13	12	=	9	0	œ	7	9	2	4	ო	7	-	0
Reset				,				,						,								,		,		0		0		0		0
Access																										~		~		~		~
Name																										LFBPRESC0		LFBCLKEN0		LFAPRESC0		LFACLKENO

Bit	Name	Reset	Access	Description
31:7	Reserved	To ensure co	ompatibility with fu	iture devices, always write bits to 0. More information in Section 2.1 (p. 3)
6	LFBPRESC0	0	R	Low Frequency B Prescaler 0 Busy
	Used to check the s	ynchronization status	of CMU_LFBPRE	ESCO.
	Value	Description		
	1	CMU_LFBPRES	C0 is busy synchron	nizing new value.
5	Reserved	To ensure co	ompatibility with fu	iture devices, always write bits to 0. More information in Section 2.1 (p. 3)
4	LFBCLKEN0	0	R	Low Frequency B Clock Enable 0 Busy
	Used to check the s	ynchronization status	of CMU_LFBCLK	ŒNO.
	Value	Description		
	0	CMU_LFBCLKE	N0 is ready for upda	te.
	1	CMU_LFBCLKE	N0 is busy synchron	izing new value.
3	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	LFAPRESC0	0	R	Low Frequency A Prescaler 0 Busy
	Used to check the s	ynchronization status	of CMU_LFAPRE	ESCO.
	Value	Description		
	0	CMU_LFAPRES	C0 is ready for upda	ate.
	1	CMU_LFAPRES	C0 is busy synchron	nizing new value.
1	Reserved	To ensure co	ompatibility with fu	iture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	LFACLKEN0	0	R	Low Frequency A Clock Enable 0 Busy

Used to check the synchronization status of CMU_LFACLKEN0.



Bit	Name	Reset	Access	Description
	Value	Description		
	0	CMU_LFACLKEN0 is re	eady for update	
	1	CMU_LFACLKEN0 is b	usy synchronizi	ng new value.

11.5.20 CMU_FREEZE - Freeze Register

Offset															Bi	t Po	siti	on														
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	စ	∞	7	9	2	4	ю	2	-	0
Reset																																0
Access																																RW
Name																																REGFREEZE

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	REGFREEZE	0	RW	Register Update Freeze
	When set, the upon registers simultan	•	cy clock control re	gisters is postponed until this bit is cleared. Use this bit to update several
	Value	Mode	Des	cription
	0	UPDATE		h write access to a Low Frequency clock control register is updated into the Low quency domain as soon as possible.
	1	FREEZE	The	LE Clock Control registers are not updated with the new written value.

11.5.21 CMU_LFACLKEN0 - Low Frequency A Clock Enable Register 0 (Async Reg)

Offset					·										Bi	t Pc	siti	on														
0x058	31	30	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	8	7	9	2	4	က	2	-	0
Reset										•					,															0	0	0
Access																														W.	W.	R
Name																														ГСР	LETIMERO	RTC

Bit	Name	Reset	Access	Description
31:3	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	LCD	0	RW	Liquid Crystal Display Controller Clock Enable
	Set to enable the cl	lock for LCD.		
1	LETIMER0	0	RW	Low Energy Timer 0 Clock Enable
	Set to enable the cl	lock for LETIMER0.		
0	RTC	0	RW	Real-Time Counter Clock Enable
	Set to enable the cl	lock for RTC.		



11.5.22 CMU_LFBCLKEN0 - Low Frequency B Clock Enable Register 0 (Async Reg)

Offset															Bi	t Pc	siti	on														
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	80	7	9	2	4	က	2	1	0
Reset																															0	0
Access																															RW	RW
Name																															LEUART1	LEUART0

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	LEUART1	0	RW	Low Energy UART 1 Clock Enable
	Set to enable the c	lock for LEUART1.		
0	LEUART0	0	RW	Low Energy UART 0 Clock Enable
	Set to enable the c	lock for LEUART0.		

11.5.23 CMU_LFAPRESC0 - Low Frequency A Prescaler Register 0 (Async Reg)

Offset															Bi	t Po	siti	on														
0x068	31	30	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	4	13	12	11	10	6	8	7	9	2	4	ю	2	-	0
Reset								•	•						•		•						2	2		-	 OXO			0X0		
Access																							Š				—— ≩ Ƴ			Z N		
Name																							5	3						RTC		

Bit	Name	Reset	Access	Description
31:10	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
9:8	LCD	0x0	RW	Liquid Crystal Display Controller Prescaler
	Configure Liqui	id Crystal Display Controlle	er prescaler	
	Value	Mode	Desc	cription
	0	DIV16	LFA	CLK _{LCD} = LFACLK/16
	1	DIV32	LFA	CLK _{LCD} = LFACLK/32
	2	DIV64	LFA	CLK _{LCD} = LFACLK/64
	3	DIV128	LFA	CLK _{LCD} = LFACLK/128
7:4	LETIMER0	0x0	RW	Low Energy Timer 0 Prescaler

Configure Low Energy Timer 0 prescaler

Value	Mode	Description
0	DIV1	LFACLK _{LETIMERO} = LFACLK
1	DIV2	LFACLK _{LETIMER0} = LFACLK/2
2	DIV4	LFACLK _{LETIMER0} = LFACLK/4
3	DIV8	LFACLK _{LETIMER0} = LFACLK/8
4	DIV16	LFACLK _{LETIMER0} = LFACLK/16
5	DIV32	LFACLK _{LETIMER0} = LFACLK/32
6	DIV64	LFACLK _{LETIMER0} = LFACLK/64



Bit	Name	Reset	Acce	ss Description	
	Value	Mode		Description	
	7	DIV128		LFACLK _{LETIMER0} = LFACLK/128	
	8	DIV256		LFACLK _{LETIMER0} = LFACLK/256	
	9	DIV512		LFACLK _{LETIMER0} = LFACLK/512	
	10	DIV1024		LFACLK _{LETIMER0} = LFACLK/1024	
	11	DIV2048		LFACLK _{LETIMER0} = LFACLK/2048	
	12	DIV4096		LFACLK _{LETIMER0} = LFACLK/4096	
	13	DIV8192		LFACLK _{LETIMER0} = LFACLK/8192	
	14	DIV16384		LFACLK _{LETIMER0} = LFACLK/16384	
	15	DIV32768		LFACLK _{LETIMER0} = LFACLK/32768	
:0	RTC	0x0	RW	Real-Time Counter Prescaler	
	Value	eal-Time Counter prescaler Mode		Description	
	0	DIV1		LFACLK _{RTC} = LFACLK	
	1	DIV2		LFACLK _{RTC} = LFACLK/2	
	2	DIV4		LFACLK _{RTC} = LFACLK/4	
	3	DIV8		LFACLK _{RTC} = LFACLK/8	
	4	DIV16		LFACLK _{RTC} = LFACLK/16	
	5	DIV32		LFACLK _{RTC} = LFACLK/32	
	6	DIV64		LFACLK _{RTC} = LFACLK/64	
	7	DIV128		LFACLK _{RTC} = LFACLK/128	
	8	DIV256		LFACLK _{RTC} = LFACLK/256	
	9	DIV512		LFACLK _{RTC} = LFACLK/512	
	10	DIV1024		LFACLK _{RTC} = LFACLK/1024	
	11	DIV2048		LFACLK _{RTC} = LFACLK/2048	
	12	DIV4096		LFACLK _{RTC} = LFACLK/4096	
	13	DIV8192		LFACLK _{RTC} = LFACLK/8192	
	14	DIV16384		LFACLK _{RTC} = LFACLK/16384	

11.5.24 CMU_LFBPRESC0 - Low Frequency B Prescaler Register 0 (Async Reg)

 $LFACLK_{RTC} = LFACLK/32768$

Offset															Bi	t Po	siti	on														
0x070	31	30	29	28	27	26	22	24	23	22	21	20	19	9	17	16	15	4	13	12	7	10	6	8	7	9	2	4	က	2	-	0
Reset																											0	8			2	2
Access																											8				<u> </u>	2
Name																											I ELIART1				LELIARTO	

Bit	Name	Reset	Access	Description
31:6	Reserved	To ensure c	ompatibility with fu	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
5:4	LEUART1	0x0	RW	Low Energy UART 1 Prescaler
	Configure Low	Energy UART 1 prescaler		
	Value	Mode	Des	scription
	0	DIV1	LFE	BCLK _{LEUART1} = LFBCLK
	1	DIV2	LFE	BCLK _{LEUART1} = LFBCLK/2
	2	DIV4	LFE	BCLK _{LEUART1} = LFBCLK/4
	3	DIV8	LFE	BCLK _{LEUART1} = LFBCLK/8

DIV32768



Bit	Name	Rese	et Acces	ss Description
3:2	Reserved	To en	sure compatibility w	ith future devices, always write bits to 0. More information in Section 2.1 (p. 3)
1:0	LEUART0	0x0	RW	Low Energy UART 0 Prescaler
	Configure Low	Energy UART 0 pre	scaler	
	Value	Mode		Description
	0	DIV1		LFBCLK _{LEUART0} = LFBCLK
	1	DIV2		LFBCLK _{LEUART0} = LFBCLK/2
	2	DIV4		LFBCLK _{LEUART0} = LFBCLK/4
	3	DIV8		LFBCLK _{LEUART0} = LFBCLK/8

11.5.25 CMU_PCNTCTRL - PCNT Control Register

Offset	Bit Position	
0x078	2 2 4 5 6 6 7 8 8 8 7 7 7 8 8 8 8 7 9 9 9 9 9 9 9 9	- 0
Reset	0 0 0 0	0
Access	M M M M M M M M M M M M M M M M M M M	¥ §
Name	PCNT2CLKSEL PCNT2CLKSEL PCNT1CLKSEL PCNT1CLKSEL PCNT1CLKSEL	PCNT0CLKEN

Bit	Name	Re	eset Acc	ess Description
31:6	Reserved	То	ensure compatibility	with future devices, always write bits to 0. More information in Section 2.1 (p. 3)
5	PCNT2CLKSEL	0	RW	PCNT2 Clock Select
	This bit controls v	which clock tha	t is used for the PCN	т.
	Value	Mode		Description
	0	LFACLK		LFACLK is clocking PCNT2.
	1	PCNT2S0		External pin PCNT2_S0 is clocking PCNT0.
4	PCNT2CLKEN	0	RW	PCNT2 Clock Enable
	This bit enables/o	disables the clo	ck to the PCNT.	
	Value	Desci	iption	
	0	PCNT	2 is disabled.	
	1	PCNT	2 is enabled.	
3	PCNT1CLKSEL	0	RW	PCNT1 Clock Select
	This bit controls v	which clock tha	t is used for the PCN	т.
	Value	Mode		Description
	0	LFACLK		LFACLK is clocking PCNT0.
	1	PCNT1S0		External pin PCNT1_S0 is clocking PCNT0.
2	PCNT1CLKEN	0	RW	PCNT1 Clock Enable
	This bit enables/o	disables the clo	ck to the PCNT.	
	Value	Desci	iption	
	0	PCNT	1 is disabled.	
	1	PCNT	1 is enabled.	
1	PCNT0CLKSEL	0	RW	PCNT0 Clock Select

This bit controls which clock that is used for the PCNT.

Value	Mode	Description
0	LFACLK	LFACLK is clocking PCNT0.



Bit	Name	Reset	Acces	s Description					
	Value	Mode		Description					
	1	PCNT0S0		External pin PCNT0_S0 is clocking PCNT0.					
0	PCNT0CLKEN	0	RW	PCNT0 Clock Enable					
	This bit enables/d	lisables the clock to the F	PCNT.						
	Value	Description							
	0	PCNT0 is disable	d.						
	1	PCNT0 is enabled	d.						

11.5.26 CMU_LCDCTRL - LCD Control Register

Offset															Bi	t Po	siti	on														
0x07C	31	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	∞	7	9	2	4	ю	2	-	0
Reset		•							•			•				•	•						•	•			0x2		0		0x0	
Access																											RW		N N		- N	
Name																											VBFDIV		VBOOSTEN		FDIV	

Bit	Name	Reset	Access	Description
31:7	Reserved	To ensure o	compatibility with f	uture devices, always write bits to 0. More information in Section 2.1 (p.
6:4	VBFDIV	0x2	RW	Voltage Boost Frequency Division
	These bits cor	ntrol the voltage boost upda	ate frequency divis	sion.
	Value	Mode	De	scription
	0	DIV1	Vo	Itage Boost update Frequency = LFACLK.
	1	DIV2	Vo	ltage Boost update Frequency = LFACLK/2.
	2	DIV4	Vo	Itage Boost update Frequency = LFACLK/4.
	3	DIV8	Vo	Itage Boost update Frequency = LFACLK/8.
	4	DIV16	Vo	Itage Boost update Frequency = LFACLK/16.
	5	DIV32	Vo	Itage Boost update Frequency = LFACLK/32.
	6	DIV64	Vo	Itage Boost update Frequency = LFACLK/64.

3 VBOOSTEN 0 RW **Voltage Boost Enable**This bit enables/disables the VBOOST function.

These bits controls the framerate according to this formula: $LFACLK_{LCD} = LFACLK_{LCDpre} / (1 + FDIV)$. Do not change this value while the LCD bit in CMU_LFACLKEN0 is set to 1.

Frame Rate Control

11.5.27 CMU_ROUTE - I/O Routing Register

Offset															Bi	t Po	siti	on														
0x080	31	30	59	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	0	8	7	9	2	4	က	7	-	0
Reset					•			•								•											•		-	0	0	0
Access																														8 ⊗	W.	₩ W
Name																														LOCATION	CLKOUT1PEN	CLKOUTOPEN

2:0

FDIV



Bit	Name	Reset	Access	Description
31:3	Reserved	To ensure c	ompatibility with	n future devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	LOCATION	0	RW	I/O Location
	Decides the locat	tion of the CMU I/O pins.		
	Value	Mode		Description
	0	LOC0	L	ocation 0
	1	LOC1	L	ocation 1
1	CLKOUT1PEN	0	RW	CLKOUT1 Pin Enable
	When set, the CL	KOUT1 pin is enabled.		
0	CLKOUT0PEN	0	RW	CLKOUT0 Pin Enable
	When set, the CL	KOUT0 pin is enabled.		

11.5.28 CMU_LOCK - Configuration Lock Register

Offset					•			,							Bi	t Pc	siti	on														,
0x084	33	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	19	6	∞	7	9	2	4	ю	2	-	0
Reset																									000000							
Access																								Š	<u>}</u>							
Name) 	FOCUME							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)

15:0 LOCKKEY 0x0000 RW **Configuration Lock Key**

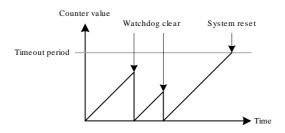
CMU_CTRL, CMU_HFCORECLKDIV, Write any other value than the unlock code to lock CMU_HFPERCLKDIV, CMU_HFRCOCTRL, CMU_LFRCOCTRL, CMU_AUXHFRCOCTRL, CMU_OSCENCMD, CMU_CMD, CMU_LFCLKSEL, CMU_HFCORECLKENO, CMU_HFPERCLKENO, CMU_LFACLKENO, CMU_LFBCLKENO, CMU_LFAPRESCO, CMU_LFBPRESC0, and CMU_PCNTCTRL from editing. Write the unlock code to unlock. When reading the register, bit 0 is set when the lock is enabled.

Mode	Value	Description
Read Operation		
UNLOCKED	0	CMU registers are unlocked.
LOCKED	1	CMU registers are locked.
Write Operation		
LOCK	0	Lock CMU registers.
UNLOCK	0x580E	Unlock CMU registers.



12 WDOG - Watchdog 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.

12.1 Introduction

The purpose of the watchdog timer is to generate a reset in case of a system failure, to increase application reliability. The failure may e.g. be caused by an external event, such as an ESD pulse, or by a software failure.

12.2 Features

- · Clock input from selectable oscillators
 - Internal 32.768 Hz RC oscillator
 - Internal 1 kHz RC oscillator
 - External 32.768 Hz XTAL oscillator
- Configurable timeout period from 9 to 256k watchdog clock cycles
- Individual selection to keep running or freeze when entering EM2 or EM3
- Selection to keep running or freeze when entering debug mode
- Selection to block the CPU from entering Energy Mode 4
- Selection to block the CMU from disabling the selected watchdog clock

12.3 Functional Description

The watchdog is enabled by setting the EN bit in WDOG_CTRL. When enabled, the watchdog counts up to the period value configured through the PERSEL field in WDOG_CTRL. If the watchdog timer is not cleared to 0 (by writing a 1 to the CLEAR bit in WDOG_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 the LOCK bit in WDOG_CTRL. Once locked, it cannot be disabled or reconfigured by software.

The watchdog counter is reset when EN is reset.



12.3.1 Clock Source

Three clock sources are available for use with the watchdog, through the CLKSEL field in WDOG_CTRL. The corresponding clocks must be enabled in the CMU. The SWOSCBLOCK bit in WDOG_CTRL can be written to prevent accidental disabling of the selected clocks. Also, setting this bit will automatically start the selected oscillator source when the watchdog is enabled. The PERSEL field in WDOG_CTRL is used to divide the selected watchdog clock, and the timeout for the watchdog timer can be calculated like this:

WDOG Timeout Equation

$$T_{TIMEOUT} = (2^{3+PERSEL} + 1)/f,$$
 (12.1)

where f is the frequency of the selected clock.

It is recommended to clear the watchdog first, if PERSEL is changed while the watchdog is enabled.

To use this module, the LE interface clock must be enabled in CMU_HFCORECLKEN0, in addition to the module clock.

Note

Before changing the clock source for WDOG, the EN bit in WDOG_CTRL should be cleared. In addition to this, the WDOG_SYNCBUSY value should be zero.

12.3.2 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 WDOG_CTRL. When code execution is resumed, the watchdog will continue counting where it left off.

12.3.3 Energy Mode Handling

The watchdog timer can be configured to either keep on running or freeze when entering EM2 or EM3. The configuration is done individually for each energy mode in the EM2RUN and EM3RUN bits in WDOG_CTRL. 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. For the watchdog there is no difference between EM0 and EM1. The watchdog does not run in EM4, and if EM4BLOCK in WDOG_CTRL is set, the CPU is prevented from entering EM4.

Note

If the WDOG is clocked by the LFXO or LFRCO, writing the SWOSCBLOCK bit will effectively prevent the CPU from entering EM3. When running from the ULFRCO, writing the SWOSCBLOCK bit will prevent the CPU from entering EM4.

12.3.4 Register access

Since this module is a Low Energy Peripheral, and runs off a clock which is asynchronous to the HFCORECLK, special considerations must be taken when accessing registers. Please refer to Section 5.3 (p. 19) for a description on how to perform register accesses to Low Energy Peripherals. note that clearing the EN bit in WDOG_CTRL will reset the WDOG module, which will halt any ongoing register synchronization.

Note

Never write to the WDOG registers when it is disabled, except to enable it by setting WDOG_CTRL_EN or when changing the clock source using WDOG_CTRL_CLKSEL. Make sure that the enable is registered (i.e. WDOG_SYNCBUSY_CTRL goes low), before writing other registers.



12.4 Register Map

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

Offset	Name	Туре	Description
0x000	WDOG_CTRL	RW	Control Register
0x004	WDOG_CMD	W1	Command Register
0x008	WDOG_SYNCBUSY	R	Synchronization Busy Register

12.5 Register Description

12.5.1 WDOG_CTRL - Control Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

Offset															Bi	t Po	siti	on														
0x000	31	30	59	28	27	26	22	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	80	7	9	2	4	က	2	-	0
Reset				•						•	•	•	•					•		0×0		L 2	L X			0	0	0	0	0	0	0
Access																				 ≷			≥ Y			RW	RW	RW W	R ⊗	RW	RW	RW
Name																				CLKSEL		- I	PERSEL			SWOSCBLOCK	EM4BLOCK	LOCK	EM3RUN	EM2RUN	DEBUGRUN	EN

Bit	Name	Reset	Access	Description
31:14	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
13:12	CLKSEL	0x0	RW	Watchdog Clock Select
	Selects the W	DOG oscillator, i.e. the cloc	k on which the wa	tchdog will run.
	Value	Mode	Des	cription
	0	ULFRCO	ULF	RCO
	1	LFRCO	LFR	CO
		LFXO	LFX	

11:8 PERSEL 0xF RW Watchdog Timeout Period Select

Select watchdog timeout period.

Value	Description
0	Timeout period of 9 watchdog clock cycles.
1	Timeout period of 17 watchdog clock cycles.
2	Timeout period of 33 watchdog clock cycles.
3	Timeout period of 65 watchdog clock cycles.
4	Timeout period of 129 watchdog clock cycles.
5	Timeout period of 257 watchdog clock cycles.
6	Timeout period of 513 watchdog clock cycles.
7	Timeout period of 1k watchdog clock cycles.
8	Timeout period of 2k watchdog clock cycles.
9	Timeout period of 4k watchdog clock cycles.
10	Timeout period of 8k watchdog clock cycles.
11	Timeout period of 16k watchdog clock cycles.
12	Timeout period of 32k watchdog clock cycles.
13	Timeout period of 64k watchdog clock cycles.
14	Timeout period of 128k watchdog clock cycles.
15	Timeout period of 256k watchdog clock cycles.



		<u> </u>		
Bit	Name	Reset	Access	Description
7	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
6	SWOSCBLOCK	0	RW	Software Oscillator Disable Block
	Set to disallow disal already running.	oling of the selected \	VDOG oscillator.	Writing this bit to 1 will turn on the selected WDOG oscillator if it is not
	Value	Description		
	0	Software is allow registers are lock		elected WDOG oscillator. See CMU for detailed description. Note that also CMU
	1	Software is not a	llowed to disable the	e selected WDOG oscillator.
5	EM4BLOCK	0	RW	Energy Mode 4 Block
	Set to prevent the El	MU from entering EM	4.	
	Value	Description		
	0	EM4 can be ente	red. See EMU for de	etailed description.
	1	EM4 cannot be e	ntered.	
4	LOCK	0	RW	Configuration lock
	Set to lock the watch	ndog configuration. Th	nis bit can only be	cleared by reset.
	Value	Description		·
	0	· · ·	uration can be chan	ned
	1		uration cannot be ch	
3	EM3RUN	0	RW	Energy Mode 3 Run Enable
J	Set to keep watchdo	-	1777	Energy Mode & Null Enable
	Value	Description		
	0	Watchdog timer i	s frozen in EM3.	
	1	Watchdog timer i	s running in EM3.	
2	EM2RUN	0	RW	Energy Mode 2 Run Enable
	Set to keep watchdo	g running in EM2.		3,
	Value	Description		
	0	Watchdog timer i	s frozen in EM2.	
	1	Watchdog timer i	s running in EM2.	
1	DEBUGRUN	0	RW	Debug Mode Run Enable
	Set to keep watchdo	g running in debug m	ode.	
	Value	Description		
	0	Watchdog timer i	s frozen in debug m	ode.
	1	Watchdog timer i	s running in debug r	mode.
0	EN	0	RW	Watchdog Timer Enable
	Set to enabled watch	ndog timer.		-
	Co. to chabica water			

12.5.2 WDOG_CMD - Command Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19) .

Offset															Bi	t Pc	siti	on														
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	∞	7	9	2	4	е	2	-	0
Reset			•	•	•											•	•				•		•		•							0
Access																																W
Name																																CLEAR



Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure c	compatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	CLEAR	0	W1	Watchdog Timer Clear
	Clear watchdo	g timer. The bit must be wi	ritten 4 watchdog c	cycles before the timeout.
	Clear watchdo	g timer. The bit must be wi		cycles before the timeout.
			Des	,

12.5.3 WDOG_SYNCBUSY - Synchronization Busy Register

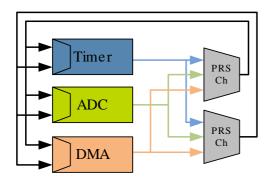
Offset															Bi	t Po	siti	on														
0x008	31	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	စ	∞	7	9	2	4	က	2	1	0
Reset																															0	0
Access																															Ж	~
Name																															CMD	CTRL

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	CMD	0	R	CMD Register Busy
	Set when the value	e written to CMD is beir	ng synchronized.	
0	CTRL	0	R	CTRL Register Busy
	Set when the value	e written to CTRL is bei	ng synchronized.	



13 PRS - Peripheral Reflex System





Quick Facts

What?

The PRS (Peripheral Reflex System) allows configurable, fast and autonomous communication between the peripherals.

Why?

Events and signals from one peripheral can be used as input signals or triggers by other peripherals and ensure timing-critical operation and reduced software overhead.

How?

Without CPU intervention the peripherals can send reflex signals (both pulses and level) 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.

13.1 Introduction

The Peripheral Reflex System (PRS) system is a network which allows the different peripheral modules to communicate directly with each other without involving the CPU. Peripheral modules which send out reflex signals are called producers. The PRS routes these reflex signals to consumer peripherals which apply actions depending on the reflex signals received. The format for the reflex signals is not given, but edge triggers and other functionality can be applied by the PRS.

13.2 Features

- 8 configurable interconnect channels
 - Each channel can be connected to any producing peripheral
 - Consumers can choose which channel to listen to
 - Selectable edge detector (rising, falling and both edges)
- Software controlled channel output
 - · Configurable level
 - Triggered pulses

13.3 Functional Description

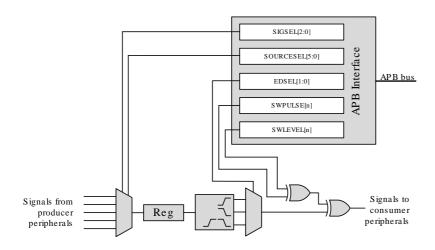
An overview of the PRS module is shown in Figure 13.1 (p. 129). The PRS contains 8 interconnect channels, and each of these can select between all the output reflex signals offered by the producers. The consumers can then choose which PRS channel to listen to and perform actions based on the reflex signals routed through that channel. The reflex signals can be both pulse signals and level signals. Synchronous PRS pulses are one HFPERCLK cycle long, and can either be sent out by a producer (e.g., ADC conversion complete) or be generated from the edge detector in the PRS channel. Level signals can have an arbitrary waveform (e.g., Timer PWM output).



13.3.1 Channel Functions

Different functions can be applied to a reflex signal within the PRS. Each channel includes an edge detector to enable generation of pulse signals from level signals. It is also possible to generate output reflex signals by configuring the SWPULSE and SWLEVEL bits. SWLEVEL is a programmable level for each channel and holds the value it is programmed to. The SWPULSE will give out a one-cycle high pulse if it is written to 1, otherwise a 0 is asserted. The SWLEVEL and SWPULSE signals are then XOR'ed with the selected input from the producers to form the output signal sent to the consumers listening to the channel.

Figure 13.1. PRS Overview



13.3.2 Producers

Each PRS channel can choose between signals from several producers, which is configured in SOURCESEL in PRS_CHx_CTRL. Each of these producers outputs one or more signals which can be selected by setting the SIGSEL field in PRS_CHx_CTRL. Setting the SOURCESEL bits to 0 (Off) leads to a constant 0 output from the input mux. An overview of the available producers is given in Table 13.1 (p. 129) .

Table 13.1. Reflex Producers

Module	Reflex Output	Output Format
ACMP	Comparator Output	Level
ADC	Single Conversion Done	Pulse
	Scan Conversion Done	Pulse
DAC	Channel 0 Conversion Done	Pulse
	Channel 0 Conversion Done	Pulse
GPIO	Pin 0 Input	Level
	Pin 1 Input	Level
	Pin 2 Input	Level
	Pin 3 Input	Level
	Pin 4 Input	Level
	Pin 5 Input	Level



Module	Reflex Output	Output Format
	Pin 6 Input	Level
	Pin 7 Input	Level
	Pin 8 Input	Level
	Pin 9 Input	Level
	Pin 10 Input	Level
	Pin 11 Input	Level
	Pin 12 Input	Level
	Pin 13 Input	Level
	Pin 14 Input	Level
	Pin 15 Input	Level
RTC	Overflow	Pulse
	Compare Match 0	Pulse
	Compare Match 1	Pulse
TIMER	Underflow	Pulse
	Overflow	Pulse
	CC0 Output	Level
	CC1 Output	Level
	CC2 Output	Level
UART	TX Complete	Pulse
	RX Data Received	Pulse
USART	TX Complete	Pulse
	RX Data Received	Pulse
	IrDA Decoder Output	Level
VCMP	Comparator Output	Level

13.3.3 Consumers

Consumer peripherals (listed in Table 13.2 (p. 130)) can be set to listen to a PRS channel and perform an action based on the signal received on that channel. Most consumers expect pulse input, while some can handle level inputs as well.

Table 13.2. Reflex Consumers

Module	Reflex Input	Input Format
ADC	Single Mode Trigger	Pulse
	Scan Mode Trigger	Pulse
DAC	Channel 0 Trigger	Pulse
	Channel 1 Trigger	Pulse
TIMER	CC0 Input	Pulse/Level



Module	Reflex Input	Input Format
	CC1 Input	Pulse/Level
	CC2 Input	Pulse/Level
	DTI Fault Source 0 (TIMER0 only)	Pulse
	DTI Fault Source 1 (TIMER0 only)	Pulse
	DTI Input (TIMER0 only)	Pulse/Level
UART	TX/RX Enable	Pulse
USART	TX/RX Enable	Pulse
	IrDA Encoder Input (USART0 only)	Level

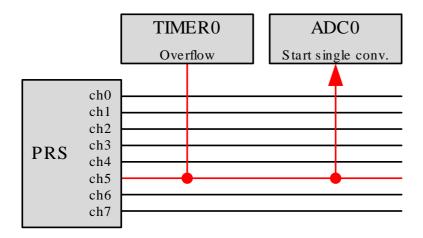
13.3.4 Example

The example below (illustrated in Figure 13.2 (p. 131)) shows how to set up ADC0 to start single conversions every time TIMER0 overflows (one HFPERCLK cycle high pulse), using PRS channel 5:

- Set SOURCESEL in PRS_CH5_CTRL to 0b011100 to select TIMER0 as input to PRS channel 5.
- Set SIGSEL in PRS_CH5_CTRL to 0b001 to select the overflow signal (from TIMER0).
- Configure ADC0 with the desired conversion set-up.
- Set SINGLEPRSEN in ADC0_SINGLECTRL to 1 to enable single conversions to be started by a high PRS input signal.
- Set SINGLEPRSSEL in ADC0_SINGLECTRL to 0x5 to select PRS channel 5 as input to start the single conversion.
- Start TIMER0 with the desired TOP value, an overflow PRS signal is output automatically on overflow.

Note that the ADC results needs to be fetched either by the CPU or DMA.

Figure 13.2. TIMER0 overflow starting ADC0 single conversions through PRS channel 5.





13.4 Register Map

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

Offset	Name	Туре	Description
0x000	PRS_SWPULSE	W1	Software Pulse Register
0x004	PRS_SWLEVEL	RW	Software Level Register
0x010	PRS_CH0_CTRL	RW	Channel Control Register
0x014	PRS_CH1_CTRL	RW	Channel Control Register
0x018	PRS_CH2_CTRL	RW	Channel Control Register
0x01C	PRS_CH3_CTRL	RW	Channel Control Register
0x020	PRS_CH4_CTRL	RW	Channel Control Register
0x024	PRS_CH5_CTRL	RW	Channel Control Register
0x028	PRS_CH6_CTRL	RW	Channel Control Register
0x02C	PRS_CH7_CTRL	RW	Channel Control Register

13.5 Register Description

13.5.1 PRS_SWPULSE - Software Pulse Register

Offset															Bi	t Po	siti	on														
0x000	31	30	29	28	27	56	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	æ	7	9	2	4	8	2	-	0
Reset		,	•	,						•	•		•										,		0	0	0	0	0	0	0	0
Access																									W	W1	W1	W1	W1	W1	W	W1
Name																									CH7PULSE	CH6PULSE	CH5PULSE	CH4PULSE	CH3PULSE	CH2PULSE	CH1PULSE	CHOPULSE

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	CH7PULSE	0	W1	Channel 7 Pulse Generation
	See bit 0.			
6	CH6PULSE	0	W1	Channel 6 Pulse Generation
	See bit 0.			
5	CH5PULSE	0	W1	Channel 5 Pulse Generation
	See bit 0.			
4	CH4PULSE	0	W1	Channel 4 Pulse Generation
	See bit 0.			
3	CH3PULSE	0	W1	Channel 3 Pulse Generation
	See bit 0.			
2	CH2PULSE	0	W1	Channel 2 Pulse Generation
	See bit 0.			
1	CH1PULSE	0	W1	Channel 1 Pulse Generation
	See bit 0.			
0	CH0PULSE	0	W1	Channel 0 Pulse Generation



13.5.2 PRS_SWLEVEL - Software Level Register

Offset												·			Bi	t Po	siti	on					·									
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	8	7	9	2	4	ю	2	-	0
Reset															,										0	0	0	0	0	0	0	0
Access																									RW	RW W	RW	S.	W.	R.	W.	R.
Name																									CH7LEVEL	CH6LEVEL	CHSLEVEL	CH4LEVEL	CH3LEVEL	CH2LEVEL	CH1LEVEL	CHOLEVEL

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	CH7LEVEL	0	RW	Channel 7 Software Level
	See bit 0.			
6	CH6LEVEL	0	RW	Channel 6 Software Level
	See bit 0.			
5	CH5LEVEL	0	RW	Channel 5 Software Level
	See bit 0.			
4	CH4LEVEL	0	RW	Channel 4 Software Level
	See bit 0.			
3	CH3LEVEL	0	RW	Channel 3 Software Level
	See bit 0.			
2	CH2LEVEL	0	RW	Channel 2 Software Level
	See bit 0.			
1	CH1LEVEL	0	RW	Channel 1 Software Level
	See bit 0.			
0	CH0LEVEL	0	RW	Channel 0 Software Level
	The value in this reg the channel output.	ister is XOR'ed with the	e corresponding b	it in the SWPULSE register and the selected PRS input signal to generate

13.5.3 PRS_CHx_CTRL - Channel Control Register

Offset		Bit Position																														
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	1	10	6	∞	7	9	2	4	က	2	1	0
Reset							()	OXO					0	0000																	0×0	
Access							Ž	<u> </u>					i	≥																	RW	
Name							1000	EDSEL					1	SOURCESEL																	SIGSEL	

Bit	Name	Reset	Access	Description						
31:26	Reserved	To ensure o	compatibility with f	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)						
25:24	EDSEL	0x0	RW	Edge Detect Select						
	Select edge de	tection.								
	Value	Mode	De	escription						
	0	OFF	Signal is left as it is							



Bit	Name	Reset	Acces	s Description
	Value	Mode		Description
	1	POSEDGE		A one HFPERCLK cycle pulse is generated for every positive edge of the incoming signal
	2	NEGEDGE		A one HFPERCLK clock cycle pulse is generated for every negative edge of the incoming signal
	3	BOTHEDGES		A one HFPERCLK clock cycle pulse is generated for every edge of the incoming signal
23:22	Reserved	To ensure co	mpatibility wi	ith future devices, always write bits to 0. More information in Section 2.1 (p. 3)

21:16 SOURCESEL 0x00 RW Source Select

Select input source to PRS channel.

Value	Mode	Description
0b000000	NONE	No source selected
0b000001	VCMP	Voltage Comparator
0b000010	ACMP0	Analog Comparator 0
0b000011	ACMP1	Analog Comparator 1
0b000110	DAC0	Digital to Analog Converter 0
0b001000	ADC0	Analog to Digital Converter 0
0b010000	USART0	Universal Synchronous/Asynchronous Receiver/Transmitter 0
0b010001	USART1	Universal Synchronous/Asynchronous Receiver/Transmitter 1
0b010010	USART2	Universal Synchronous/Asynchronous Receiver/Transmitter 2
0b011100	TIMER0	Timer 0
0b011101	TIMER1	Timer 1
0b011110	TIMER2	Timer 2
0b101000	RTC	Real-Time Counter
0b101001	UART0	Universal Asynchronous Receiver/Transmitter 0
0b110000	GPIOL	General purpose Input/Output
0b110001	GPIOH	General purpose Input/Output

15:3 Reserved To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

2:0 SIGSEL 0x0 RW **Signal Select**

Select signal input to PRS channel.

Value	Mode	Description
SOURCESEL = 0b000000 (NONE)		
0bxxx	OFF	Channel input selection is turned off
SOURCESEL = 0b000001 (VCMP)		
0b000	VCMPOUT	Voltage comparator output VCMPOUT
SOURCESEL = 0b000010 (ACMP0)		
0b000	ACMP0OUT	Analog comparator output ACMP0OUT
SOURCESEL = 0b000011 (ACMP1)		
0b000	ACMP1OUT	Analog comparator output ACMP1OUT
SOURCESEL = 0b000110 (DAC0)		
0b000	DAC0CH0	DAC ch0 conversion done DAC0CH0
0b001	DAC0CH1	DAC ch1 conversion done DAC0CH1
SOURCESEL = 0b001000 (ADC0)		
0b000	ADCOSINGLE	ADC single conversion done ADC0SINGLE
0b001	ADCOSCAN	ADC scan conversion done ADC0SCAN
SOURCESEL = 0b010000 (USART0)		
0b000	USART0IRTX	USART 0 IRDA out USART0IRTX
0b001	USART0TXC	USART 0 TX complete USART0TXC
0b010	USART0RXDATAV	USART 0 RX Data Valid USART0RXDATAV
SOURCESEL = 0b010001 (USART1)		
0b001	USART1TXC	USART 1 TX complete USART1TXC
0b010	USART1RXDATAV	USART 1 RX Data Valid USART1RXDATAV
SOURCESEL = 0b010010 (USART2)		

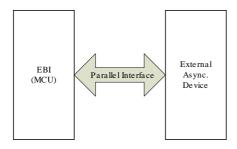


	Name Re	set Access Description	
	Value	Mode	Description
	0b001	USART2TXC	USART 2 TX complete USART2TXC
	0b010	USART2RXDATAV	USART 2 RX Data Valid USART2RXDATAV
	SOURCESEL = 0b011100 (TIMER0)		
	0b000	TIMER0UF	Timer 0 Underflow TIMER0UF
	0b001	TIMER0OF	Timer 0 Overflow TIMER0OF
ı	0b010	TIMER0CC0	Timer 0 Compare/Capture 0 TIMER0CC0
	0b011	TIMER0CC1	Timer 0 Compare/Capture 1 TIMER0CC1
ĺ	0b100	TIMER0CC2	Timer 0 Compare/Capture 2 TIMER0CC2
	SOURCESEL = 0b011101 (TIMER1)		
ĺ	0b000	TIMER1UF	Timer 1 Underflow TIMER1UF
ĺ	0b001	TIMER10F	Timer 1 Overflow TIMER1OF
İ	0b010	TIMER1CC0	Timer 1 Compare/Capture 0 TIMER1CC0
	0b011	TIMER1CC1	Timer 1 Compare/Capture 1 TIMER1CC1
İ	0b100	TIMER1CC2	Timer 1 Compare/Capture 2 TIMER1CC2
	SOURCESEL = 0b011110 (TIMER2)		
İ	0b000	TIMER2UF	Timer 2 Underflow TIMER2UF
ĺ	0b001	TIMER2OF	Timer 2 Overflow TIMER2OF
İ	0b010	TIMER2CC0	Timer 2 Compare/Capture 0 TIMER2CC0
İ	0b011	TIMER2CC1	Timer 2 Compare/Capture 1 TIMER2CC1
ı	0b100	TIMER2CC2	Timer 2 Compare/Capture 2 TIMER2CC2
ı	SOURCESEL = 0b101000 (RTC)		
	0ь000	RTCOF	RTC Overflow RTCOF
ı	0b001	RTCCOMP0	RTC Compare 0 RTCCOMP0
İ	0b010	RTCCOMP1	RTC Compare 1 RTCCOMP1
ı	SOURCESEL = 0b101001 (UART0)		
İ	0b001	UART0TXC	USART 0 TX complete UART0TXC
ı	0b010	UART0RXDATAV	USART 0 RX Data Valid UART0RXDATAV
ĺ	SOURCESEL = 0b110000 (GPIO)		
ĺ	0b000	GPIOPIN0	GPIO pin 0 GPIOPIN0
	0b001	GPIOPIN1	GPIO pin 1 GPIOPIN1
j	0b010	GPIOPIN2	GPIO pin 2 GPIOPIN2
ĺ	0b011	GPIOPIN3	GPIO pin 3 GPIOPIN3
ĺ	0b100	GPIOPIN4	GPIO pin 4 GPIOPIN4
	0b101	GPIOPIN5	GPIO pin 5 GPIOPIN5
ĺ	0b110	GPIOPIN6	GPIO pin 6 GPIOPIN6
	0b111	GPIOPIN7	GPIO pin 7 GPIOPIN7
	SOURCESEL = 0b110001 (GPIO)		
	0b000	GPIOPIN8	GPIO pin 8 GPIOPIN8
	0b001	GPIOPIN9	GPIO pin 9 GPIOPIN9
	0b010	GPIOPIN10	GPIO pin 10 GPIOPIN10
	0b011	GPIOPIN11	GPIO pin 11 GPIOPIN11
	0b100	GPIOPIN12	GPIO pin 12 GPIOPIN12
	0b101	GPIOPIN13	GPIO pin 13 GPIOPIN13
	0b110	GPIOPIN14	GPIO pin 14 GPIOPIN14
	0b111	GPIOPIN15	GPIO pin 15 GPIOPIN15



14 EBI - External Bus Interface





Quick Facts

What?

The EBI is used for accessing external parallel devices. The devices appear as a part of the EFM32G's internal memory map and are therefore extremely simple to use.

Why?

Even though the EFM32G is versatile, there might be a need for specific external devices such as extra RAM, FLASH, LCD. The EBI simplifies the access to such devices.

How?

Through memory mapping the devices appear as a part of the internal memory map. When the processor performs read or writes to the address range of the EBI, the EBI handles the data transfers to and from the external devices. The EBI may be interfaced by the DMA, thus enabling operation in EM1.

14.1 Introduction

The External Bus Interface provides access to external parallel interface devices such as SRAM, FLASH, ADCs and LCDs. The interface is memory mapped into the address bus of the Cortex-M3. This enables seamless access from software without manually manipulating the IO settings each time a read or write is performed. The data and address lines are multiplexed in order to reduce the number of pins required to interface the external devices. The bus timing is adjustable to meet specifications of the external devices. The interface is limited to asynchronous devices.

14.2 Features

- Programmable interface for various memory types
 - 4 memory bank regions
 - Individual chip select line (EBI_CSn) per memory bank
 - · Accurate control of setup, strobe, hold and turn-around timing
 - Individual active high / active low setting of interface control signals
 - Slave read/write cycle extension
- · Multiplexed data and address lines for reduced pin count
 - Up to 24 address lines
 - Up to 16-bit data bus width
 - 8-bit true parallel operation

14.3 Functional Description

An overview of the EBI module is shown in Figure 14.1 (p. 137).

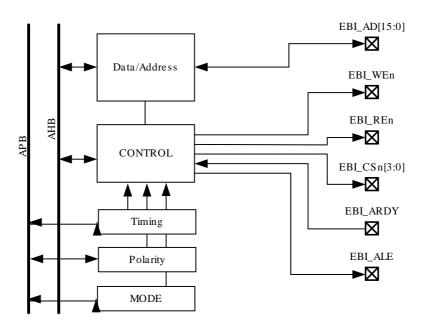
The EBI has multiplexed and non-multiplexed addressing modes. Fastest operation is achieved when using a non-multiplexed addressing mode. The multiplexed addressing modes are somewhat slower



and require an external latch, but they use a significantly lower number of pins. The use of the 16 EBI_AD pin connections depends on the addressing mode. They are used for both address and data in the multiplexed modes. Also for the non-multiplexed 8-bit address mode both the address and data fit into these 16 EBI_AD pins. If more address bits or data bits are needed, external latches can be used to support up to 24-bit addresses or 16-bit data in the multiplexed addressing modes using only the 16 EBI_AD pins.

When a read operation is requested by the Cortex-M3 or DMA via the EBI's AHB interface, the address is transferred onto the EBI_AD bus. After a specific number of cycles, the EBI_REn pin is activated and data is read from the EBI_AD bus. When a write operation is requested, the address is transferred onto the EBI_AD bus and subsequently the write data is transferred onto the EBI_AD bus as the EBI_WEn pin is activated. The detailed operation in the supported modes is presented in the following sections.

Figure 14.1. EBI Overview



14.3.1 8-bit Address Mode

In this mode, 8-bit address and 8-bit data is supported. The address is put on the higher 8 bits of the EBI_AD lines while the data uses the lower 8 bits. This mode is set by programming the MODE field in the EBI_CTRL register to D8A8. Read and write signals in 8-bit mode are shown in Figure 14.2 (p. 137) and Figure 14.3 (p. 138) respectively.

Figure 14.2. EBI Non-multiplexed 8-bit Data, 8-bit Address Read Operation

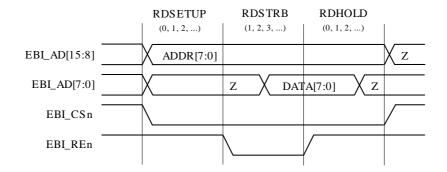
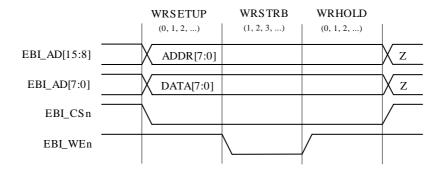


Figure 14.3. EBI Non-multiplexed 8-bit Data, 8-bit Address Write Operation



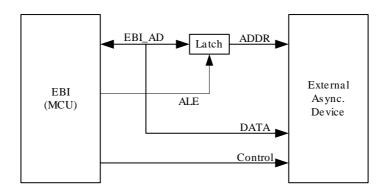
14.3.2 16-bit Address Mode

In this mode, 16-bit address and 16-bit data is supported, but the utilization of an external latch is required. The 16-bit address and 16-bit data bits are multiplexed on the EBI_AD lines. An illustration of such a setup is shown in Figure 14.4 (p. 138). This mode is set by programming the MODE field in the EBI_CTRL register to D16A16ALE.

Note

In this mode the 16-bit address is organized in 2-byte chunks at memory addresses aligned to 2-byte offsets. Consequently, the LSB of the 16-bit address will always be 0. In order to double the address space, the 16-bit address is internally shifted one bit to the right so that the LSB of the address driven into the EBI_AD bus, i.e. the EBI_AD[0]-bit, corresponds to the second least significant bit of the address, i.e. ADDR[1]. At the external device, the LSB of the address must be tied either low or high in order to create a full address.

Figure 14.4. EBI Address Latch Setup



At the start of the transaction the address is output on the EBI_AD lines. The Latch is controlled by the ALE (Address Latch Enable) signal and stores the address. Then the data is read or written according to operation. Read and write signals are shown in Figure 14.5 (p. 139) and Figure 14.6 (p. 139) respectively.

Figure 14.5. EBI Multiplexed 16-bit Data, 16-bit Address Read Operation

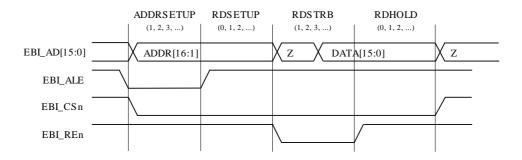
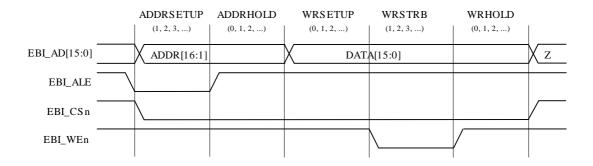


Figure 14.6. EBI Multiplexed 16-bit Data, 16-bit Address Write Operation



14.3.3 24-bit Address Mode

This mode allows 24-bit address with 8-bit data multiplexed on the EBI_AD lines. The upper 8 bits of the EBI_AD lines are consecutively used for the highest 8 bits and the lowest 8 bits of the address. The lower 8 bits of the EBI_AD lines are used for the middle 8 address bits and for data. This mode is set by programming the MODE field in the EBI_CTRL register to D8A24ALE. Read and write signals are shown in Figure 14.7 (p. 139) and Figure 14.8 (p. 140) respectively.

Figure 14.7. EBI Multiplexed 8-bit Data, 24-bit Address Read Operation

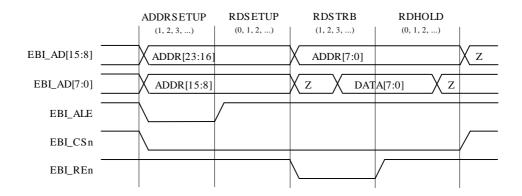
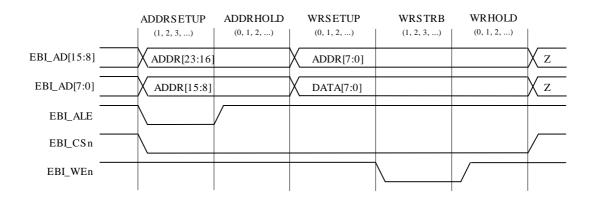




Figure 14.8. EBI Multiplexed 8-bit Data, 24-bit Address Write Operation



14.3.4 Timing

The duration of the states in the transaction is defined by the corresponding uppercase name above the state, e.g. the address setup state in Figure 14.8 (p. 140) is active for a number of internal clock cycles defined by ADDRSET bitfield in the EBI_ADDRTIMING register. Similar timing can be defined by the RDSTRB bitfield in the EBI_RDTIMING register and WRSTRB in the EBI_WRTIMING register. These parameters all have a minimum duration of 1 cycle, which is set by HW in case the bitfield is programmed to 0.

The setup and hold timing parameters are ADDRHOLD in the EBI_ADDRTIMING register, RDHOLD and RDSETUP in the EBI_RDTIMING register and WRHOLD and WR SETUP in the EBI_WRTIMING register. Writing a value m to one of these biffields results in a duration of the corresponding state of m cycles. If these parameters are set to 0, it effectively means that the state is skipped.

14.3.5 Data Access Width

It is important that the setting of the data width of the external device is respected. If the width of a request does not match the data width specified in the MODE field of the EBI_CTRL register, a bus fault is generated.

14.3.6 Bank Access

The EBI is split in 4 different address regions, each connected to an individual EBI_CSn line. When accessing one of the memory regions, the corresponding CSn line is asserted. This way up to 4 separate devices can share the EBI lines and be identified by the EBI_CSn line. Each bank can individually be enabled or disabled in the EBI_CTRL register. The bank separation is 64 MB. Refer to the memory map of the EFM32G for a more detailed specification on the memory locations available.



Figure 14.9. EBI Default Memory Map (ALTMAP = 0)

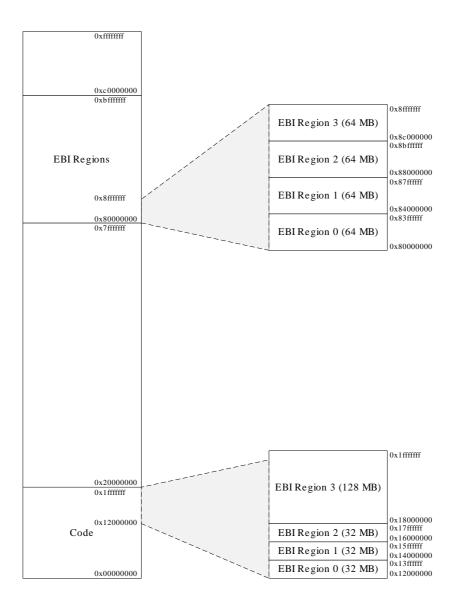
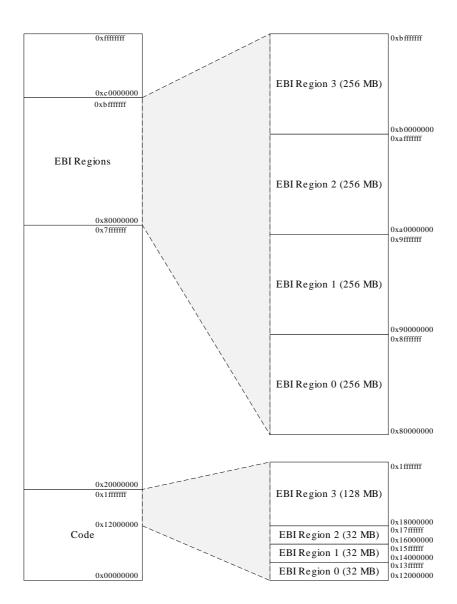




Figure 14.10. EBI Alternative Memory Map (ALTMAP = 1)



14.3.7 WAIT/ARDY.

Some external devices are able to indicate that they are not finished with either write or read operation by asserting the WAIT / ARDY line. This input signal is used to extend the REn/WEn cycles for slow devices. The interpretation of the polarity of this signal can be configured with the ARDYPOL bit in EBI_POLARITY. E.g. if the ARDYPOL is set to ACTIVELOW, then the REn/WEn cycle is extended while the ARDY line is kept low. The ARDY functionality is enabled by setting the ARDYEN bit in the EBI_CTRL register. It is also possible to enable a timeout check, which generates a bus error if the ARDY is not deasserted within the timeout period. This prevents a system lock up condition in the case that the external device does not deassert ARDY. The timeout functionality is disabled by setting ARDYTODIS in the EBI_CTRL register.

14.3.8 Control Signal Polarity

It is possible to individually configure the control signals to be active high/low by setting or clearing the appropriate bits in the EBI_POLARITY register.

14.3.9 Pin Configuration

In order to give the EBI access to the external pins of the EFM32G, the GPIO must be configured accordingly. The lines must be set to Push-Pull, which is described in detail in the GPIO section.



All the EBI pins are enabled in the EBI_ROUTE register. The EBI_AD, EBI_WEn and EBI_REn pins are all enabled by the EBIPEN bit, the EBI_CSn pins are enabled by the corresponding CSxPEN bit, the EBI_ALE pin is enabled by the ALEPEN bit , and the EBI_ARDY pin is enabled by the ARDYPEN bit of the EBI_ROUTE register.



14.4 Register Map

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

Offset	Name	Туре	Description
0x000	EBI_CTRL	RW	Control Register
0x004	EBI_ADDRTIMING	RW	Address Timing Register
0x008	EBI_RDTIMING	RW	Read Timing Register
0x00C	EBI_WRTIMING	RW	Write Timing Register
0x010	EBI_POLARITY	RW	Polarity Register
0x014	EBI_ROUTE	RW	I/O Routing Register

14.5 Register Description

14.5.1 EBI_CTRL - Control Register

Offset		Bit Position																														
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	œ	7	9	2	4	ю	2	-	0
Reset												•		,	0	0		•			0	0	0	0						,	6) X
Access												RW	W.					RW	RW	RW	RW							ž	 ≩ Y			
Name															ARDYTODIS	ARDYEN					BANK3EN	BANK2EN	BANK1EN	BANKOEN							L	MODE

Bit	Name	Reset	Access	Description										
31:18	Reserved	To ensure c	ompatibility with f	with future devices, always write bits to 0. More information in Section 2.1 (p. 3										
17	ARDYTODIS	0	RW	ARDY Timeout Disable										
	Enables or dis	ables the ARDY timeout fu	nctionality. The ti	meout value is 32 internal clock cycles.										
16	ARDYEN	0	RW	ARDY Enable										
	Enables or dis	ables the ARDY functional	ty.											
15:12	Reserved	future devices, always write bits to 0. More information in Section 2.1 (p. 3)												
11	BANK3EN	0	RW	Bank 3 Enable										
	This field enab	les or disables bank 3.												
10	BANK2EN	0	RW	Bank 2 Enable										
	This field enab	This field enables or disables bank 2.												
9	BANK1EN	0	RW	Bank 1 Enable										
	This field enab	les or disables bank 1.												
8	BANK0EN	0	RW	Bank 0 Enable										
	This field enab	les or disables bank 0.												
7:2	Reserved	To ensure c	ompatibility with f	future devices, always write bits to 0. More information in Section 2.1 (p. 3)										
1:0	MODE	0x0	RW	Mode										
	This field sets	This field sets the access mode the EBI will use for interfacing devices.												
	Value	Mode	De	Description										
	0	D8A8	8 b	8 bit data, 8 bit address, ALE not used.										
	1	D16A16ALE	16	16 bit data, 16 bit address, ALE is used for address latching.										
	2	D8A24ALE	8 b	oit data, 24 bit address, ALE is used for address latching.										



14.5.2 EBI_ADDRTIMING - Address Timing Register

Offset															Bit	Ро	sitio	on														
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	8	7	9	2	4	3	2	-	0
Reset																							5	<u>.</u>							0	000
Access																							7	2							i	X N
Name																							טוסטט ע	ADDANGE							<u>.</u> H L	ADDRSETUP

Bit	Name	Reset	Access	Description
31:10	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
9:8	ADDRHOLD	0x1	RW	Address Hold Time
	Sets the number of cy	cles the address is	held after ALE is a	asserted.
7:2	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1:0	ADDRSETUP	0x0	RW	Address Setup Time
	Sets the number of cy	cles the address is o	driven onto the AD	DRDAT bus before ALE is asserted. If set to 0, 1 cycle is inserted by HW.

14.5.3 EBI_RDTIMING - Read Timing Register

Offset															Bi	t Po	siti	on														
0x008	33	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	∞	7	9	2	4	က	2	-	0
Reset									•						5	2						,	OXO								2	e e e
Access															3	<u> </u>						3	<u>}</u>								3	<u>}</u>
Name															0	חטרטא						90	פאופטא								0.1	2

Bit	Name	Reset	Access	Description
31:18	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
17:16	RDHOLD	0x0	RW	Read Hold Time
	Sets the number o	f cycles CSn is held act	ive after the REn	is deasserted. This interval is used for bus turnaround.
15:12	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
11:8	RDSTRB	0x0	RW	Read Strobe Time
	Sets the number of	cycles the REn is held a	active. After the sp	ecified number of cycles, data is read. If set to 0, 1 cycle is inserted by HW. $ \frac{1}{2} \left(\frac{1}{2} \right) = \frac{1}{2} \left(\frac{1}{2} \right)$
7:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1:0	RDSETUP	0x0	RW	Read Setup Time
	Sets the number of	f cycles the address se	tup before REn is	asserted.



14.5.4 EBI_WRTIMING - Write Timing Register

Offset															Bi	t Po	siti	on														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	∞	7	9	2	4	က	2	-	0
Reset														,	5	5				,			OXO									OXO
Access															2	2						i	≥ Y								i	≩
Name																1							WKS I								i i	WKSELUP

Bit	Name	Reset	Access	Description
31:18	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
17:16	WRHOLD	0x1	RW	Write Hold Time
	Sets the number of	cycles CSn is held act	ive after the WEn	is deasserted.
15:12	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
11:8	WRSTRB	0x0	RW	Write Strobe Time
	Sets the number of	cycles the WEn is held	d active. If set to 0	, 1 cycle is inserted by HW.
7:2	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1:0	WRSETUP	0x0	RW	Write Setup Time
	Sets the number of	cycles the address se	tup before WEn is	asserted.

14.5.5 EBI_POLARITY - Polarity Register

Offset	Bit Position				,	
0x010	3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	4	ю	2	-	0
Reset		0	0	0	0	0
Access		R W	W.	RW	RW	R₩
Name		ARDYPOL	ALEPOL	WEPOL	REPOL	CSPOL

Bit	Name	Reset	Acces	s Description
31:5	Reserved	To ensure c	ompatibility wit	h future devices, always write bits to 0. More information in Section 2.1 (p. 3)
4	ARDYPOL	0	RW	ARDY Polarity
	Sets the polar	ity of the EBI_ARDY line.		
	Value	Mode		Description
	0	ACTIVELOW		ARDY is active low.
	1	ACTIVEHIGH		ARDY is active high.
3	ALEPOL	0	RW	Address Latch Polarity
	Sets the polar	ity of the EBI_ALE line.		
	Value	Mode		Description
	0	ACTIVELOW		ALE is active low.
	1	ACTIVEHIGH		ALE is active high.
2	WEPOL	0	RW	Write Enable Polarity

Sets the polarity of the EBI_WEn line.



Bit	Name	Reset	Acce	ss Description	
	Value	Mode		Description	
	0	ACTIVELOW		WEn is active low.	
	1	ACTIVEHIGH		WEn is active high.	
1	REPOL	0	RW	Read Enable Polarity	
	Sets the polar	rity of the EBI_REn line.			
	Value	Mode		Description	
	0	ACTIVELOW		REn is active low.	
	1	ACTIVEHIGH		REn is active high.	
0	CSPOL	0	RW	Chip Select Polarity	
	Sets the polar	rity of the EBI_CSn line.			
	Value	Mode		Description	
	0	ACTIVELOW		CSn is active low.	
	1	ACTIVEHIGH		CSn is active high.	

14.5.6 EBI_ROUTE - I/O Routing Register

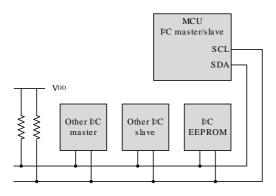
Offset															Bi	t Po	siti	on														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	8	7	9	2	4	က	2	-	0
Reset											•		•										,			0	0	0	0	0	0	0
Access																										RW	RW	RW	RW	RW	RW W	RW
Name																										ARDYPEN	ALEPEN	CS3PEN	CS2PEN	CS1PEN	CSOPEN	EBIPEN

Bit	Name	Reset	Access	Description
31:7	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
6	ARDYPEN	0	RW	EBI_ARDY Pin Enable
	When set, the EBI_	ARDY pin is enabled		
5	ALEPEN	0	RW	EBI_ALE Pin Enable
	When set, the EBI_	ALE pin is enabled		
4	CS3PEN	0	RW	EBI_CS3 Pin Enable
	When set, the EBI_	CS3 pin is enabled		
3	CS2PEN	0	RW	EBI_CS2 Pin Enable
	When set, the EBI_	CS2 pin is enabled		
2	CS1PEN	0	RW	EBI_CS1 Pin Enable
	When set, the EBI_	CS1 pin is enabled		
1	CS0PEN	0	RW	EBI_CS0 Pin Enable
	When set, the EBI_	CS0 pin is enabled		
0	EBIPEN	0	RW	EBI Pin Enable
	When set, the EBI_	AD[15:0], EBI_WEn an	d EBI_REn pins	are enabled



15 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.

15.1 Introduction

The I²C module provides an interface between the MCU and a serial I²C-bus. It is capable of acting as both master and slave, and supports multi-master 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. Slave arbitration and timeouts are also provided to allow implementation of an SMBus compliant system. The interface provided to software by the I²C module allows both fine-grained control of the transmission process and close to automatic transfers. Automatic recognition of slave addresses is provided in all energy modes (except EM4).

15.2 Features

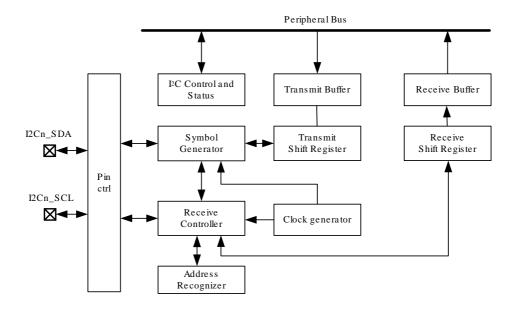
- True multi-master capability
- Support for different bus speeds
 - Standard-mode (Sm) bit rate up to 100 kbit/s
 - · Fast-mode (Fm) bit rate up to 400 kbit/s
 - Fast-mode Plus (Fm+) bit rate up to 1 Mbit/s
- Arbitration for both master and slave (allows SMBus ARP)
- Clock synchronization and clock stretching
- · Hardware address recognition
 - · 7-bit masked address
 - · General call address
 - · Active in all energy modes (except EM4)
- 10-bit address support
- Error handling
 - Clock low timeout
 - · Clock high timeout
 - Arbitration lost
 - · Bus error detection
- · Double buffered data
- Full DMA support



15.3 Functional Description

An overview of the I²C module is shown in Figure 15.1 (p. 149).

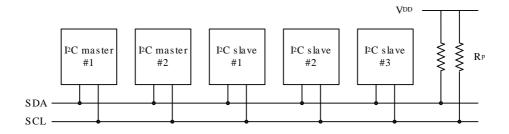
Figure 15.1. I²C Overview



15.3.1 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 15.2 (p. 149). As a true multi-master bus it includes collision detection and arbitration to resolve situations where multiple masters transmit data at the same time without data loss.

Figure 15.2. I²C-Bus Example



Each device on the bus is addressable by a unique address, and an I²C master can address all the devices on the bus, including other masters.

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 Equation 15.1 (p. 149).

PC Pull-up Resistor Equation

$$Rp(max) = (tr/0.8473) x Cb.$$
 (15.1)

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 drive strength can be used to control slew rate.



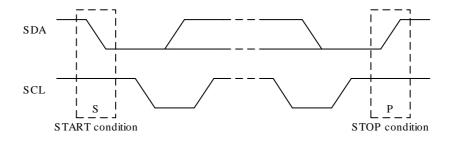
Note

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.

15.3.1.1 START and STOP Conditions

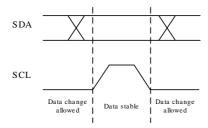
START and STOP conditions are used to initiate and stop transactions on the I^2 C-bus. All transactions on the bus begin with a START condition (S) and end with a STOP condition (P). As shown in Figure 15.3 (p. 150), a START condition is generated by pulling the SDA line low while SCL is high, and a STOP condition is generated by pulling the SDA line high while SCL is high.

Figure 15.3. 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 15.2 (p. 149).

Figure 15.4. I²C Bit Transfer on I²C-Bus



15.3.1.2 Bus Transfer

When a master wants to initiate a transfer on the bus, it waits until the bus is idle and transmits a START condition on the bus. The master then transmits the address of the slave it wishes to interact with and a single R/W bit telling whether it wishes to read from the slave (R/W bit set to 1) or write to the slave (R/W bit set to 0).

After the 7-bit address and the R/W bit, the master releases the bus, allowing the slave to acknowledge the request. During the next bit-period, the slave 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 slave or master 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 master ends the transmission after a (N)ACK by sending a STOP condition on the bus. After a STOP condition, any master wishing to initiate a transfer



on the bus can try to gain control of it. If the current master 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 I²C transfers are shown in Figure 15.5 (p. 151), Figure 15.6 (p. 151), and Figure 15.7 (p. 151) . 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 15.5. I²C Single Byte Write to Slave



Figure 15.6. I²C Double Byte Read from Slave



Figure 15.7. I²C Single Byte Write, then Repeated Start and Single Byte Read



15.3.1.3 Addresses

 I^2C 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 slave that the master wants to contact. In the 7-bit address space, several addresses are reserved. These addresses are summarized in Table 15.1 (p. 151), and include a General Call address which can be used to broadcast a message to all slaves on the I^2C -bus.

Table 15.1. I²C Reserved I²C Addresses

I ² C 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 slave addressing mode



15.3.1.4 10-bit Addressing

To address a slave 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 eight bit of the first byte determines whether the master wishes to read from or write to the slave. The second byte contains the eight least significant bits of the slave address.

When a slave receives a 10-bit address, it must acknowledge both the address bytes if they match the address of the slave.

When performing a master transmitter operation, the master transmits the two address bytes and then the remaining data, as shown in Figure 15.8 (p. 152).

Figure 15.8. I²C Master Transmitter/Slave Receiver with 10-bit Address



When performing a master receiver operation however, the master first transmits the two address bytes in a master transmitter operation, then sends a repeated START followed by the first address byte and then receives data from the addressed slave. The slave 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 15.9 (p. 152) .

Figure 15.9. I²C Master Receiver/Slave Transmitter with 10-bit Address



15.3.1.5 Arbitration, Clock Synchronization, Clock Stretching

Arbitration and clock synchronization are features aimed at allowing multi-master 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 masters driving the bus at once, and is a requirement for effective arbitration.

Slaves 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 I²C module for both masters and slaves. Clock synchronization and clock stretching is also supported.

15.3.2 Enable and Reset

The I^2C is enabled by setting the EN bit in the $I2Cn_CTRL$ register. Whenever this bit is cleared, the internal state of the I^2C is reset, terminating any ongoing transfers.

Note



When enabling the I²C, the ABORT command or the Bus Idle Timeout feature must be applied prior to use even if the BUSY flag is not set.

15.3.3 Safely Disabling and Changing Slave Configuration

The I²C slave is partially asynchronous, and some precautions are necessary to always ensure a safe slave disable or slave configuration change. These measures should be taken, if (while the slave is enabled) the user cannot guarantee that an address match will not occur at the exact time of slave disable or slave configuration change.

Worst case consequences for an address match while disabling slave or changing configuration is that the slave may end up in an undefined state. To reset the slave back to a known state, the EN bit in I2Cn_CTRL must be reset. This should be done regardless of whether the slave is going to be re-enabled or not.

15.3.4 Clock Generation

The SCL signal generated by the I²C master determines the maximum transmission rate on the bus. The clock is generated as a division of the peripheral clock, and is given by Equation 15.2 (p. 153):

$$f_{SCL} = 1/(T_{low} + T_{high}), \tag{15.2}$$

where

T_{low} and T_{high} is the low and high periods of the clock signal respectively, given below. When the clock is not streched, the low and high periods of the clock signal are:

$$T_{high} = (N_{high} \times (CLKDIV + 1))/f_{HFPERCLK},$$

$$T_{low} = (N_{low} \times (CLKDIV + 1))/f_{HFPERCLK}.$$
(15.3)

Equation 15.3 (p. 153) and Equation 15.2 (p. 153) does not apply for low clock division factors (0, 1 and 2) because of synchronization. For these clock division factors, the formulas for computing high and low periods of the clock signal are given in Table 15.2 (p. 153).

Table 15.2. I²C High and Low Periods for Low CLKDIV

CLKDIV	Standard (4:4)		Asymmetric (6:3)		Fast (11:6)	
	T _{low}	T _{high}	T _{low}	T _{high}	T _{low}	T _{high}
0	7/f _{HFPERCLK}	7/f _{HFPERCLK}	9/f _{HFPERCLK}	6/f _{HFPERCLK}	14/f _{HFPERCLK}	9/f _{HFPERCLK}
1	10/f _{HFPERCLK}	10/f _{HFPERCLK}	14/f _{HFPERCLK}	8/f _{HFPERCLK}	24/f _{HFPERCLK}	14/f _{HFPERCLK}
2	15/f _{HFPERCLK}	15/f _{HFPERCLK}	21/f _{HFPERCLK}	12/f _{HFPERCLK}	36/f _{HFPERCLK}	21/f _{HFPERCLK}

The values of N_{low} and N_{high} and thus the ratio between the high and low parts of the clock signal is controlled by CLHR in the I2Cn_CTRL register. The available modes are summarized in Table 15.3 (p. 154) along with the highest I²C-bus frequencies in the given modes that can be achieved without violating the timing specifications of the I²C-bus. The maximum data hold time is dependent on the DIV and is given by:

Maximum Data Hold Time

$$t_{HD,DAT-max} = (4+DIV)/f_{HFPERCLK}.$$
 (15.4)

www.silabs.com

Note

DIV must be set to 1 or higher during slave mode operation.



Table 15.3. I²C Clock Mode

HFPERCLK frequency (MHz)	Clock Low High Ratio (CLHR)	Sm max frequency (kHz)	Fm max frequency (kHz)	Fm+ max frequency (kHz)
32	0	93	400	1000
	1	82	400	969
	2	72	400	842
28	0	92	400	1000
	1	81	400	848
	2	71	400	736
21	0	93	400	1000
	1	83	400	954
	2	72	368	552
14	0	92	400	999
	1	81	400	636
	2	68	368	608
11	0	91	400	785
	1	81	333	733
	2	71	289	478
6.6	0	91	400	471
	1	81	299	439
	2	64	286	286
1.2	0	59	85	85
	1	54	79	79
	2	52	52	52

15.3.5 Arbitration

Arbitration is enabled by default, but can be disabled by setting the ARBDIS bit in I2Cn_CTRL. When arbitration is enabled, the value on SDA is sensed every time the I²C module attempts to change its value. If the sensed value is different than the value the I²C module tried to output, it is interpreted as a simultaneous transmission by another device, and that the I²C module has lost arbitration.

Whenever arbitration is lost, the ARBLOST interrupt flag in I2Cn_IF is set, any lines held are released, and the I²C device goes idle. If an I²C master loses arbitration during the transmission of an address, another master may be trying to address it. The master therefore receives the rest of the address, and if the address matches the slave address of the master, the master goes into either slave transmitter or slave receiver mode.

Note

Arbitration can be lost both when operating as a master and when operating as a slave.

15.3.6 Buffers

15.3.6.1 Transmit Buffer and Shift Register

The I²C transmitter is double buffered through the transmit buffer and transmit shift register as shown in Figure 15.1 (p. 149). A byte is loaded into the transmit buffer by writing to I2Cn_TXDATA. 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 I2Cn_STATUS and the TXC interrupt flags in I2Cn_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.

Whenever a byte is loaded from the transmit buffer to the transmit shift register, the TXBL flag in I2Cn_STATUS and the TXBL interrupt flag in I2Cn_IF are set. This indicates that there is room in the buffer for more data. TXBL is cleared automatically when data is written to the buffer.

If a write is attempted to the transmit buffer while it is not empty, the TXOF interrupt flag in I2Cn_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 I2Cn_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.

15.3.6.2 Receive Buffer and Shift Register

Like the transmitter, the I²C receiver is double buffered. The receiver uses the receive buffer and receive shift register as shown in Figure 15.1 (p. 149). 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. Otherwise, the byte waits in the shift register until space becomes available in the buffer.

When a byte becomes available in the receive buffer, the RXDATAV in I2Cn_STATUS and RXDATAV interrupt flag in I2Cn_IF are set. The data can now be fetched from the buffer using I2Cn_RXDATA. Reading from this register will pull a byte out of the buffer, making room for a new byte and clearing RXDATAV in I2Cn_STATUS and RXDATAV in I2Cn_IF in the process.

If a read from the receive buffer is attempted through I2Cn_RXDATA while the buffer is empty, the RXUF interrupt flag in I2Cn_IF is set, and the data read from the buffer is undefined.

I2Cn_RXDATAP can be used to read data from the receive buffer without removing it from the buffer. The RXUF interrupt flag in I2Cn_IF will never be set as a result of reading from I2Cn_RXDATAP, but the data read through I2Cn_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 picked up) before starting a new transaction.

15.3.7 Master Operation

A bus transaction is initiated by transmitting a START condition (S) on the bus. This is done by setting the START bit in I2Cn_CMD. The command schedules a START condition, and makes the I²C module generate a start condition whenever the bus becomes free.

The I²C-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 master issuing the START condition. The bus is considered free when a STOP condition is transmitted on the bus. After a STOP is detected, all masters 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 master must transmit a slave address (ADDR) with an R/W bit on the bus. If this address is available in the transmit buffer, the master transmits it immediately, but if the buffer is empty, the master holds the I²C-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 slave, depending on the value of the R/W bit (bit 0 in the address byte). If the bit was cleared, the master has entered a master transmitter role, where it now transmits data to the slave. If the bit was set, it has entered a master receiver role, where it now should receive data from the slave. In either case, an unlimited number of bytes can be transferred in one direction during the transmission.

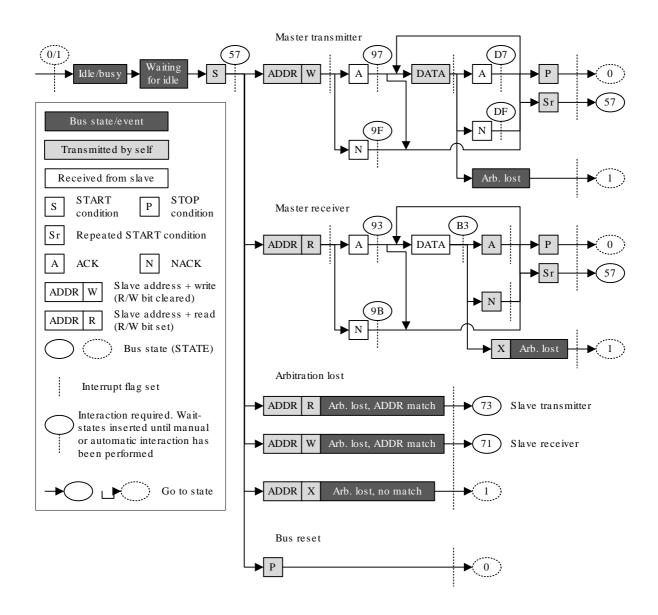
At the end of the transmission, the master either transmits a repeated START condition (Sr) if it wishes to continue with another transfer, or transmits a STOP condition (P) if it wishes to release the bus.

15.3.7.1 Master State Machine

The master state machine is shown in Figure 15.10 (p. 156). A master 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 15.10. I²C Master State Machine





15.3.7.2 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 I2Cn_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 I2Cn_STATE register.

As an example, Table 15.5 (p. 159) shows the different states the I²C goes through when operating as a Master Transmitter, i.e. a master that transmits data to a slave. 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, I2Cn_STATE has a value 0x57, which can be used to identify exactly what the I²C module is waiting for.

Note

The bus would never stop at state 0x57 if the address was available in the transmit buffer.

The different interactions used by the I²C module are listed in Table 15.4 (p. 157) in prioritized order. If a set of different courses of action are possible from a given state, the course of action using the highest priority interactions, that first has everything it is waiting for is the one that is taken.

Table 15.4. I²C Interactions in Prioritized Order

Interaction	Priority	Software action	Automatically continues if
STOP*	1	Set the STOP command bit in I2Cn_CMD	PSTOP is set (STOP pending) in I2Cn_STATUS
ABORT	2	Set the ABORT command bit in I2Cn_CMD	Never, the transmission is aborted
CONT*	3	Set the CONT command bit in I2Cn_CMD	PCONT is set in I2Cn_STATUS (CONT pending)
NACK*	4	Set the NACK command bit in I2Cn_CMD	PNACK is set in I2Cn_STATUS (NACK pending)
ACK*	5	Set the ACK command bit in I2Cn_CMD	AUTOACK is set in I2Cn_CTRL or PACK is set in I2Cn_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 I2Cn_CMD	PSTART is set in I2Cn_STATUS (START pending)
TXDATA	9	Write data to the transmit buffer	Data is available in transmit buffer
RXDATA	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 15.4 (p. 157) 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 I2Cn_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 15.4 (p. 157) is applied to the bus.

Pending commands can be cleared by setting the CLEARPC command bit in I2Cn_CMD.

15.3.7.2.1 Automatic ACK Interaction

When receiving addresses and data, an ACK command in I2Cn_CMD is normally required after each received byte. When AUTOACK is set in I2Cn_CTRL, an ACK is always pending, and the ACK-pending bit PACK in I2Cn_STATUS is thus always set, even after an ACK has been consumed. This can be used to reduce the amount of software interaction required during a transfer.

15.3.7.3 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 I^2C -STATUS is thus set. To be able to carry through master 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 I²C module detects that the bus is idle can be significant. There are two ways of assuring that the I²C module gets out of the busy state.

- Use the ABORT command in I2Cn_CMD. When the ABORT command is issued, the I²C module is instructed that the bus is idle. The I²C module can then initiate master 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 I2Cn_CTRL to an appropriate timeout period and set GIBITO in I2Cn_CTRL. If activity has not been detected on the bus within the timeout period, the bus is then automatically assumed idle, and master operations can be initiated.

Note

If operating in slave mode, the above approach is not necessary.

15.3.7.4 Master Transmitter

To transmit data to a slave, the master must operate as a master transmitter. Table 15.5 (p. 159) shows the states the I²C module goes through while acting as a master 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 I2Cn_IF is set, the module is waiting for an interaction, and the bus is frozen. The value of I2Cn_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 I2Cn_IF is set when the I²C module transmits the START.

A master operation is started by issuing a START command by setting START in I2Cn_CMD. ADDR +W, i.e. the address of the slave to address + 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 I2Cn_STATE will then be 0x57. 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 I2Cn_TXDATA with the address of the slave in the 7 most significant bits and the least significant bit cleared (ADDR+W). This address will then be transmitted, and the slave will reply with an ACK or a NACK. If no slave replies to the address, the response will also be NACK. If the address was acknowledged, the master now has four choices. It can send a data byte by placing it in I2Cn_TXDATA (the master should check the TXBL interrupt flag before writing to I2Cn_TXDATA), this byte is then transmitted. The master 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 a NACK was received, the master 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 master has the same options as for when the address was sent.

The master may lose arbitration at any time during transmission. In this case, the ARBLOST interrupt flag in I2Cn_IF is set. If the arbitration was lost during the transfer of an address, and SLAVE in I2Cn_CTRL is set, the master then checks which address was transmitted. If it was the address of the master, then the master goes to slave mode.

After a master 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 master gains the bus next. The MSTOP interrupt flag in I2Cn_IF is set when a STOP condition is transmitted by the master.

Table 15.5. I²C Master Transmitter

I2Cn_STA	Description	I2Cn_IF	Required interaction	Response
0x57	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.
0x57	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	
0x97	ADDR+W transmitted,	ACK interrupt flag	TXDATA	DATA will be sent
	ACK received	(BUSHOLD interrupt flag)	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	NACK (BUSHOLD interrupt flag)	CONT + TXDATA	DATA will be sent
	received		STOP	STOP will be sent. Bus will be released



I2Cn_STA	Description	I2Cn_IF	Required interaction	Response
			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	ACK interrupt flag	TXDATA	DATA will be sent
	received	(BUSHOLD interrupt flag)	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	NACK(BUSHOLD interrupt flag)	CONT + TXDATA	DATA will be sent
	received		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		None	
		flag	START	START will be sent when bus becomes idle

15.3.7.5 Master Receiver

To receive data from a slave, the master must operate as a master receiver, see Table 15.6 (p. 161). This is done by transmitting ADDR+R as the address byte instead of ADDR+W, which is transmitted to become a master transmitter. The address byte loaded into the data register thus has to contain the 7-bit slave address in the 7 most significant bits of the byte, and have the least significant bit set.

When the address has been transmitted, the master receives an ACK or a NACK. If an ACK is received, the ACK interrupt flag in I2Cn_IF is set, and if space is available in the receive shift register, reception of a byte from the slave 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 slave.

If a NACK was received, the CONT command in I2Cn_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 master must ACK or NACK the received byte. If an ACK is pending or AUTOACK in I2Cn_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 I2Cn_CMD is set alone, a repeated start condition is transmitted after the ACK/NACK. If STOP in I2Cn_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 master transmitter, arbitration can be lost as a master receiver. When this happens the ARBLOST interrupt flag in I2Cn_IF is set, and the master has a possibility of being selected as a slave given the correct conditions.

Table 15.6. I²C Master Receiver

I2Cn_STAT	Description	I2Cn_IF	Required interaction	Response
0x57	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.
0x57	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	
0x93	ADDR+R transmitted,	ACK interrupt	RXDATA	Start receiving
	ACK received	flag(BUSHOLD)	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	ansmitted,NACK	CONT + RXDATA	Continue, start receiving
	received		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
0xB3	Data received	RXDATA interrupt flag(BUSHOLD	ACK + RXDATA	ACK will be transmitted, reception continues
		interrupt flag)	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	



I2Cn_STAT	Description	I2Cn_IF	Required interaction	Response
			START	START will be sent when bus becomes idle
-	Arbitration lost	ARBLOST interrupt	None	
		flag	START	START will be sent when bus becomes idle

15.3.8 Bus States

The I2Cn_STATE register can be used to determine which state the I^2C module and the I^2C bus are in at a given time. The register consists of the STATE bit-field, which shows which state the I^2C 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 I^2C module waiting for a software response.

The possible values of the STATE field are summarized in Table 15.7 (p. 162). When this field is cleared, the I^2C module is not a part of any ongoing transmission. The remaining status bits in the I^2Cn_STATE register are listed in Table 15.8 (p. 162).

Table 15.7. 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 being transmitted
ADDR	3	Address being transmitted or has been received
ADDRACK	4	Address ACK/NACK being transmitted or received
DATA	5	Data being transmitted or received
DATAACK	6	Data ACK/NACK being transmitted or received

Table 15.8. 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 master. Cleared at all other times.
TRANSMITTER	Set when operating as a transmitter; either a master transmitter or a slave transmitter. Cleared at all other times
BUSHOLD	Set when the bus is held by this I ² C 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

I2Cn_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 I2Cn_STATUS is set.

15.3.9 Slave Operation

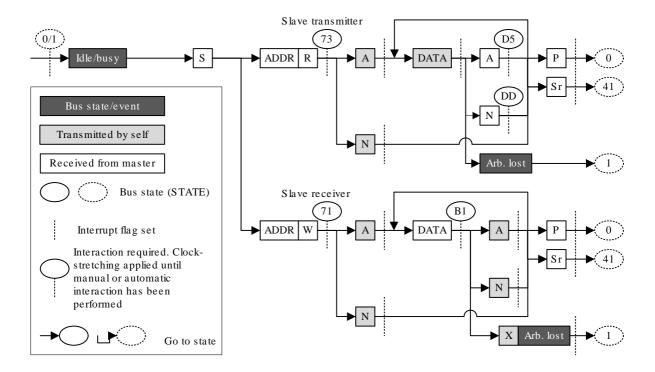
The I²C module operates in master mode by default. To enable slave operation, i.e. to allow the device to be addressed as an I²C slave, the SLAVE bit in I2Cn_CTRL must be set. In this case the slave operates in a mixed mode, both capable of starting transmissions as a master, and being addressed as a slave. When operating in the slave mode, HFPERCLK frequency must be higher than 4.2 MHz for Standard-mode, 11 MHz for Fast-mode, and 24.4 MHz for Fast-mode Plus.



15.3.9.1 Slave State Machine

The slave state machine is shown in Figure 15.11 (p. 163). 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 15.11. I²C Slave State Machine



15.3.9.2 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 Section 15.3.11 (p. 167). Address recognition is supported in all energy modes (except EM4).

The slave address, i.e. the address which the I²C module should be addressed with, is defined in the I2Cn_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 I2Cn_SADDR. The mask is defined in I2Cn_SADDRMASK, and for every zero in the mask, the corresponding bit in the slave address is treated as a don't-care.

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 I2Cn_SADDR, while a mask 0x70 will match all addresses where the three most significant bits in I2Cn_SADDR and the incoming address are equal.

If GCAMEN in I2Cn_CTRL is set, the general call address is always accepted regardless of the result of the address recognition. The start-byte, i.e. the general call address with the R/W bit set is ignored unless it is included in the defined slave address.

When an address is accepted by the address comparator, the decision of whether to ACK or NACK the address is passed to software.



15.3.9.3 Slave Transmitter

When SLAVE in I2Cn_CTRL is set, the RSTART interrupt flag in I2Cn_IF will be set when repeated START conditions are detected. After a START or repeated START condition, the bus master 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 slave 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 slave, and the slave 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 master wishes to read from the slave, the slave now goes into the slave transmitter mode. Software interaction is now required to decide whether the slave 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 slave responds with a command. The slave can reject the request with a single NACK command.

The slave will in that case go to an idle state, and wait for the next start condition. To continue the transmission, the slave must make sure data is loaded into the transmit buffer and send an ACK. The loaded data will then be transmitted to the master, and an ACK or NACK will be received from the master.

Data transmission can also continue after a NACK if a CONT command is issued along with the NACK. This is not standard I²C however.

If the master 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 master has received enough bytes and wishes to end the transmission. The slave now automatically goes idle, unless CONT in I2Cn_CMD is set and data is available for transmission. The latter is not standard I²C.

The master ends the transmission by sending a STOP or a repeated START. The SSTOP interrupt flag in I2Cn_IF is set when the master 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 I2Cn_IF will be set regardless of whether the slave is participating in the transmission or not, as long as SLAVE in I2Cn_CTRL is set and a STOP condition is detected

If arbitration is lost at any time during transmission, the ARBLOST interrupt flag in I2Cn_IF is set, the bus is released and the slave goes idle.

See Table 15.9 (p. 165) for more information.



Table 15.9. I²C Slave Transmitter

I2Cn_STAT	Description	I2Cn_IF	Required interaction	Response
0x41	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, slave 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 interrupt flag	None	The slave goes idle
	NACK received	(BUSHOLD interrupt flag)	CONT + TXDATA	DATA will be transmitted
-	Stop received	SSTOP interrupt flag	None	The slave goes idle
			START	START will be sent when bus becomes idle
-	Arbitration lost	ARBLOST interrupt	None	The slave goes idle
		flag	START	START will be sent when the bus becomes idle

15.3.9.4 Slave Receiver

A slave receiver operation is started in the same way as a slave transmitter operation, with the exception that the address transmitted by the master has the R/W bit cleared (W), indicating that the master wishes to write to the slave. The slave then goes into slave receiver mode.

To receive data from the master, the slave should respond to the address with an ACK and make sure space is available in the receive buffer. Transmission will then continue, and the slave will receive a byte from the master.

If a NACK is sent without a CONT, the transmission is ended for the slave, and it goes idle. If the slave issues both the NACK and CONT commands and has space available in the receive buffer, it will be open for continuing reception from the master.

When a byte has been received from the master, the slave must ACK or NACK the byte. The responses here are the same as for the reception of the address byte.

The master ends the transmission by sending a STOP or a repeated START. The SSTOP interrupt flag is set when the master transmits a STOP condition. If the transmission is ended with a repeated START, then the SSTOP interrupt flag in I2Cn_IF is not set.

Note

The SSTOP interrupt flag in I2Cn_IF will be set regardless of whether the slave is participating in the transmission or not, as long as SLAVE in I2Cn_CTRL is set and a STOP condition is detected

If arbitration is lost at any time during transmission, the ARBLOST interrupt flag in I2Cn_IF is set, the bus is released and the slave goes idle.



See Table 15.10 (p. 166) for more information.

Table 15.10. I²C - Slave Receiver

I2Cn_STAT	Description	I2Cn_IF	Required interaction	Response
-	Repeated START received	RSTART interrupt flag (BUSHOLD interrupt flag)	RXDATA	Receive and compare address
0x71	ADDR + W received	ADDR interrupt flag RXDATA interrupt flag	ACK + RXDATA	ACK will be sent and data will be received
		(BUSHOLD interrupt flag)	NACK	NACK will be sent, slave goes idle
			NACK + CONT + RXDATA	NACK will be sent and DATA will be received.
0xB1	Data received	RXDATA interrupt flag (BUSHOLD interrupt	ACK + RXDATA	ACK will be sent and data will be received
		flag)	NACK	NACK will be sent and slave will go idle
			NACK + CONT + RXDATA	NACK will be sent and data will be received
-	Stop received	SSTOP interrupt flag	None	The slave goes idle
			START	START will be sent when bus becomes idle
-	Arbitration lost	ARBLOST interrupt	None	The slave goes idle
		flag	START	START will be sent when the bus becomes idle

15.3.10 Transfer Automation

The I²C can be set up to complete transfers with a minimal amount of interaction.

15.3.10.1 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.

15.3.10.2 Automatic ACK

When AUTOACK in I2Cn_CTRL is set, an ACK is sent automatically whenever an ACK interaction is possible and no higher priority interactions are pending.

15.3.10.3 Automatic STOP

A STOP can be generated automatically on two conditions. These apply only to the master transmitter.

If AUTOSN in I2Cn_CTRL is set, the I²C module ends a transmission by transmitting a STOP condition when operating as a master transmitter and a NACK is received.

If AUTOSE in I2Cn_CTRL is set, the I²C 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 slave. 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.



15.3.11 Using 10-bit Addresses

When using 10-bit addresses in slave mode, set the I2Cn_SADDR register to 1111 0XX where XX are the two most significant bits of the 10-bit address, and set I2Cn_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 slave 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 slave, the slave should ACK the byte to continue the transmission, and if it does not match, the slave should NACK it.

When the master is operating as a master transmitter, the data bytes will follow after the second address byte. When the master is operating as a master 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 slave that found a match on the entire 10-bit address in the previous message should ACK this address. The repeated start should take the master into a master receiver mode, and after the single address byte sent this time around, the slave begins transmission to the master.

15.3.12 Error Handling

15.3.12.1 ABORT Command

Some bus errors may require software intervention to be resolved. The I²C module provides an ABORT command, which can be set in I2Cn_CMD, to help resolve bus errors.

When the bus for some reason is locked up and the I^2C module is in the middle of a transmission it cannot get out of, or for some other reason the I^2C wants to abort a transmission, the ABORT command can be used.

Setting the ABORT command will make the I²C 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 I²C module forget about any ongoing transfers.

15.3.12.2 Bus Reset

A bus reset can be performed by setting the START and STOP commands in I2Cn_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.

15.3.12.3 I²C-Bus Errors

An I²C-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 I²C-bus. If the I²C 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 I2Cn_IF is set to indicate the error, and the module automatically takes a course of action as defined in Table 15.11 (p. 167).

Table 15.11. I²C Bus Error Response

	Misplaced START	Misplaced STOP
In a master/slave operation	Treated as START. Receive address.	Go idle. Perform any pending actions.

15.3.12.4 Bus Lockup

A lockup occurs when a master or slave on the I²C-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 slave-only devices operating on an I²C-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 slaves 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 master should send 9 clock pulses on SCL while tristating the SDA. This procedure is performed in the GPIO module after clearing the I2C_ROUTE 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 slaves.

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 Section 15.3.12.6 (p. 168)

15.3.12.5 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 μ s before the bus is considered idle.

The bus idle timeout BITO in I2Cn_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 I2Cn_IF is set. The bus can also be set idle automatically on a bus idle timeout. This is enabled by setting GIBITO in I2Cn_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 I2Cn_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 I2Cn_STATUS is set. The timeout can be used to get the I²C module out of the busy-state it enters when reset, see Section 15.3.7.3 (p. 158).

15.3.12.6 Clock Low Timeout

The clock timeout, which can be configured in CLTO in I2Cn_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 I2Cn_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.

15.3.13 DMA Support

The I²C module has full DMA support. The DMA controller can write to the transmit buffer using the I2Cn_TXDATA register, and it can read from the receive buffer using the RXDATA register. A request for the DMA controller to read from the I²C receive buffer can come from the following source:

· Data available in the receive buffer

A write request can come from one of the following sources:



- · Transmit buffer and shift register empty. No data to send
- · Transmit buffer empty

15.3.14 Interrupts

The interrupts generated by the I²C module are combined into one interrupt vector, I2C_INT. If I²C interrupts are enabled, an interrupt will be made if one or more of the interrupt flags in I2Cn_IF and their corresponding bits in I2Cn_IEN are set.

15.3.15 Wake-up

The I²C receive section can be active all the way down to energy mode EM3, and can wake up the CPU on address interrupt. All address match modes are supported.



15.4 Register Map

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

Offset	Name	Туре	Description
0x000	I2Cn_CTRL	RW	Control Register
0x004	I2Cn_CMD	W1	Command Register
0x008	I2Cn_STATE	R	State Register
0x00C	I2Cn_STATUS	R	Status Register
0x010	I2Cn_CLKDIV	RW	Clock Division Register
0x014	I2Cn_SADDR	RW	Slave Address Register
0x018	I2Cn_SADDRMASK	RW	Slave Address Mask Register
0x01C	I2Cn_RXDATA	R	Receive Buffer Data Register
0x020	I2Cn_RXDATAP	R	Receive Buffer Data Peek Register
0x024	I2Cn_TXDATA	W	Transmit Buffer Data Register
0x028	I2Cn_IF	R	Interrupt Flag Register
0x02C	I2Cn_IFS	W1	Interrupt Flag Set Register
0x030	I2Cn_IFC	W1	Interrupt Flag Clear Register
0x034	I2Cn_IEN	RW	Interrupt Enable Register
0x038	I2Cn_ROUTE	RW	I/O Routing Register

15.5 Register Description

15.5.1 I2Cn_CTRL - Control Register

Offset		Bit Po	sition										
0x000	1 1 <th>18 17 91</th> <th>5 4</th> <th>13</th> <th>1 0</th> <th>o 8</th> <th>7</th> <th>2</th> <th>4</th> <th>3</th> <th>2</th> <th>-</th> <th>0</th>	18 17 91	5 4	13	1 0	o 8	7	2	4	3	2	-	0
Reset		0×0	0	0x0		0x0		0 0	0	0	0	0	0
Access		R W	XX XX	₩ W		RW		R ⊗ ⊗	R W	RW	RW	RW	RW
Name		СГТО	GIBITO	ВІТО		CLHR		GCAMEN	AUTOSN	AUTOSE		SLAVE	Z

Bit	Name	Reset	Access	Description
31:19	Reserved	To ensure con	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
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.

Value	Mode	Description
0	OFF	Timeout disabled
1	40PCC	Timeout after 40 prescaled clock cycles. In standard mode at 100 kHz, this results in a 50us timeout.
2	80PCC	Timeout after 80 prescaled clock cycles. In standard mode at 100 kHz, this results in a 100us timeout.
3	160PCC	Timeout after 160 prescaled clock cycles. In standard mode at 100 kHz, this results in a 200us timeout.
4	320PPC	Timeout after 320 prescaled clock cycles. In standard mode at 100 kHz, this results in a 400us timeout.
5	1024PPC	Timeout after 1024 prescaled clock cycles. In standard mode at 100 kHz, this results in a 1280us timeout.

15 GIBITO 0 RW **Go Idle on Bus Idle Timeout**



Bit	Name	Reset	Access	Description
	When set, the	bus automatically goes idle	on a bus idle tin	neout, allowing new transfers to be initiated.
	Value	Description		
	0	A bus idle timeout	has no effect on t	the bus state.
	1	A bus idle timeout	tells the I ² C modu	ule that the bus is idle, allowing new transfers to be initiated.
14	Reserved	To ensure co.	mpatibility with f	iuture devices, always write bits to 0. More information in Section 2.1 (p. 3)
13:12	BITO	0x0	RW	Bus Idle Timeout
	bus transactio by BITO, it set idle timeout is STOP conditio	n, i.e. the BUSY flag is set, is the BITO interrupt flag. The active as long as BUSY is	a timer is starte ne BITO interrup set. It is thus s	a given amount time between a START and STOP condition. When in a sed whenever SCL goes high. When the timer reaches the value defined at flag will then be set periodically as long as SCL remains high. The bus stopped automatically on a timeout if GIBITO is set. It is also stopped and is issued. The timeout is activated whenever the bus goes BUSY, i.e.
	Value	Mode	De	scription
	0	OFF	Tin	neout disabled
	1	40PCC		neout after 40 prescaled clock cycles. In standard mode at 100 kHz, this results in 50us timeout.
	2	80PCC		neout after 80 prescaled clock cycles. In standard mode at 100 kHz, this results in 00us timeout.
	3	160PCC		neout after 160 prescaled clock cycles. In standard mode at 100 kHz, this results a 200us timeout.
11:10	Reserved	To ensure co.	mpatibility with f	iuture devices, always write bits to 0. More information in Section 2.1 (p. 3)
9:8	CLHR	0x0	RW	Clock Low High Ratio
	_			e clock signal generated on SCL as master.
	Value	Mode		scription
	0	STANDARD		e ratio between low period and high period counters (N _{low} :N _{high}) is 4:4
	1	ASYMMETRIC		e ratio between low period and high period counters (N _{low} :N _{high}) is 6:3
	2	FAST	Th	e ratio between low period and high period counters (N _{low} :N _{high}) is 11:6
7	Reserved	To ensure co	mpatibility with f	iuture devices, always write bits to 0. More information in Section 2.1 (p. 3)
6	GCAMEN	0	RW	General Call Address Match Enable
	Set to enable a	address match on general ca	all in addition to	the programmed slave address.
	Value	Description		
	0		see will be NACK's	ed if it is not included by the slave address and address mask.
	1			vived, a software response is required.
	400010			
5	ARBDIS	0	RW	Arbitration Disable
	A master or sla	ave will not release the bus u	upon losing arbi	tration.
	Value	Description		
	0	When a device los	ses arbitration, the	ARB interrupt flag is set and the bus is released.
	1	When a device los	ses arbitration, the	ARB interrupt flag is set, but communication proceeds.
4	AUTOSN	0	RW	Automatic STOP on NACK
7		-		n a NACK is received from a slave.
	Value	Description		
	0	·	atically sent if a N	ACK is received from a slave.
	1	· ·		TOP if a NACK is received from a slave.
3	AUTOSE	0	RW	Automatic STOP when Empty
	Write to 1 to m	nake a master transmitter se	nd a STOP whe	n no more data is available for transmission.
	Value	Description		
	0	· ·	nt manually when	no more data is to be transmitted.
	1		-	TOP when no more data is available for transmission.
2	AUTOACK			
4	AUTUAUK	0	RW	Automatic Acknowledge

www.silabs.com

Set to enable automatic acknowledges.



Bit	Name	Reset	Access	Description
	Value	Description		
	0	Software must giv	e one ACK comma	and for each ACK transmitted on the I ² C bus.
	1	Addresses that are	e not automatically	NACK'ed, and all data is automatically acknowledged.
1	SLAVE	0	RW	Addressable as Slave
	Set this bit to allo	ow the device to be selected	ed as an I ² C slav	e.
	Value	Description		
	0	All addresses will	be responded to wi	ith a NACK
	1			d slave address or the general call address (if enabled) require a response from natically responded to with a NACK.
0	EN	0	RW	I ² C Enable
	Use this bit to er	able or disable the I ² C mo	dule.	
	Value	Description		
	0	The I ² C module is	disabled. And its in	nternal state is cleared
	1	The I ² C module is	enabled.	

15.5.2 I2Cn_CMD - Command Register

Offset															Bi	t Pc	siti	on														
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	ю	2	-	0
Reset																									0	0	0	0	0	0	0	0
Access																									W1	W1	×	W1	W	W	×	W
Name																									CLEARPC	CLEARTX	ABORT	CONT	NACK	ACK	STOP	START

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	CLEARPC	0	W1	Clear Pending Commands
	Set to clear pending	commands.		
6	CLEARTX	0	W1	Clear TX
	Set to clear transmit	buffer and shift regist	er. Will not abort	ongoing transfer.
5	ABORT	0	W1	Abort transmission
		•	•	en used in combination with STOP, a STOP condition is sent as soon as on is subject to clock synchronization.
4	CONT	0	W1	Continue transmission
	Set to continue trans	smission after a NAC	Chas been receiv	ed.
3	NACK	0	W1	Send NACK
	Set to transmit a NA	CK the next time an a	cknowledge is re	quired.
2	ACK	0	W1	Send ACK
	Set to transmit an A	CK the next time an a	cknowledge is red	quired.
1	STOP	0	W1	Send stop condition
	Set to send stop cor	ndition as soon as pos	sible.	
0	START	0	W1	Send start condition
	as the bus is idle. If t	the current transmission	on is owned by thi	ission is ongoing and not owned, the start condition will be sent as soon s module, a repeated start condition will be sent. Use in combination with TART when the bus becomes idle.



15.5.3 I2Cn_STATE - State Register

Offset															Bi	t Po	siti	on							·							
0x008	31	30	29	28	27	56	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	∞	7	9	2	4	က	7	-	0
Reset													•										•			0×0		0	0	0	0	-
Access																										ď		22	œ	~	~	~
Name																										STATE		BUSHOLD	NACKED	TRANSMITTER	MASTER	BUSY

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure c	ompatibility with	future devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:5	STATE	0x0	R	Transmission State
	The state of any	current transmission. Cl	eared if the I ² C i	module is idle.
	Value	Mode	D	Description
	0	IDLE	N	lo transmission is being performed.
	1	WAIT	V	Vaiting for idle. Will send a start condition as soon as the bus is idle.
	2	START	S	start transmitted or received
	3	ADDR	А	ddress transmitted or received
	4	ADDRACK	А	ddress ack/nack transmitted or received
	5	DATA	D	Pata transmitted or received
	6	DATAACK	D	ata ack/nack transmitted or received
4	BUSHOLD	0	R	Bus Held
	Set if the bus is o	currently being held by th	nis I ² C module.	
3	NACKED	0	R	Nack Received
	Set if a NACK wa	as received and STATE	is ADDRACK or	DATAACK.
2	TRANSMITTER	0	R	Transmitter
	•	ing as a master transmitt or the current mode is no		nsmitter. When cleared, the system may be operating as a master receiver,
1	MASTER	0	R	Master
	Set when operati	ing as an I ² C master. Wh	nen cleared, the	system may be operating as an I ² C slave.
0	BUSY	1	R	Bus Busy
		•		control of the bus or not has no effect on the value of this bit. When the vn, and thus BUSY is set. Use the ABORT command or a bus idle timeout

15.5.4 I2Cn_STATUS - Status Register

to force the I²C module out of the BUSY state.

Offset															Bi	t Po	siti	on														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	8	7	9	2	4	ю	7	-	0
Reset				•							•				,								,	0	-	0	0	0	0	0	0	0
Access																								œ	~	2	~	œ	œ	œ	~	22
Name																								RXDATAV	TXBL	TXC	PABORT	PCONT	PNACK	PACK	PSTOP	PSTART

Bit	Name	Reset	Access	Description
31:9	Reserved	To ensure con	npatibility with fut	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)



Bit	Name	Reset	Access	Description
8	RXDATAV	0	R	RX Data Valid
	Set when data is a	available in the receive b	uffer. Cleared wh	en the receive buffer is empty.
7	TXBL	1	R	TX Buffer Level
	Indicates the level	of the transmit buffer. S	et when the trans	smit buffer is empty, and cleared when it is full.
6	TXC	0	R	TX Complete
	Set when a transm	nission has completed an	d no more data is	available in the transmit buffer. Cleared when a new transmission starts.
5	PABORT	0	R	Pending abort
	An abort is pendin	g and will be transmitted	l as soon as poss	sible.
4	PCONT	0	R	Pending continue
	A continue is pend	ling and will be transmitt	ed as soon as po	ssible.
3	PNACK	0	R	Pending NACK
	A not-acknowledge	e is pending and will be	transmitted as so	on as possible.
2	PACK	0	R	Pending ACK
	An acknowledge is	s pending and will be tra	nsmitted as soon	as possible.
1	PSTOP	0	R	Pending STOP
	A stop condition is	pending and will be trar	nsmitted as soon	as possible.
0	PSTART	0	R	Pending START
	A start condition is	pending and will be tran	nsmitted as soon	as possible.

15.5.5 I2Cn_CLKDIV - Clock Division Register

Offset															Bi	t Po	ositi	on	-													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	1	10	တ	∞	7	9	2	4	က	2	-	0
Reset																												000x0				
Access																												W.				
Name																												ΛIQ				

Bit	Name	Reset	Access	Description
31:9	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
8:0	DIV	0x000	RW	Clock Divider
	Specifies the clock of	livider for the I ² C. Not	e that DIV must b	e 1 or higher when slave is enabled.

15.5.6 I2Cn_SADDR - Slave Address Register

Offset															Bi	t Po	siti	on														
0x014	31	30	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	∞	7	9	2	4	က	2	-	0
Reset																												0x0				
Access																												RW				
Name																												ADDR				



Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	mpatibility with f	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:1	ADDR	0x00	RW	Slave address
	Specifies the slave ac	ddress of the device.		
0	Reserved	To ensure co	mpatibility with f	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)

15.5.7 I2Cn_SADDRMASK - Slave Address Mask Register

Offset															Bi	t Pc	siti	on														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	œ	7	9	2	4	က	2	-	0
Reset																												0x00				
Access																												RW				
Name																												MASK				

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:1	MASK	0x00	RW	Slave Address Mask
	Specifies the significant bit match the exact address specifies		ess. Setting the	e mask to 0x00 will match all addresses, while setting it to 0x7F will only
0	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)

15.5.8 I2Cn_RXDATA - Receive Buffer Data Register

Offset															Bi	t Po	siti	on														
0x01C	31	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	∞	7	9	2	4	က	7	-	0
Reset																													00X0			
Access																												(Y			
Name																												į	KXDATA			

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	ompatibility with f	tuture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:0	RXDATA	0x00	R	RX Data
	Use this register to r	ead from the receive	buffer. Buffer is e	emptied on read access.



15.5.9 I2Cn_RXDATAP - Receive Buffer Data Peek Register

Offset														Bi	t Po	siti	on														
0x020	33	93	29	28	27	26	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	80	7	9	2	4	က	2	1	0
Reset																												0x0			
Access		-																										ď			
Name																												RXDATAP			

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure compa	ntibility with futu	re devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:0	RXDATAP	0x00	R	RX Data Peek
	Use this register to read from	m the receive buffe	r. Buffer is not	emptied on read access.

15.5.10 I2Cn_TXDATA - Transmit Buffer Data Register

Offset															Bi	t Po	siti	on														
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	=	10	6	8	7	9	2	4	က	2	-	0
Reset																													00X0			
Access																												:	>			
Name																												į	TXDATA			

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:0	TXDATA	0x00	W	TX Data
	Use this register to write a	byte to the transmit	buffer.	

15.5.11 I2Cn_IF - Interrupt Flag Register

Offset															Bi	t Pc	siti	on														
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	8	7	9	2	4	ю	2	-	0
Reset		,			•				•							0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0
Access																~	~	~	~	~	22	~	~	œ	~	~	~	~	~	~	~	~
Name																SSTOP	CLTO	ВІТО	RXUF	TXOF	BUSHOLD	BUSERR	ARBLOST	MSTOP	NACK	ACK	RXDATAV	TXBL	TXC	ADDR	RSTART	START

Bit	Name	Reset	Access	Description
31:17	Reserved	To ensure compa	atibility with futu	re devices, always write bits to 0. More information in Section 2.1 (p. 3)
16	SSTOP	0	R	Slave STOP condition Interrupt Flag
	Set when a STOP condition	has been received	d. Will be set re	gardless of the EFM32 being involved in the transaction or not.
15	CLTO	0	R	Clock Low Timeout Interrupt Flag



Bit	Name	Reset	Access	Description
	Set on each clock	low timeout. The timeou	ıt value can be se	t in CLTO bit field in the I2Cn_CTRL register.
14	ВІТО	0	R	Bus Idle Timeout Interrupt Flag
	Set on each bus id	dle timeout. The timeout	value can be set	in the BITO bit field in the I2Cn_CTRL register.
13	RXUF	0	R	Receive Buffer Underflow Interrupt Flag
	Set when data is r	ead from the receive bu	ffer through the I2	Cn_RXDATA register while the receive buffer is empty.
12	TXOF	0	R	Transmit Buffer Overflow Interrupt Flag
	Set when data is v	written to the transmit bu	ffer while the tran	smit buffer is full.
11	BUSHOLD	0	R	Bus Held Interrupt Flag
	Set when the bus	becomes held by the I ² C	C module.	
10	BUSERR	0	R	Bus Error Interrupt Flag
	Set when a bus er	ror is detected. The bus	error is resolved	automatically, but the current transfer is aborted.
9	ARBLOST	0	R	Arbitration Lost Interrupt Flag
	Set when arbitration	on is lost.		
8	MSTOP	0	R	Master STOP Condition Interrupt Flag
		condition has been suc nterrupt flag is not set.	ccessfully transmi	tted. If arbitration is lost during the transmission of the STOP condition,
7	NACK	0	R	Not Acknowledge Received Interrupt Flag
	Set when a NACK	has been received.		
6	ACK	0	R	Acknowledge Received Interrupt Flag
	Set when an ACK	has been received.		
5	RXDATAV	0	R	Receive Data Valid Interrupt Flag
	Set when data is a	available in the receive b	uffer. Cleared aut	tomatically when the receive buffer is read.
4	TXBL	1	R	Transmit Buffer Level Interrupt Flag
	Set when the trans	smit buffer becomes em	pty. Cleared autor	matically when new data is written to the transmit buffer.
3	TXC	0	R	Transfer Completed Interrupt Flag
	Set when the trans	smit shift register becom	es empty and the	re is no more data in the transmit buffer.
2	ADDR	0	R	Address Interrupt Flag
	Set when incomin	g address is accepted, i.	e. own address o	r general call address is received.
1	RSTART	0	R	Repeated START condition Interrupt Flag
	Set when a repeat	ted start condition is dete	ected.	
0	START	0	R	START condition Interrupt Flag
	Set when a start of	condition is successfully	transmitted.	

15.5.12 I2Cn_IFS - Interrupt Flag Set Register

Offset															Bi	t Po	siti	on														
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	0	∞	7	9	2	4	က	2	-	0
Reset																0	0	0	0	0	0	0	0	0	0	0			0	0	0	0
Access																×	×	×	M	×	×	×	×	W	W	×			W	W1	W	×
Name																SSTOP	CLTO	ВІТО	RXUF	TXOF	BUSHOLD	BUSERR	ARBLOST	MSTOP	NACK	ACK			TXC	ADDR	RSTART	START

Bit	Name	Reset	Access	Description
31:17	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)



Bit	Name	Reset	Access	Description
16	SSTOP	0	W1	Set SSTOP Interrupt Flag
	Write to 1 to set the	SSTOP interrupt flag.		
15	CLTO	0	W1	Set Clock Low Interrupt Flag
	Write to 1 to set the	e CLTO interrupt flag.		
14	BITO	0	W1	Set Bus Idle Timeout Interrupt Flag
	Write to 1 to set the	e BITO interrupt flag.		
13	RXUF	0	W1	Set Receive Buffer Underflow Interrupt Flag
	Write to 1 to set the	e RXUF interrupt flag.		
12	TXOF	0	W1	Set Transmit Buffer Overflow Interrupt Flag
	Write to 1 to set the	TXOF interrupt flag.		
11	BUSHOLD	0	W1	Set Bus Held Interrupt Flag
	Write to 1 to set the	BUSHOLD interrupt flag.		
10	BUSERR	0	W1	Set Bus Error Interrupt Flag
	Write to 1 to set the	BUSERR interrupt flag.		
9	ARBLOST	0	W1	Set Arbitration Lost Interrupt Flag
	Write to 1 to set the	e ARBLOST interrupt flag.		
8	MSTOP	0	W1	Set MSTOP Interrupt Flag
	Write to 1 to set the	MSTOP interrupt flag.		
7	NACK	0	W1	Set Not Acknowledge Received Interrupt Flag
	Write to 1 to set the	NACK interrupt flag.		
6	ACK	0	W1	Set Acknowledge Received Interrupt Flag
	Write to 1 to set the	e ACK interrupt flag.		
5:4	Reserved	To ensure compa	atibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
3	TXC	0	W1	Set Transfer Completed Interrupt Flag
	Write to 1 to set the	e TXC interrupt flag.		
2	ADDR	0	W1	Set Address Interrupt Flag
	Write to 1 to set the	e ADDR interrupt flag.		
1	RSTART	0	W1	Set Repeated START Interrupt Flag
	Write to 1 to set the	RSTART interrupt flag.		
0	START	0	W1	Set START Interrupt Flag
	Write to 1 to set the	START interrupt flag.		

15.5.13 I2Cn_IFC - Interrupt Flag Clear Register

Offset									,						Bi	t Po	siti	on					,	,								
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	8	7	9	2	4	8	2	-	0
Reset				•							•		•			0	0	0	0	0	0	0	0	0	0	0			0	0	0	0
Access																W	W	٧	W1	W	W	W 1	W1	W1	W	W1			W1	W	W1	W
Name																SSTOP	CLTO	BITO	RXUF	TXOF	BUSHOLD	BUSERR	ARBLOST	MSTOP	NACK	ACK			TXC	ADDR	RSTART	START

Bit	Name	Reset	Access	Description
31:17	Reserved	To ensure comp	atibility with fut	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
16	SSTOP	0	W1	Clear SSTOP Interrupt Flag



Name	Reset	Access	Description
			0.0.1.1.1.4.4.5
	-	VV1	Clear Clock Low Interrupt Flag
	-	W1	Clear Bus Idle Timeout Interrupt Flag
Write to 1 to clear	the BITO interrupt flag.		
RXUF	0	W1	Clear Receive Buffer Underflow Interrupt Flag
Write to 1 to clear	the RXUF interrupt flag.		
TXOF	0	W1	Clear Transmit Buffer Overflow Interrupt Flag
Write to 1 to clear	the TXOF interrupt flag.		
BUSHOLD	0	W1	Clear Bus Held Interrupt Flag
Write to 1 to clear	the BUSHOLD interrupt	flag.	
BUSERR	0	W1	Clear Bus Error Interrupt Flag
Write to 1 to clear	the BUSERR interrupt fla	ag.	
ARBLOST	0	W1	Clear Arbitration Lost Interrupt Flag
Write to 1 to clear	the ARBLOST interrupt f	lag.	
MSTOP	0	W1	Clear MSTOP Interrupt Flag
Write to 1 to clear	the MSTOP interrupt flag].	
NACK	0	W1	Clear Not Acknowledge Received Interrupt Flag
Write to 1 to clear	the NACK interrupt flag.		
ACK	0	W1	Clear Acknowledge Received Interrupt Flag
Write to 1 to clear	the ACK interrupt flag.		
Reserved		mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
TXC	0	W1	Clear Transfer Completed Interrupt Flag
Write to 1 to clear	the TXC interrupt flag.		, , ,
ADDR	0	W1	Clear Address Interrupt Flag
Write to 1 to clear	the ADDR interrupt flag.		
RSTART	0	W1	Clear Repeated START Interrupt Flag
Write to 1 to clear	the RSTART interrupt fla	ıg.	· · · · · · ·
START	0	W1	Clear START Interrupt Flag
	the START interrupt flag		
	CLTO Write to 1 to clear to BITO Write to 1 to clear to RXUF Write to 1 to clear to TXOF Write to 1 to clear to BUSHOLD Write to 1 to clear to BUSERR Write to 1 to clear to ARBLOST Write to 1 to clear to MSTOP Write to 1 to clear to ACK Write to 1 to clear to ACK Write to 1 to clear to ACK Write to 1 to clear to ACK Write to 1 to clear to ACK Write to 1 to clear to ADDR Write to 1 to clear to ADDR Write to 1 to clear to ADDR Write to 1 to clear to ADDR Write to 1 to clear to ADDR	Write to 1 to clear the SSTOP interrupt flag CLTO 0 Write to 1 to clear the CLTO interrupt flag. BITO 0 Write to 1 to clear the BITO interrupt flag. RXUF 0 Write to 1 to clear the RXUF interrupt flag. TXOF 0 Write to 1 to clear the TXOF interrupt flag. BUSHOLD 0 Write to 1 to clear the BUSHOLD interrupt BUSERR 0 Write to 1 to clear the BUSERR interrupt flag. ARBLOST 0 Write to 1 to clear the ARBLOST interrupt flag. MSTOP 0 Write to 1 to clear the MSTOP interrupt flag. NACK 0 Write to 1 to clear the NACK interrupt flag. ACK 0 Write to 1 to clear the ACK interrupt flag. Reserved To ensure con TXC 0 Write to 1 to clear the TXC interrupt flag. RDDR 0 Write to 1 to clear the ADDR interrupt flag. RSTART 0 Write to 1 to clear the RSTART interrupt flag.	Write to 1 to clear the SSTOP interrupt flag. CLTO 0 W1 Write to 1 to clear the CLTO interrupt flag. BITO 0 W1 Write to 1 to clear the BITO interrupt flag. RXUF 0 W1 Write to 1 to clear the RXUF interrupt flag. TXOF 0 W1 Write to 1 to clear the TXOF interrupt flag. BUSHOLD 0 W1 Write to 1 to clear the BUSHOLD interrupt flag. BUSERR 0 W1 Write to 1 to clear the BUSERR interrupt flag. ARBLOST 0 W1 Write to 1 to clear the ARBLOST interrupt flag. ARBLOST 0 W1 Write to 1 to clear the MSTOP interrupt flag. NACK 0 W1 Write to 1 to clear the NACK interrupt flag. ACK 0 W1 Write to 1 to clear the ACK interrupt flag. ACK 0 W1 Write to 1 to clear the ACK interrupt flag. Reserved To ensure compatibility with further to 1 to clear the TXC interrupt flag. RESERVED NACK 0 W1 Write to 1 to clear the TXC interrupt flag. RESERVED NACK NACK NACK NACK NACK NACK NACK NACK

15.5.14 I2Cn_IEN - Interrupt Enable Register

Offset															Bi	t Po	siti	on														
0x034	31	30	29	28	27	56	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	တ	∞	7	9	2	4	ю	2	-	0
Reset			•	•								•	•		•	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access																RW	RW W	R W	RW	R W	R ≪	W.	R W	R W	RW	RW	RW	RW	R ≪	RW	R W	RW
Name																SSTOP	СГТО	ВІТО	RXUF	TXOF	BUSHOLD	BUSERR	ARBLOST	MSTOP	NACK	ACK	RXDATAV	TXBL	TXC	ADDR	RSTART	START

Bit	Name	Reset	Access	Description
31:17	Reserved	To ensure compa	ntibility with futu	re devices, always write bits to 0. More information in Section 2.1 (p. 3)
16	SSTOP	0	RW	SSTOP Interrupt Enable
	Enable interrupt on SSTOP	•		



Bit	Name	Reset	Access	Description								
15	CLTO	0	RW	Clock Low Interrupt Enable								
	Enable interrupt on clock low timeout.											
14	ВІТО	0	RW	Bus Idle Timeout Interrupt Enable								
	Enable interrupt on bus idle timeout.											
13	RXUF	0	RW	Receive Buffer Underflow Interrupt Enable								
	Enable interrupt on receive buffer underflow.											
12	TXOF	0	RW	Transmit Buffer Overflow Interrupt Enable								
	Enable interrupt on transmit buffer overflow.											
1	BUSHOLD	0	RW	Bus Held Interrupt Enable								
	Enable interrupt on bus-held.											
10	BUSERR	0	RW	Bus Error Interrupt Enable								
	Enable interrupt on	bus error.										
9	ARBLOST	0	RW	Arbitration Lost Interrupt Enable								
	Enable interrupt on loss of arbitration.											
8	MSTOP	0	RW	MSTOP Interrupt Enable								
	Enable interrupt on MSTOP.											
7	NACK	0	RW	Not Acknowledge Received Interrupt Enable								
	Enable interrupt who	en not-acknowledge is	received.									
3	ACK	0	RW	Acknowledge Received Interrupt Enable								
	Enable interrupt on	acknowledge received.										
5	RXDATAV	0	RW	Receive Data Valid Interrupt Enable								
	Enable interrupt on	receive buffer full.										
ļ.	TXBL	0	RW	Transmit Buffer level Interrupt Enable								
	Enable interrupt on	transmit buffer level.										
3	TXC	0	RW	Transfer Completed Interrupt Enable								
	Enable interrupt on	transfer completed.										
2	ADDR	0	RW	Address Interrupt Enable								
	Enable interrupt on	recognized address.										
	RSTART	0	RW	Repeated START condition Interrupt Enable								
	Enable interrupt on	transmitted or received	repeated STAR	T condition.								
0	START	0	RW	START Condition Interrupt Enable								
	Enable interrupt on	transmitted or received	START condition	n.								

15.5.15 I2Cn_ROUTE - I/O Routing Register

Offset	Bit Position																															
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	∞	7	9	2	4	က	2	1	0
Reset				•						,		•											ç	e e							0	0
Access																							2	<u>}</u>							RW	RW
Name																							F	3							SCLPEN	SDAPEN

Bit	Name	Reset	Access	Description
31:10	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)

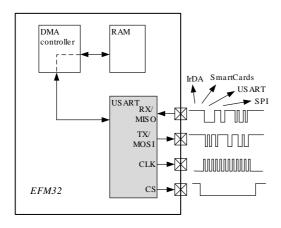


		,		·								
Bit	Name	Reset	Acces	s Description								
9:8	LOCATION	0x0	RW	I/O Location								
	Decides the loc	eation of the I ² C I/O pins.										
	Value	Mode		Description								
	0	LOC0		Location 0								
	1	LOC1		Location 1								
	2	LOC2		Location 2								
	3	LOC3		Location 3								
7:2	Reserved	To ensure con	mpatibility wit	h future devices, always write bits to 0. More information in Section 2.1 (p. 3)								
1	SCLPEN	0	RW	SCL Pin Enable								
	When set, the S	SCL pin of the I ² C is enabled	d.									
0	SDAPEN	0	RW	SDA Pin Enable								
	When set, the S	SDA pin of the I ² C is enabled	d.									



16 USART - Universal Synchronous Asynchronous Receiver/Transmitter





Quick Facts

What?

The USART handles high-speed UART, SPIbus, 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 multi-processor mode allows the USART to remain idle when not addressed. Triple buffering and DMA support makes high datarates possible with minimal CPU intervention and it is possible to transmit and receive large frames while the MCU remains in EM1.

16.1 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.

16.2 Features

- · Asynchronous and synchronous (SPI) communication
- Full duplex and half duplex
- Separate TX/RX enable
- Separate receive / transmit 2-level buffers, with additional separate shift registers
- Programmable baud rate, generated as an fractional division from the peripheral clock (HFPERCLK_{USARTn})
- Max bit-rate
 - SPI master mode, peripheral clock rate/2
 - SPI slave mode, peripheral clock rate/8
 - 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
 - · Master and slave mode
- Data can be transmitted LSB first or MSB first

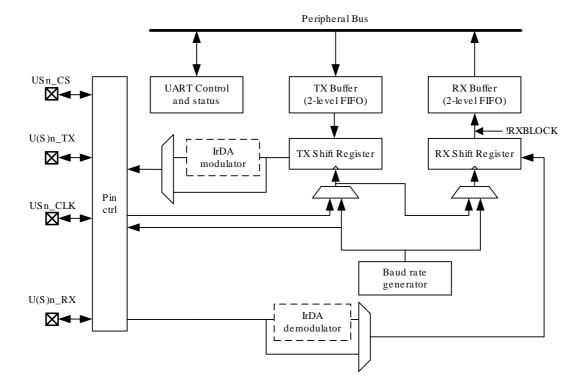


- Configurable number of data bits, 4-16 (plus the parity bit, if enabled)
 - HW parity bit generation and check
- Configurable number of stop bits in asynchronous mode: 0.5, 1, 1.5, 2
- · HW collision detection
- Multi-processor mode
- IrDA modulator on USART0
- SmartCard (ISO7816) mode
- · Separate interrupt vectors for receive and transmit interrupts
- · Loopback mode
 - Half duplex communication
 - Communication debugging

16.3 Functional Description

An overview of the USART module is shown in Figure 16.1 (p. 183).

Figure 16.1. USART Overview



16.3.1 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 bus master, and both the master and slave sample and transmit data according to this clock. Both master and slave 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 16.1 (p. 184). Full duplex and half duplex communication is supported in both asynchronous and synchronous mode.

Table 16.1. 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 16.2 (p. 184) 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 Section 16.3.2.5 (p. 192) and Section 16.3.3.3 (p. 200) respectively.

Table 16.2. USART Pin Usage

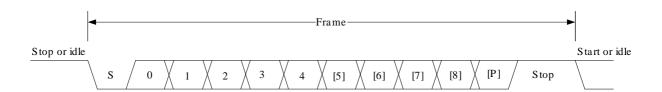
				Pin func	tionality				
SYNC	LOOPBK	MASTER	U(S)n_TX (MOSI)	U(S)n_RX (MISO)	USn_CLK	USn_CS			
0	0	х	Data out	Data in	-	[Driver enable]			
1	1	х	Data out/in	-	-	[Driver enable]			
1	0	0	Data in	Data out	Clock in	Slave select			
1	0	1	Data out	Data in	Clock out	[Auto slave select]			
1	1	0	Data out/in	-	Clock in	Slave select			
1	1	1	Data out/in	-	Clock out	[Auto slave select]			

16.3.2 Asynchronous Operation

16.3.2.1 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 16.2 (p. 184).

Figure 16.2. USART Asynchronous Frame Format



The number of data bits in a frame is set by DATABITS in USARTn_FRAME, see Table 16.3 (p. 185), and the number of stop-bits is set by STOPBITS in USARTn_FRAME, see Table 16.4 (p. 185). 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 16.3. 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 16.4. 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.

16.3.2.1.1 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 16.5 (p. 186). 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 16.5. USART Parity Bits

STOP BITS [1:0]	Description
00	No parity bit (Default)
01	Reserved
10	Even parity
11	Odd parity

16.3.2.2 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 Equation 16.1 (p. 186)

USART Baud Rate

$$br = f_{HFPERCLK}/(oversample \times (1 + USARTn_CLKDIV/256))$$
(16.1)

where $f_{HFPERCLK}$ is the peripheral clock (HFPERCLK_{USARTn}) frequency and oversample is the oversampling rate as defined by OVS in USARTn_CTRL, see Table 16.6 (p. 186).

Table 16.6. 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 15-bit value, with a 13-bit integral part and a 2-bit fractional part. The fractional part is configured in the two LSBs of DIV in USART_CLKDIV. The lowest achievable baud rate at 32 MHz is about 244 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 br_{desired}, the clock divider USARTn_CLKDIV can be calculated by using Equation 16.2 (p. 186) :

USART Desired Baud Rate
$$USARTn_CLKDIV = 256 \times (f_{HFPERCLK}/(oversample \times br_{desired}) - 1)$$
(16.2)

Table 16.7 (p. 187) 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 16.7. USART Baud Rates @ 4MHz Peripheral Clock

Desired	USART	Tn_OVS =00		USART		
baud rate [baud/s]	USARTn_CLKDIV/256	Actual baud rate [baud/s]	Error %	USARTn_CLKDIV/256	Actual baud rate [baud/s]	Error %
600	415,75	599,88	-0,02	832,25	600,06	0,01
1200	207,25	1200,48	0,04	415,75	1199,76	-0,02
2400	103,25	2398,082	-0,08	207,25	2400,96	0,04
4800	51	4807,692	0,16	103,25	4796,163	-0,08
9600	25	9615,385	0,16	51	9615,385	0,16
14400	16,25	14492,75	0,64	33,75	14388,49	-0,08
19200	12	19230,77	0,16	25	19230,77	0,16
28800	7,75	28571,43	-0,79	16,25	28985,51	0,64
38400	5,5	38461,54	0,16	12	38461,54	0,16
57600	3,25	58823,53	2,12	7,75	57142,86	-0,79
76800	2,25	76923,08	0,16	5,5	76923,08	0,16
115200	1,25	111111,1	-3,55	3,25	117647,1	2,12
230400	0	250000	8,51	1,25	222222,2	-3,55

16.3.2.3 Data Transmission

Asynchronous data transmission is initiated by writing data to the transmit buffer using one of the methods described in Section 16.3.2.3.1 (p. 187). 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 transmitter is idle. The TXC status flag is cleared when a new frame becomes available for transmission, but the TXC interrupt flag must be cleared by software.

16.3.2.3.1 Transmit Buffer Operation

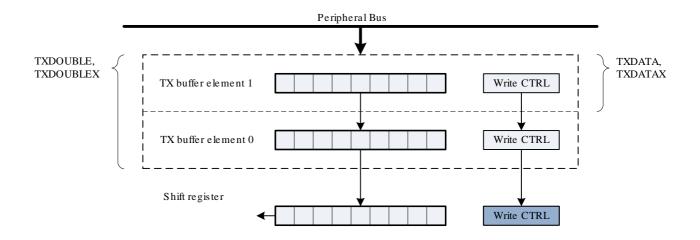
The transmit-buffer is a 2-level FIFO buffer. A frame can be loaded into the buffer by writing to USARTn_TXDATA, USARTn_TXDATAX, USARTn_TXDOUBLE or USARTn_TXDOUBLEX. Using USARTn_TXDATA allows 8 bits to be written to the buffer, while using USARTn_TXDOUBLE will write 2 frames of 8 bits to the buffer. 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 16.3 (p. 188) shows the basics of the transmit buffer when DATABITS in USARTn_FRAME is configured to less than 10 bits.

Figure 16.3. 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 transmitter is idle, 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.

16.3.2.3.2 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

16.3.2.4 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.

16.3.2.4.1 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.

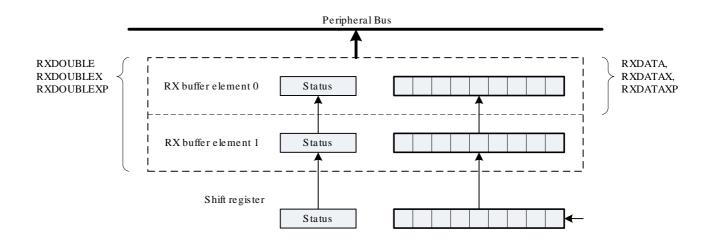
When a frame is read from the receive buffer using USARTn_RXDATA or USARTn_RXDATAX, the frame is pulled out of the buffer, making room for a new frame. USARTn_RXDOUBLE and USARTn_RXDOUBLEX pull two frames out of the buffer. If an attempt is done to read more frames from the buffer than what is available, 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 16.4 (p. 190).

Figure 16.4. 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.

16.3.2.4.2 Blocking Incoming Data

When using hardware frame recognition, as detailed in Section 16.3.2.8 (p. 196) and Section 16.3.2.9 (p. 197), 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 Section 16.3.2.8 (p. 196) 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.

16.3.2.4.3 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.



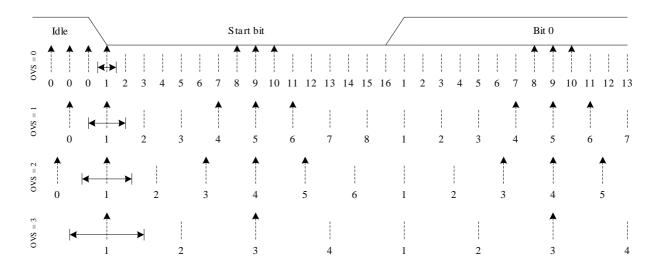
When a high-to-low transition is registered on the input while the receiver is idle, this is recognized as a start-bit, and the baud rate generator is synchronized with the incoming frame.

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 16.5 (p. 191). 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 16.5 (p. 191) .

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 16.5. USART Sampling of Start and Data Bits

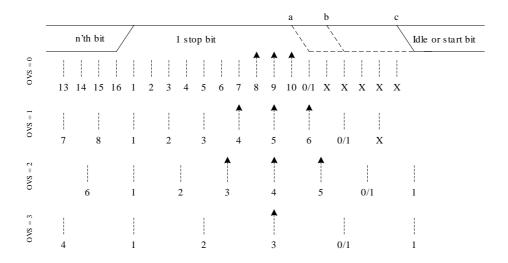


If the baud rate of the transmitter and receiver differ, the location each bit is sampled will be shifted towards the previous or next bit in the frame. This is acceptable for small errors in the baud rate, but for larger errors, it will result in transmission errors.

When the number of stop bits is 1 or more, stop bits are sampled like the start and data bits as seen in Figure 16.6 (p. 192). 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 16.6 (p. 192), 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 16.6. 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.

16.3.2.4.4 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.

16.3.2.4.5 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.

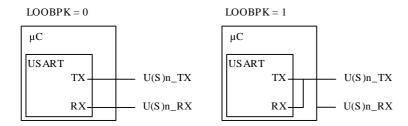
16.3.2.5 Local Loopback

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 16.7 (p. 193). 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 16.7. USART Local Loopback



16.3.2.6 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.

16.3.2.6.1 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 bus master, which first transmits a request to one of the bus slaves, 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 slave.

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.

16.3.2.6.2 Single Data-link with External Driver

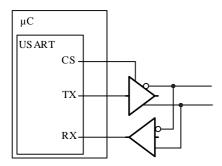
Some communication schemes, such as RS-485 rely on an external driver. Here, the driver has an extra input which enables it, and instead of tristating the transmitter when receiving data, the external driver must be disabled.



This can be done manually by assigning a GPIO to turn the driver on or off, or it can be handled automatically by the USART. If AUTOCS in USARTn_CTRL is set, the USn_CS output is automatically activated one baud period before the transmitter starts transmitting data, and deactivated when the last bit has been transmitted and there is no more data in the transmit buffer to transmit, or the transmitter becomes disabled. This feature can be used to turn the external driver on when transmitting data, and turn it off when the data has been transmitted.

Figure 16.8 (p. 194) shows an example configuration where USn_CS is used to automatically enable and disable an external driver.

Figure 16.8. 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/slave select when in synchronous mode (e.g. SPI).

16.3.2.6.3 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.

16.3.2.7 Large Frames

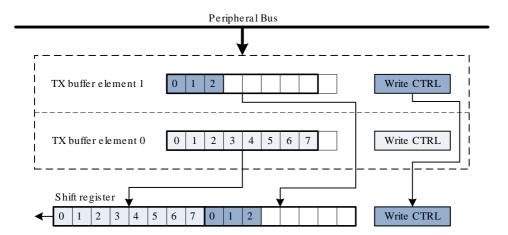
As each frame in the transmit and receive buffers holds a maximum of 9 bits, both the elements in the buffers are combined when working with USART-frames of 10 or more data bits.

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 16.9 (p. 195). The first element in the transmit buffer, i.e. element 0 in Figure 16.9 (p. 195) is the first element written to the FIFO, or the least significant byte when writing two bytes at a time using USARTn_TXDOUBLE.



Figure 16.9. USART Transmission of Large Frames



As shown in Figure 16.9 (p. 195), 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 16.10. USART Transmission of Large Frames, MSBF

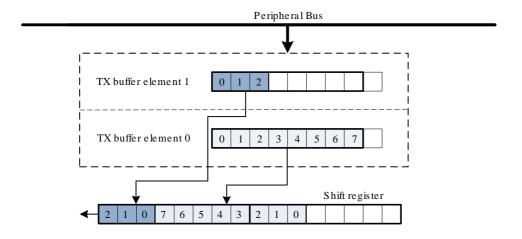


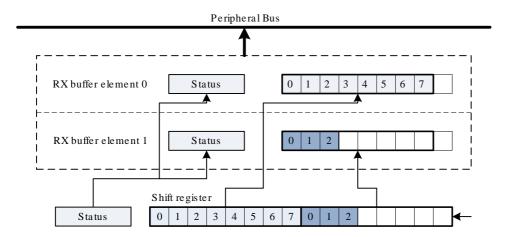
Figure 16.10 (p. 195) illustrates the order of the transmitted bits when an 11 bit frame is transmitted with MSBF set. If MSBF is set and the frame is smaller than 10 bits, only the contents of transmit buffer 0 will be transmitted.

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 16.11 (p. 196). 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 16.11. 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.

16.3.2.8 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.

Example 16.1 (p. 196) explains basic usage of the multi-processor mode:

Example 16.1. USART Multi-processor Mode Example

- 1. All slaves enable multi-processor mode and, enable and block the receiver. They will now not receive data unless it is an address frame. MPAB in USARTn_CTRL is set to identify frames with the 9th bit high as address frames.
- 2. The master sends a frame containing the address of a slave and with the 9th bit set.
- 3. All slaves receive the address frame and get an interrupt. They can read the address from the receive buffer. The selected slave unblocks the receiver to start receiving data from the master.
- 4. The master sends data with the 9th bit cleared.
- 5. Only the slave with RX enabled receives the data. When transmission is complete, the slave blocks the receiver and waits for a new address frame.

When a slave 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.

16.3.2.9 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.

16.3.2.10 SmartCard Mode

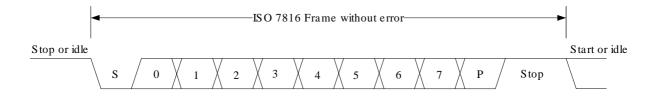
In SmartCard mode, the USART supports the ISO 7816 I/O line T0 mode. With exception of the stop-bits (guard time), the 7816 data frame is equal to the regular asynchronous frame. In this mode, the receiver pulls the line low for one baud, half a baud into the guard time to indicate a parity error. This 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 16.12 (p. 197) . 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.

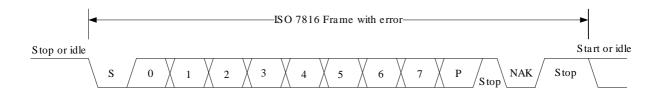
Figure 16.12. USART ISO 7816 Data Frame Without Error



If a parity error is detected by the receiver, it pulls the line I/O line low after half a stop bit, see Figure 16.13 (p. 198). 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 16.13. USART ISO 7816 Data Frame With Error



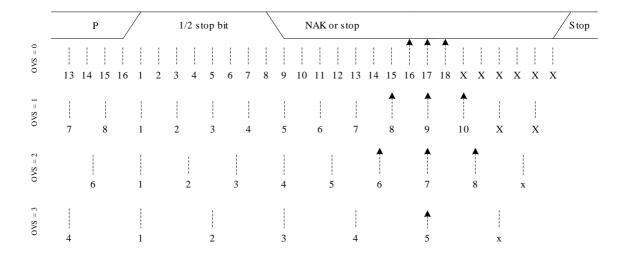
On a parity error, the NAK is generated by hardware. The NAK generated by the receiver is sampled as the stop-bit of the frame. Because of this, parity errors when in SmartCard mode are reported with both a parity error and a framing error.

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 16.14 (p. 198) . 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 16.14. 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.

16.3.3 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.



16.3.3.1 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.

16.3.3.2 Clock Generation

The bit-rate in synchronous mode is given by Equation 16.3 (p. 199). As in the case of asynchronous operation, the clock division factor have a 13-bit integral part and a 2-bit fractional part.

$$br = f_{HFPERCLK}/(2 \times (1 + USARTn_CLKDIV/256))$$
 (16.3)

Given a desired baud rate brdesired, the clock divider USARTn_CLKDIV can be calculated using Equation 16.4 (p. 199)

USART Synchronous Mode Clock Division Factor

$$USARTn_CLKDIV = 256 \times (f_{HFPERCLK}/(2 \times brdesired) - 1)$$
 (16.4)

When the USART operates in master mode, the highest possible bit rate is half the peripheral clock rate. When operating in slave mode however, the highest bit rate is an eighth of the peripheral clock:

Master mode: br_{max} = f_{HFPERCLK}/2
 Slave mode: br_{max} = f_{HFPERCLK}/8

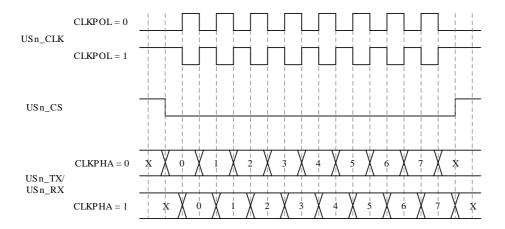
On every clock edge data on the data lines, MOSI and MISO, is either set up or sampled. When CLKPHA in USARTn_CTRL is cleared, data is sampled on the leading clock edge and set-up is done on the trailing edge. If CLKPHA is set however, data is set-up on the leading clock edge, and sampled on the trailing 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 16.8 (p. 199). Figure 16.15 (p. 200) shows the resulting timing of data set-up and sampling relative to the bus clock.

Table 16.8. 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 16.15. USART SPI Timing



If CPHA=1, the TX underflow flag, TXUF, will be set on the first setup clock edge of a frame in slave mode if TX data is not available. If CPHA=0, TXUF is set if data is not available in the transmit buffer three HFPERCLK 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.

16.3.3.3 Master Mode

When in master mode, the USART is in full control of the data flow on the synchronous bus. When operating in full duplex mode, the slave cannot transmit data to the master without the master transmitting to the slave. The master 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 master 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.

16.3.3.3.1 Operation of USn_CS Pin

When operating in master mode, 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 slave by setting AUTOCS in USARTn_CTRL. If AUTOCS is set, USn_CS is activated when a transmission begins, and deactivated directly after the last bit has been transmitted and there is no more data in the transmit buffer. 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 master 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 slave mode.

16.3.3.4 Slave Mode

When the USART is in slave mode, data transmission is not controlled by the USART, but by an external master. The USART is therefore not able to initiate a transmission, and has no control over the number of bytes written to the master.



The output and input to the USART are also swapped when in slave mode, making the receiver take its input from USn_TX (MOSI) and the transmitter drive USn_RX (MISO).

To transmit data when in slave mode, the slave must load data into the transmit buffer and enable the transmitter. The data will remain in the USART until the master starts a transmission by pulling the USn_CS input of the slave low and transmitting data. For every frame the master transmits to the slave, a frame is transferred from the slave to the master. After a transmission, MISO remains in the same state as the last bit transmitted. This also applies if the master transmits to the slave and the slave TX buffer is empty.

If the transmitter is enabled in synchronous slave mode and the master starts transmission of a frame, the underflow interrupt flag TXUF in USARTn_IF will be set if no data is available for transmission to the master.

If the slave needs to control its own chip select signal, this can be achieved by clearing CSPEN in the ROUTE 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.

16.3.3.5 Synchronous Half Duplex Communication

Half duplex communication in synchronous mode is very similar to half duplex communication in asynchronous mode as detailed in Section 16.3.2.6 (p. 193). The main difference is that in this mode, the master must generate the bus clock even when it is not transmitting data, i.e. it must provide the slave with a clock to receive data. To generate the bus clock, the master should transmit data with the transmitter tristated, i.e. TXTRI in USARTn_STATUS set, when receiving data. If 2 bytes are expected from the slave, then transmit 2 bytes with the transmitter tristated, and the slave uses the generated bus clock to transmit data to the master. TXTRI can be set by setting the TXTRIEN command bit in USARTn_CMD.

Note

When operating as SPI slave in half duplex mode, TX has to be tristated (not disabled) during data reception if the slave is to transmit data in the current transfer.

16.3.4 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 USARTn_TRIGCTRL. 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.

16.3.5 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.

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.

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.

16.3.6 Transmission Delay

By configuring TXDELAY in USARTn_CTRL, the transmitter can be forced to wait a number of bitperiods from it is ready to transmit data, to 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 baud 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.

TXDELAY in USARTn_CTRL only applies to asynchronous transmission.

16.3.7 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

The reception interrupt on the other hand groups the reception-related interrupts, triggered by the following interrupt flags:

- RXDATAV
- RXFULL
- RXOF
- RXUF
- PERR
- FERRMPAF
- SSM

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.

16.3.8 IrDA Modulator/Demodulator

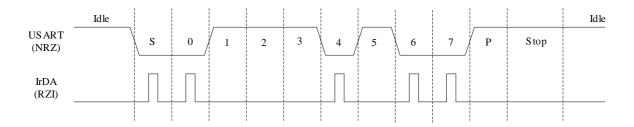
The IrDA modulator on USARTO implements the physical layer of the IrDA specification, which is necessary for communication over IrDA. The modulator takes the signal output from the USART module,



and modulates it before it leaves USART0 . In the same way, the input signal is demodulated before it enters the actual USART module. The modulator is only available on USART0 , and 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 16.16 (p. 203) .

Figure 16.16. 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 16.9 (p. 203).

Table 16.9. 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.

The IrDA module can also modulate a signal from the PRS system, and transmit a modulated signal to the PRS system. To use a PRS channel as transmitter source instead of the USART, set IRPRSEN in USARTn_IRCTRL high. The channel is selected by configuring IRPRSSEL in USARTn_IRCTRL.



16.4 Register Map

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

Offset	Name	Туре	Description
0x000	USARTn_CTRL	RW	Control Register
0x004	USARTn_FRAME	RW	USART Frame Format Register
0x008	USARTn_TRIGCTRL	RW	USART Trigger Control register
0x00C	USARTn_CMD	W1	Command Register
0x010	USARTn_STATUS	R	USART Status Register
0x014	USARTn_CLKDIV	RW	Clock Control Register
0x018	USARTn_RXDATAX	R	RX Buffer Data Extended Register
0x01C	USARTn_RXDATA	R	RX Buffer Data Register
0x020	USARTn_RXDOUBLEX	R	RX Buffer Double Data Extended Register
0x024	USARTn_RXDOUBLE	R	RX FIFO Double Data Register
0x028	USARTn_RXDATAXP	R	RX Buffer Data Extended Peek Register
0x02C	USARTn_RXDOUBLEXP	R	RX Buffer Double Data Extended Peek Register
0x030	USARTn_TXDATAX	W	TX Buffer Data Extended Register
0x034	USARTn_TXDATA	W	TX Buffer Data Register
0x038	USARTn_TXDOUBLEX	W	TX Buffer Double Data Extended Register
0x03C	USARTn_TXDOUBLE	W	TX Buffer Double Data Register
0x040	USARTn_IF	R	Interrupt Flag Register
0x044	USARTn_IFS	W1	Interrupt Flag Set Register
0x048	USARTn_IFC	W1	Interrupt Flag Clear Register
0x04C	USARTn_IEN	RW	Interrupt Enable Register
0x050	USARTn_IRCTRL	RW	IrDA Control Register
0x054	USARTn_ROUTE	RW	I/O Routing Register

16.5 Register Description

16.5.1 USARTn_CTRL - Control Register

Byte order swapped

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	8	7	9	2	4	3	2	-	0
Reset				0	OXO			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		000		0	0	0	0	0
Access				W.	NA NA			W.	W.	RW W	RW	RW	RW	RW	W.	RW	RW	W.	RW	RW	W.	W.	RW	RW		X N		RW	RW	RW	RW	RW
Name				BYTESWAP	TXDELAY			ERRSTX	ERRSRX	ERRSDMA	BIT8DV	SKIPPERRF	SCRETRANS	SCMODE	AUTOTRI	AUTOCS	CSINV	VNIXT	RXINV	TXBIL	CSMA	MSBF	CLKPHA	CLKPOL		SAO)	MPAB	MPM	CCEN	LOOPBK	SYNC

			0) 0)												
Bit	Name	Reset	Access	Description											
31:29	Reserved	To ensure con	npatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)											
28	BYTESWAP	0	0 RW Byteswap In Double Accesses												
	Set to switch the or	der of the bytes in doubl	e accesses.												
	Value	Description													
	0	Normal byte order	Normal byte order												



	Name		Reset	Acces	ss Description								
27:26	TXDELAY		0x0	RW	TX Delay Transmission								
	Configurable de	elay before	new transfers. F	rames sent	back-to-back are not delayed.								
	Value	Mode			Description								
	0	NONE			Frames are transmitted immediately								
	1	SINGLE			Transmission of new frames are delayed by a single baud period								
	2	DOUBL	E		Transmission of new frames are delayed by two baud periods								
	3	TRIPLE			Transmission of new frames are delayed by three baud periods								
25	Reserved		To ensure co	mpatibility w	ith future devices, always write bits to 0. More information in Section 2.1 (p. 3								
24	ERRSTX		0	RW	Disable TX On Error								
	When set, the t	ransmitter	is disabled on fra	aming and pa	arity errors (asynchronous mode only) in the receiver.								
	Value		Description										
	0		Received framing	and parity err	ors have no effect on transmitter								
	1		Received framing	and parity err	ors disable the transmitter								
23	ERRSRX		0	RW	Disable RX On Error								
	When set, the r	eceiver is	disabled on fram	ing and parit	y errors (asynchronous mode only).								
	Value		Τ		, , ,								
	0		Description	v orrore have	no offset on receiver								
	1		Framing and parity errors have no effect on receiver Framing and parity errors disable the receiver										
22	ERRSDMA		0	RW	Halt DMA On Error								
	When set, DMA	A requests	will be cleared o	n framing an	d parity errors (asynchronous mode only).								
	Value		Description										
	0		Framing and parit	ty errors have	no effect on DMA requests from the USART								
	1		DMA requests fro	m the USART	are blocked while the PERR or FERR interrupt flags are set								
					are stocked thine tile i Eritt or i Eritt interrupt hage are set								
21	BIT8DV		0	RW	Bit 8 Default Value								
21	The default val		9th bit. If 9-bit fra	RW	Bit 8 Default Value								
	The default val		9th bit. If 9-bit fra of BIT8DV.	RW ames are use	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the								
21	The default val 9th bit is set to SKIPPERRF	the value o	Oth bit. If 9-bit fra of BIT8DV.	RW ames are uso RW	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames								
20	The default val 9th bit is set to SKIPPERRF When set, the r	the value o	Oth bit. If 9-bit fra of BIT8DV.	RW ames are uso RW	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the								
	The default val 9th bit is set to SKIPPERRF	the value o	Oth bit. If 9-bit fra of BIT8DV.	RW ames are uso RW	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames								
20	The default val 9th bit is set to SKIPPERRF When set, the r	the value of	Oth bit. If 9-bit fra of BIT8DV. 0 scards frames wi	RW ames are use RW th parity erro	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames rs (asynchronous mode only). The PERR interrupt flag is still set.								
20	The default val 9th bit is set to SKIPPERRF When set, the r	the value of	Oth bit. If 9-bit fra of BIT8DV. 0 scards frames wi	RW ames are use RW th parity erro	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames rs (asynchronous mode only). The PERR interrupt flag is still set. SmartCard Retransmit								
20	The default val 9th bit is set to SKIPPERRF When set, the r SCRETRANS When in Smarte SCMODE	the value of the v	Oth bit. If 9-bit fra of BIT8DV. 0 scards frames with 0 s, a NACK'ed frames	RW ames are use RW th parity erro RW me will be ke	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames rs (asynchronous mode only). The PERR interrupt flag is still set. SmartCard Retransmit ept in the shift register and retransmitted if the transmitter is still enabled.								
20	The default val 9th bit is set to SKIPPERRF When set, the r SCRETRANS When in Smarte SCMODE	the value of the v	oth bit. If 9-bit fra of BIT8DV. 0 scards frames wi 0 e, a NACK'ed fran	RW ames are use RW th parity erro RW me will be ke	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames rs (asynchronous mode only). The PERR interrupt flag is still set. SmartCard Retransmit ept in the shift register and retransmitted if the transmitter is still enabled.								
20 19 18	The default val 9th bit is set to SKIPPERRF When set, the r SCRETRANS When in Smarte SCMODE Use this bit to e	eceiver dis	oth bit. If 9-bit fra of BIT8DV. 0 scards frames wi 0 e, a NACK'ed fram 0 sisable SmartCare	RW ames are use RW th parity erro RW me will be ke RW d mode. RW	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames rs (asynchronous mode only). The PERR interrupt flag is still set. SmartCard Retransmit ept in the shift register and retransmitted if the transmitter is still enabled. SmartCard Mode Automatic TX Tristate								
20 19 18	The default val 9th bit is set to SKIPPERRF When set, the r SCRETRANS When in Smarth SCMODE Use this bit to e AUTOTRI When enabled,	eceiver dis	oth bit. If 9-bit fra of BIT8DV. 0 scards frames wi 0 e, a NACK'ed fram 0 stable SmartCard 0 et by hardware wi	RW ames are use RW th parity erro RW me will be ke RW d mode. RW	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames rs (asynchronous mode only). The PERR interrupt flag is still set. SmartCard Retransmit ept in the shift register and retransmitted if the transmitter is still enabled. SmartCard Mode Automatic TX Tristate								
20 19 18	The default val 9th bit is set to SKIPPERRF When set, the r SCRETRANS When in Smartt SCMODE Use this bit to e AUTOTRI When enabled, Value	eceiver dis	oth bit. If 9-bit fra of BIT8DV. 0 scards frames wi 0 e, a NACK'ed fram 0 sisable SmartCare 0 et by hardware w	RW ames are use RW th parity erro RW me will be ke RW d mode. RW	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames rs (asynchronous mode only). The PERR interrupt flag is still set. SmartCard Retransmit ept in the shift register and retransmitted if the transmitter is still enabled. SmartCard Mode Automatic TX Tristate transmitter is idle, and TXTRI is cleared by hardware when transmission starts								
20 19 18	The default val 9th bit is set to SKIPPERRF When set, the r SCRETRANS When in Smarth SCMODE Use this bit to e AUTOTRI When enabled, Value 0	eceiver dis	oth bit. If 9-bit fra of BIT8DV. 0 scards frames wi 0 e, a NACK'ed fram 0 isable SmartCare 0 et by hardware w Description The output on U(s	RW ames are use RW th parity erro RW me will be ke RW d mode. RW henever the	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames rs (asynchronous mode only). The PERR interrupt flag is still set. SmartCard Retransmit ept in the shift register and retransmitted if the transmitter is still enabled. SmartCard Mode Automatic TX Tristate transmitter is idle, and TXTRI is cleared by hardware when transmission starts								
20 19 18 17	The default val 9th bit is set to SKIPPERRF When set, the r SCRETRANS When in Smarte SCMODE Use this bit to e AUTOTRI When enabled, Value 0 1	eceiver dis	oth bit. If 9-bit fra of BIT8DV. 0 scards frames wi 0 e, a NACK'ed fram 0 sisable SmartCard 0 et by hardware w Description The output on U(\$\text{U}(S)n_TX\$ is trista	RW ames are use RW th parity erro RW me will be ke RW d mode. RW thenever the	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames rs (asynchronous mode only). The PERR interrupt flag is still set. SmartCard Retransmit ept in the shift register and retransmitted if the transmitter is still enabled. SmartCard Mode Automatic TX Tristate transmitter is idle, and TXTRI is cleared by hardware when transmission starts the transmitter is idle is defined by TXINV the transmitter is idle								
20 19 18	The default val 9th bit is set to SKIPPERRF When set, the r SCRETRANS When in Smarth SCMODE Use this bit to e AUTOTRI When enabled, Value 0 1 AUTOCS	ceceiver discernable or d	oth bit. If 9-bit fra of BIT8DV. 0 scards frames wi 0 e, a NACK'ed fram 0 isable SmartCard 0 et by hardware w Description The output on U(S)n_TX is trista	RW ames are use RW th parity erro RW me will be ke RW d mode. RW thenever the	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames rs (asynchronous mode only). The PERR interrupt flag is still set. SmartCard Retransmit ept in the shift register and retransmitted if the transmitter is still enabled. SmartCard Mode Automatic TX Tristate transmitter is idle, and TXTRI is cleared by hardware when transmission starts the transmitter is idle is defined by TXINV the transmitter is idle Automatic Chip Select								
20 19 18 17	The default val 9th bit is set to SKIPPERRF When set, the r SCRETRANS When in Smarth SCMODE Use this bit to e AUTOTRI When enabled, Value 0 1 AUTOCS	the value of receiver discrete	oth bit. If 9-bit fra of BIT8DV. 0 scards frames wi 0 e, a NACK'ed fram 0 isable SmartCard 0 et by hardware w Description The output on U(S)n_TX is trista	RW ames are use RW th parity erro RW me will be ke RW d mode. RW thenever the	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames rs (asynchronous mode only). The PERR interrupt flag is still set. SmartCard Retransmit ept in the shift register and retransmitted if the transmitter is still enabled. SmartCard Mode Automatic TX Tristate transmitter is idle, and TXTRI is cleared by hardware when transmission starts the transmitter is idle is defined by TXINV the transmitter is idle Automatic Chip Select								
20 19 18 17	The default val 9th bit is set to SKIPPERRF When set, the r SCRETRANS When in Smarth SCMODE Use this bit to e AUTOTRI When enabled, Value 0 1 AUTOCS When enabled	the value of receiver discrete	oth bit. If 9-bit fra of BIT8DV. 0 scards frames wi 0 e, a NACK'ed fram 0 isable SmartCard 0 et by hardware w Description The output on U(S)n_TX is trista	RW ames are use RW th parity erro RW me will be ke RW d mode. RW thenever the	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames rs (asynchronous mode only). The PERR interrupt flag is still set. SmartCard Retransmit ept in the shift register and retransmitted if the transmitter is still enabled. SmartCard Mode Automatic TX Tristate transmitter is idle, and TXTRI is cleared by hardware when transmission starts the transmitter is idle is defined by TXINV the transmitter is idle Automatic Chip Select								
20 19 18 17	The default val 9th bit is set to SKIPPERRF When set, the r SCRETRANS When in Smarth SCMODE Use this bit to e AUTOTRI When enabled, Value 0 1 AUTOCS When enabled transmission er CSINV	the value of receiver discrete was considered and the control of t	oth bit. If 9-bit fra of BIT8DV. 0 scards frames wi 0 e, a NACK'ed fram 0 stable SmartCard 0 et by hardware w Description The output on U(s) U(S)n_TX is trista 0 ut on USn_CS	RW ames are use RW th parity erro RW me will be ke RW d mode. RW thenever the Sin_TX when to sited whenever RW will be active RW	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames rs (asynchronous mode only). The PERR interrupt flag is still set. SmartCard Retransmit ept in the shift register and retransmitted if the transmitter is still enabled. SmartCard Mode Automatic TX Tristate transmitter is idle, and TXTRI is cleared by hardware when transmission starts the transmitter is idle is defined by TXINV the transmitter is idle Automatic Chip Select vated one baud-period before transmission starts, and deactivated when Chip Select Invert								
20 19 18 17	The default val 9th bit is set to SKIPPERRF When set, the r SCRETRANS When in Smarth SCMODE Use this bit to e AUTOTRI When enabled, Value 0 1 AUTOCS When enabled transmission er CSINV	the value of receiver discrete was considered and the control of t	oth bit. If 9-bit fra of BIT8DV. 0 scards frames wi 0 e, a NACK'ed fram 0 stable SmartCard 0 et by hardware w Description The output on U(s) U(S)n_TX is trista 0 ut on USn_CS	RW ames are use RW th parity erro RW me will be ke RW d mode. RW thenever the Sin_TX when to sited whenever RW will be active RW	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames rs (asynchronous mode only). The PERR interrupt flag is still set. SmartCard Retransmit ept in the shift register and retransmitted if the transmitter is still enabled. SmartCard Mode Automatic TX Tristate transmitter is idle, and TXTRI is cleared by hardware when transmission starts the transmitter is idle is defined by TXINV the transmitter is idle Automatic Chip Select vated one baud-period before transmission starts, and deactivated when Chip Select Invert								
20 19 18 17	The default val 9th bit is set to SKIPPERRF When set, the r SCRETRANS When in Smartt SCMODE Use this bit to e AUTOTRI When enabled, Value 0 1 AUTOCS When enabled transmission er CSINV Default value is	the value of receiver discrete was considered and the control of t	oth bit. If 9-bit fra of BIT8DV. 0 scards frames wi 0 s, a NACK'ed fram 0 sisable SmartCard 0 et by hardware w Description The output on U(s) U(S)n_TX is trista 0 ut on USn_CS 0 . This affects bot	RW ames are use RW th parity erro RW me will be ke RW d mode. RW thenever the selection RW will be active	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames rs (asynchronous mode only). The PERR interrupt flag is still set. SmartCard Retransmit ept in the shift register and retransmitted if the transmitter is still enabled. SmartCard Mode Automatic TX Tristate transmitter is idle, and TXTRI is cleared by hardware when transmission starts the transmitter is idle is defined by TXINV the transmitter is idle Automatic Chip Select vated one baud-period before transmission starts, and deactivated wher								
20 19 18 17	The default val 9th bit is set to SKIPPERRF When set, the r SCRETRANS When in Smarth SCMODE Use this bit to e AUTOTRI When enabled, Value 0 1 AUTOCS When enabled transmission er CSINV Default value is Value	the value of receiver discrete was considered and the control of t	oth bit. If 9-bit fra of BIT8DV. 0 scards frames wi 0 e, a NACK'ed fram 0 stable SmartCard 0 et by hardware w Description The output on U(s) U(S)n_TX is tristate 0 ut on USn_CS 0 . This affects bote	RW ames are use RW th parity erro RW me will be ke RW d mode. RW henever the S)n_TX when to sted whenever RW will be activ RW h the selection	Bit 8 Default Value ed, and an 8-bit write operation is done, leaving the 9th bit unspecified, the Skip Parity Error Frames rs (asynchronous mode only). The PERR interrupt flag is still set. SmartCard Retransmit ept in the shift register and retransmitted if the transmitter is still enabled. SmartCard Mode Automatic TX Tristate transmitter is idle, and TXTRI is cleared by hardware when transmission starts the transmitter is idle is defined by TXINV the transmitter is idle Automatic Chip Select vated one baud-period before transmission starts, and deactivated wher Chip Select Invert								

The output from the USART transmitter can optionally be inverted by setting this bit.



	Name	Reset	Acces	ss Description
	Value	Description		
	0	Output from the	ransmitter is pa	assed unchanged to U(S)n_TX
	1	Output from the t	ransmitter is in	verted before it is passed to U(S)n_TX
3	RXINV	0	RW	Receiver Input Invert
	Setting this bi	t will invert the input to the U	JSART receiv	ver.
	Value	Description		
	0	Input is passed of	lirectly to the re	eceiver
	1		-	sed to the receiver
2	TXBIL	0	RW	
2		-		TX Buffer Interrupt Level
	Determines th	ne interrupt and status level	of the transm	it buffer.
	Value	Mode		Description
	0	EMPTY		TXBL and the TXBL interrupt flag are set when the transmit buffer becomes empt 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 empt TXBL is cleared when the buffer becomes full.
1	CSMA	0	RW	Action On Slave-Select In Master Mode
	This register of	letermines the action to be p	erformed wh	en slave-select is configured as an input and driven low while in master mod
	Value	Mode		Description
	0	NOACTION		No action taken
	1	GOTOSLAVEMODE		Go to slave mode
0	MSBF	0	RW	Most Significant Bit First
	Value 0 1	Description Data is sent with Data is sent with		
	CLKPHA	0	RW	Clock Edge For Setup/Sample
		-		ng to the bus clock when in synchronous mode.
		-		· · · · · · · · · · · · · · · · · · ·
	Value 0	Mode		Description
	U	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 closin synchronous mode
	CLKPOL	0	RW	Clock Polarity
	Determines th	ne clock polarity of the bus c	lock used in s	synchronous mode.
	Value	Mode		Description
		111000		2000.19.1011
	0	IDLELOW		The bus clock used in synchronous mode has a low base value
		IDLELOW IDLEHIGH		The bus clock used in synchronous mode has a low base value The bus clock used in synchronous mode has a high base value
	0	IDLEHIGH	ompatibility w	The bus clock used in synchronous mode has a high base value
	0	IDLEHIGH	ompatibility w	The bus clock used in synchronous mode has a high base value
	0 1 Reserved	To ensure co	RW	The bus clock used in synchronous mode has a high base value ith future devices, always write bits to 0. More information in Section 2.1 (p. Oversampling
	0 1 Reserved OVS Sets the num	To ensure co	RW	The bus clock used in synchronous mode has a high base value ith future devices, always write bits to 0. More information in Section 2.1 (p. Oversampling
	0 1 Reserved OVS Sets the num better perform	Ox0 ber of clock periods in a Unance.	RW	The bus clock used in synchronous mode has a high base value ith future devices, always write bits to 0. More information in Section 2.1 (p. Oversampling id. More clock cycles gives better robustness, while less clock cycles gives
	0 1 Reserved OVS Sets the numbetter perform Value	DLEHIGH To ensure consumer of the consumer of	RW	The bus clock used in synchronous mode has a high base value ith future devices, always write bits to 0. More information in Section 2.1 (p. Oversampling id. More clock cycles gives better robustness, while less clock cycles gives Description
	0 1 Reserved OVS Sets the num better perform Value 0	DLEHIGH To ensure co 0x0 ber of clock periods in a Unance. Mode X16	RW	The bus clock used in synchronous mode has a high base value ith future devices, always write bits to 0. More information in Section 2.1 (p. Oversampling id. More clock cycles gives better robustness, while less clock cycles gives Description Regular UART mode with 16X oversampling in asynchronous mode
	0 1 Reserved OVS Sets the numbetter perform Value 0 1	DLEHIGH To ensure co 0x0 ber of clock periods in a Unance. Mode X16 X8	RW	The bus clock used in synchronous mode has a high base value ith future devices, always write bits to 0. More information in Section 2.1 (p. Oversampling id. More clock cycles gives better robustness, while less clock cycles give Description Regular UART mode with 16X oversampling in asynchronous mode Double speed with 8X oversampling in asynchronous mode
:5	OVS Sets the numbetter perform Value 0 1 2	DLEHIGH To ensure co 0x0 ber of clock periods in a Unance. Mode X16 X8 X6	RW	The bus clock used in synchronous mode has a high base value ith future devices, always write bits to 0. More information in Section 2.1 (p. Oversampling id. More clock cycles gives better robustness, while less clock cycles gives Description Regular UART mode with 16X oversampling in asynchronous mode Double speed with 8X oversampling in asynchronous mode 6X oversampling in asynchronous mode



Bit	Name	Reset	Access	Description
	Multi-processor n	node uses the 9th bit of t	he USART frame	s to tell whether the frame is an address frame or a data frame.
	Value	Description		
	0	The 9th bit of inco	oming frames has n	no special function
	1		ne with the 9th bit e MPAB interrupt flag	equal to MPAB will be loaded into the receive buffer regardless of RXBLOCK and being set
2	CCEN	0	RW	Collision Check Enable
	Enables collision	checking on data when	operating in half of	duplex modus.
	Value	Description		
	0	Collision check is	disabled	
	1	Collision check is	enabled. The recei	iver must be enabled for the check to be performed
1	LOOPBK	0	RW	Loopback Enable
	Allows the receive	er to be connected direct	tly to the USART	transmitter for loopback and half duplex communication.
	Value	Description		
	0	The receiver is co	onnected to and rec	eives data from U(S)n_RX
	1	The receiver is co	onnected to and rec	eeives data from U(S)n_TX
0	SYNC	0	RW	USART Synchronous Mode
	Determines wheth	her the USART is operat	ing in asynchrono	ous or synchronous mode.
	Value	Description		
	0	The USART oper	rates in asynchrono	us mode

16.5.2 USARTn_FRAME - USART Frame Format Register

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	8	7	9	2	4	8	2	-	0
Reset																		0×1			>	2						, Y	2			
Access																				S ≷			>								<u> </u>	
Name																				STOPBITS			VTIQVQ							OFIGVE	2 10 7	
																				STC			۵	-						ځ	5	

				0,													
Bit	Name	Reset	Access	s Description													
31:14	Reserved	To ensure c	ompatibility with	h future devices, always write bits to 0. More information in Section 2.1 (p. 3)													
13:12	STOPBITS	0x1	RW	Stop-Bit Mode													
	Determines the	number of stop-bits used.															
	Value	Mode	1	Description													
	0	HALF	7	The transmitter generates a half stop bit. Stop-bits are not verified by receiver													
	1	ONE	(e 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	7	The transmitter generates two stop bits. The receiver checks the first stop-bit only													
11:10	Reserved	To ensure c	ompatibility with	h future devices, always write bits to 0. More information in Section 2.1 (p. 3)													
9:8	PARITY	0x0	RW	Parity-Bit Mode													
	Determines wh	ether parity bits are enable	ed, and whethe	er even or odd parity should be used. Only available in asynchronous mode.													
	Value	Mode]	Description													
	0	NONE	F	Parity bits are not used													
	2	EVEN	E	Even parity are used. Parity bits are automatically generated and checked by hardware.													

Odd parity is used. Parity bits are automatically generated and checked by hardware.

ODD



Bit	Name	Reset	Access	Description								
7:4	Reserved	To ensure c	ompatibility with fu	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)								
3:0	DATABITS	0x5	RW	Data-Bit Mode								
	This register se	ets the number of data bits	in a USART fram	e.								
	Value	Mode	Des	scription								
	1	FOUR	Eac	ch frame contains 4 data bits								
	2	FIVE	Eac	ch frame contains 5 data bits								
	3	SIX	Eac	ch frame contains 6 data bits								
	4	SEVEN	Eac	ch frame contains 7 data bits								
	5	EIGHT	Eac	ch frame contains 8 data bits								
	6	NINE	Eac	ch frame contains 9 data bits								
	7	TEN	Eac	ch frame contains 10 data bits								
	8	ELEVEN	Eac	ch frame contains 11 data bits								
	9	TWELVE	Eac	Each frame contains 12 data bits								
	10	THIRTEEN	Eac	ch frame contains 13 data bits								
	11	FOURTEEN	Eac	Each frame contains 14 data bits								
	12	FIFTEEN	Eac	Each frame contains 15 data bits								
	13	SIXTEEN	Eac	ch frame contains 16 data bits								

16.5.3 USARTn_TRIGCTRL - USART Trigger Control register

Offset	Bit Position			
0x008	33 34 35 36 37 38 38 38 38 38 38 38 38 38 48 40 <td>رى ح</td> <td>. ო</td> <td>0 + 0</td>	رى ح	. ო	0 + 0
Reset		0 0	,	0x0
Access		%		₩ %
Name		TXTEN		TSEL

Bit	Name	Reset	Access	Description
31:6	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
5	TXTEN	0	RW	Transmit Trigger Enable
	When set, the	PRS channel selected by T	SEL sets TXEN,	enabling the transmitter on positive trigger edges.
4	RXTEN	0	RW	Receive Trigger Enable
	When set, the	PRS channel selected by T	SEL sets RXEN,	enabling the receiver on positive trigger edges.
3	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
2:0	TSEL	0x0	RW	Trigger PRS Channel Select
	Select USAR	Γ PRS trigger channel. The F	PRS signal can e	nable RX and/or TX, depending on the setting of RXTEN and TXTEN.
	Value	Mode	Des	cription
	0	PRSCH0	PRS	S Channel 0 selected
	1	PRSCH1	PRS	S Channel 1 selected
	2	PRSCH2	PRS	S Channel 2 selected
	3	PRSCH3	PRS	S Channel 3 selected
	4	PRSCH4	PRS	S Channel 4 selected
	5	PRSCH5	PRS	S Channel 5 selected
	6	PRSCH6	PRS	S Channel 6 selected



16.5.4 USARTn_CMD - Command Register

Offset															Bi	t Po	siti	on														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	7	10	6	ω	7	9	2	4	က	2	-	0
Reset																					0	0	0	0	0	0	0	0	0	0	0	0
Access																					W	W1	W1	W1	W1	W1	W1	W	W1	N V	W1	W
Name																					CLEARRX	CLEARTX	TXTRIDIS	TXTRIEN	RXBLOCKDIS	RXBLOCKEN	MASTERDIS	MASTEREN	TXDIS	TXEN	RXDIS	RXEN

Bit	Name	Reset	Access	Description
31:12	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
11	CLEARRX	0	W1	Clear RX
	Set to clear receive b	uffer and the RX shi	ft register.	
10	CLEARTX	0	W1	Clear TX
	Set to clear transmit b	ouffer and the TX shi	ift register.	
9	TXTRIDIS	0	W1	Transmitter Tristate Disable
	Disables tristating of	the transmitter outpu	ıt.	
8	TXTRIEN	0	W1	Transmitter Tristate Enable
	Tristates the transmit	ter output.		
7	RXBLOCKDIS	0	W1	Receiver Block Disable
	Set to clear RXBLOC	K, resulting in all inc	oming frames bei	ng loaded into the receive buffer.
6	RXBLOCKEN	0	W1	Receiver Block Enable
	Set to set RXBLOCK	, resulting in all incor	ming frames being	discarded.
5	MASTERDIS	0	W1	Master Disable
	Set to disable master	mode, clearing the l	MASTER status b	it and putting the USART in slave mode.
4	MASTEREN	0	W1	Master Enable
		, ,		Master mode should not be enabled while TXENS is set to 1. To enable , or enable them both in the same write operation.
3	TXDIS	0	W1	Transmitter Disable
	Set to disable transm	ission.		
2	TXEN	0	W1	Transmitter Enable
	Set to enable data tra	insmission.		
1	RXDIS	0	W1	Receiver Disable
	Set to disable data re	ception. If a frame is	under reception	when the receiver is disabled, the incoming frame is discarded.
0	RXEN	0	W1	Receiver Enable
	Set to activate data re	eception on U(S)n_R	ex.	



16.5.5 USARTn_STATUS - USART Status Register

Offset															Bi	t Pc	siti	on														
0x010	31	30	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	41	13	12	7	10	6	∞	7	9	2	4	က	2	-	0
Reset																								0	0	-	0	0	0	0	0	0
Access																								~	~	2	~	~	~	~	~	22
Name																								RXFULL	RXDATAV	TXBL	DXL	TXTRI	RXBLOCK	MASTER	TXENS	RXENS

	_			
Bit	Name	Reset	Access	Description
31:9	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
8	RXFULL	0	R	RX FIFO Full
	Set when the RXFIFO is further frame in the receive shift re		ne receive buff	er is no longer full. When this bit is set, there is still room for one more
7	RXDATAV	0	R	RX Data Valid
	Set when data is available	in the receive buffe	r. Cleared whe	en the receive buffer is empty.
6	TXBL	1	R	TX Buffer Level
	Indicates the level of the tra TXBL is set whenever the t			TXBL is set whenever the transmit buffer is empty, and if TXBIL is set, v.
5	TXC	0	R	TX Complete
	Set when a transmission h is written to the transmit bu		no more data	is available in the transmit buffer and shift register. Cleared when data
4	TXTRI	0	R	Transmitter Tristated
	Set when the transmitter is is always read as 0.	tristated, and clea	red when tran	smitter output is enabled. If AUTOTRI in USARTn_CTRL is set this bit
3	RXBLOCK	0	R	Block Incoming Data
	When set, the receiver disc instant the frame has been			ng frame will not be loaded into the receive buffer if this bit is set at the
2	MASTER	0	R	SPI Master Mode
	Set when the USART opera	ates as a master. S	et using the M	ASTEREN command and clear using the MASTERDIS command.
1	TXENS	0	R	Transmitter Enable Status
	Set when the transmitter is	enabled.		
0	RXENS	0	R	Receiver Enable Status
	Set when the receiver is en	abled.		

16.5.6 USARTn_CLKDIV - Clock Control Register

Offset															Bi	t Po	siti	on														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	1	10	6	8	7	9	2	4	က	2	-	0
Reset																			0x0000													
Access																			RW													
Name																			DIV													



Bit	Name	Reset	Access	Description
31:21	Reserved	To ensure compa	atibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
20:6	DIV	0x0000	RW	Fractional Clock Divider
	Specifies the fractional clos	ck divider for the US	SART.	
5:0	Reserved	To ensure compa	atibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)

16.5.7 USARTn_RXDATAX - RX Buffer Data Extended Register

Offset															Bi	t Po	siti	on														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	∞	7	9	2	4	က	2	-	0
Reset				•													0	0					,					000x0				
Access																	œ	~										2				
Name																	FERR	PERR										RXDATA				

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15	FERR	0	R	Data Framing Error
	Set if data in buffer	has a framing error. Ca	an be the result o	f a break condition.
14	PERR	0	R	Data Parity Error
	Set if data in buffer	has a parity error (asyı	nchronous mode	only).
13:9	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
8:0	RXDATA	0x000	R	RX Data
	Use this register to	access data read from	the USART. Buff	er is cleared on read access.

16.5.8 USARTn_RXDATA - RX Buffer Data Register

Offset								,				,			Bi	t Po	siti	on						,								
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	=	10	6	∞	7	9	2	4	က	2	-	0
Reset																													0x0			
Access																													Y			
Name																												į	KXDATA			

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	ompatibility with f	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:0	RXDATA	0x00	R	RX Data
	Use this register to	access data read from	USART. Buffer i	s cleared on read access. Only the 8 LSB can be read using this register.



16.5.9 USARTn_RXDOUBLEX - RX Buffer Double Data Extended Register

Offset															Bi	t Po	siti	on														
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	7	10	6	∞	7	9	2	4	က	2	-	0
Reset	0	0						A A A A A A A A A A A A A A A A A A A																								
Access	~	22																														
Name	FERR1	PERR1										RXDATA1					FERR0	PERR0										RXDATA0				

Bit	Name	Reset	Access	Description
31	FERR1	0	R	Data Framing Error 1
	Set if data in buffer	has a framing error. Ca	an be the result o	f a break condition.
30	PERR1	0	R	Data Parity Error 1
	Set if data in buffer	has a parity error (asyr	nchronous mode	only).
29:25	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
24:16	RXDATA1	0x000	R	RX Data 1
	Second frame read	I from buffer.		
15	FERR0	0	R	Data Framing Error 0
	Set if data in buffer	has a framing error. Ca	an be the result o	f a break condition.
14	PERR0	0	R	Data Parity Error 0
	Set if data in buffer	has a parity error (asyr	nchronous mode	only).
13:9	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
8:0	RXDATA0	0x000	R	RX Data 0
	First frame read fro	om buffer.		

16.5.10 USARTn_RXDOUBLE - RX FIFO Double Data Register

Offset															Bi	t Pc	siti	on														
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	8	7	9	2	4	က	2	-	0
Reset																					0000								0000			
Access																					ď								ď			
Name																					RXDATA1								RXDATA0			

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:8	RXDATA1	0x00	R	RX Data 1
	Second frame read fro	om buffer.		
7:0	RXDATA0	0x00	R	RX Data 0
	First frame read from b	ouffer.		



16.5.11 USARTn_RXDATAXP - RX Buffer Data Extended Peek Register

Offset															Bi	t Po	siti	on														
0x028	31	30	29	28	27	56	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	80	7	9	2	4	က	2	-	0
Reset																	0	0										000x0				
Access																	2	~										2				
Name																	FERRP	PERRP										RXDATAP				

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15	FERRP	0	R	Data Framing Error Peek
	Set if data in buffer has a	a framing error. Ca	an be the result of	f a break condition.
14	PERRP	0	R	Data Parity Error Peek
	Set if data in buffer has a	a parity error (asyr	nchronous mode	only).
13:9	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
8:0	RXDATAP	0x000	R	RX Data Peek
	Use this register to acces	ss data read from	the USART.	

16.5.12 USARTn_RXDOUBLEXP - RX Buffer Double Data Extended Peek Register

Offset															Bi	t Pc	siti	on														
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	=	10	0	œ	7	9	2	4	ю	2	-	0
Reset	0	0										000×0					0	0										000×0				
Access	~	~										œ					œ	~										ď				
Name	FERRP1	PERRP1										RXDATAP1					FERRP0	PERRP0										RXDATAP0				

Bit	Name	Reset	Access	Description								
31	FERRP1	0	R	Data Framing Error 1 Peek								
	Set if data in buffer h	nas a framing error. C	an be the result o	f a break condition.								
30	PERRP1	0	R	Data Parity Error 1 Peek								
	Set if data in buffer h	nas a parity error (asy	error (asynchronous mode only).									
29:25	Reserved	To ensure co	To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (
24:16	RXDATAP1	0x000	R	RX Data 1 Peek								
	Second frame read f	from FIFO.										
15	FERRP0	0	R	Data Framing Error 0 Peek								
	Set if data in buffer h	nas a framing error. C	an be the result o	f a break condition.								
14	PERRP0	0	R	Data Parity Error 0 Peek								



Bit	Name	Reset	Access	Description
	Set if data in buffer h	as a parity error (asyı	nchronous mode	only).
13:9	Reserved	To ensure co	mpatibility with fu	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
8:0	RXDATAP0	0x000	R	RX Data 0 Peek
	First frame read from	FIFO.		

16.5.13 USARTn_TXDATAX - TX Buffer Data Extended Register

Offset															Bi	t Po	siti	on														
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	80	7	9	2	4	က	2	-	0
Reset																	0	0	0	0	0							000x0				
Access																	>	>	>	>	>							>				
Name																	RXENAT	TXDISAT	TXBREAK	TXTRIAT	UBRXAT							TXDATAX				

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15	RXENAT	0	W	Enable RX After Transmission
	Set to enable reception	after transmission		
14	TXDISAT	0	W	Clear TXEN After Transmission
	Set to disable transmitte	er and release data	a bus directly after	r transmission.
13	TXBREAK	0	W	Transmit Data As Break
	Set to send data as a bound of TXDATA.	oreak. Recipient w	ill see a framing o	error or a break condition depending on its configuration and the value
12	TXTRIAT	0	W	Set TXTRI After Transmission
	Set to tristate transmitte	er by setting TXTR	after transmissio	n.
11	UBRXAT	0	W	Unblock RX After Transmission
	Set clear RXBLOCK aft	er transmission, u	nblocking the rece	viver.
10:9	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
8:0	TXDATAX	0x000	W	TX Data
	Use this register to write	e data to the USAF	RT. If TXEN is set,	a transfer will be initiated at the first opportunity.

16.5.14 USARTn_TXDATA - TX Buffer Data Register

Offset															Bi	t Po	siti	on														
0x034	31	30	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	41	13	12	7	9	6	80	7	9	2	4	က	2	-	0
Reset																													0x00			
Access																													≥			
Name																													TXDATA			



Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	mpatibility with t	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:0	TXDATA	0x00	W	TX Data
	This frame will be a	dded to TX buffer. Only	/ 8 LSB can be	written using this register. 9th bit and control bits will be cleared.

16.5.15 USARTn_TXDOUBLEX - TX Buffer Double Data Extended Register

Offset															Ві	t Po	siti	on														
0x038	31	30	29	28	27	56	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	∞	7	9	2	4	က	2	-	0
Reset	0	0	0	0	0							000x0					0	0	0	0	0							0×000				
Access	≥	>	>	≥	>							≥					≥	≥	>	≥	≥							≥				
Name	RXENAT1	TXDISAT1	TXBREAK1	TXTRIAT1	UBRXAT1							TXDATA1					RXENAT0	TXDISAT0	TXBREAKO	TXTRIAT0	UBRXAT0							TXDATA0				

Bit	Name	Reset	Access	Description
31	RXENAT1	0	W	Enable RX After Transmission
	Set to enable recepti	ion after transmission		
30	TXDISAT1	0	W	Clear TXEN After Transmission
	Set to disable transm	nitter and release data	a bus directly afte	r transmission.
29	TXBREAK1	0	W	Transmit Data As Break
	Set to send data as of USARTn_TXDAT/		ill see a framing	error or a break condition depending on its configuration and the value
28	TXTRIAT1	0	W	Set TXTRI After Transmission
	Set to tristate transm	nitter by setting TXTR	after transmission	n.
27	UBRXAT1	0	W	Unblock RX After Transmission
	Set clear RXBLOCK	after transmission, u	nblocking the rece	eiver.
26:25	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
24:16	TXDATA1	0x000	W	TX Data
	Second frame to writ	te to FIFO.		
15	RXENAT0	0	W	Enable RX After Transmission
	Set to enable recepti	ion after transmission		
14	TXDISAT0	0	W	Clear TXEN After Transmission
	Set to disable transm	nitter and release data	a bus directly afte	r transmission.
13	TXBREAK0	0	W	Transmit Data As Break
	Set to send data as of TXDATA.	a break. Recipient w	ill see a framing	error or a break condition depending on its configuration and the value
12	TXTRIAT0	0	W	Set TXTRI After Transmission
	Set to tristate transm	nitter by setting TXTR	after transmission	n.
11	UBRXAT0	0	W	Unblock RX After Transmission
	Set clear RXBLOCK	after transmission, u	nblocking the rece	eiver.
10:9	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
8:0	TXDATA0	0x000	W	TX Data
	First frame to write to	buffer.		



16.5.16 USARTn_TXDOUBLE - TX Buffer Double Data Register

Offset															Bi	t Po	siti	on														
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	8	7	9	2	4	3	2	-	0
Reset																					0000								0000			
Access																					≥								≥			
Name																					TXDATA1							į	TXDATA0			

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:8	TXDATA1	0x00	W	TX Data
	Second frame to wi	rite to buffer.		
7:0	TXDATA0	0x00	W	TX Data
	First frame to write	to buffer.		

16.5.17 USARTn_IF - Interrupt Flag Register

Offset	Bit Position																															
0x040	31	30	29	28	27	56	22	24	23	22	21	20	19	9	17	16	15	4	13	12	7	10	6	∞	7	9	2	4	ю	2	-	0
Reset					•				•								•			0	0	0	0	0	0	0	0	0	0	0	_	0
Access																				œ	œ	22	22	œ	œ	22	œ	œ	~	~	~	~
Name																				CCF	SSM	MPAF	FERR	PERR	TXUF	TXOF	RXUF	RXOF	RXFULL	RXDATAV	TXBL	TXC

Bit	Name	Reset	Access	Description										
31:13	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)										
12	CCF	0	R	Collision Check Fail Interrupt Flag										
	Set when a collision check notices an error in the transmitted data.													
11	SSM	0	R	Slave-Select In Master Mode Interrupt Flag										
	Set when the device is selected as a slave when in master mode.													
10	MPAF	0	R	Multi-Processor Address Frame Interrupt Flag										
	Set when a multi-processor address frame is detected.													
9	FERR	0	R Framing Error Interrupt Flag											
	Set when a frame with a framing error is received while RXBLOCK is cleared.													
8	PERR	0	0 R Parity Error Interrupt Flag											
	Set when a frame with a parity error (asynchronous mode only) is received while RXBLOCK is cleared.													
7	TXUF	0	R	TX Underflow Interrupt Flag										
	Set when operating as a synchronous slave, no data is available in the transmit buffer when the master starts transmission of a new frame.													
6	TXOF	0	R	TX Overflow Interrupt Flag										
	Set when a write is	s done to the transmit b	uffer while it is ful	I. The data already in the transmit buffer is preserved.										
5	RXUF	0	R	RX Underflow Interrupt Flag										
	Set when trying to read from the receive buffer when it is empty.													
4	RXOF	0	R	RX Overflow Interrupt Flag										



Bit	Name	Reset	Access	Description
	Set when data is ir	ncoming while the recei	ve shift register is	full. The data previously in the shift register is lost.
3	RXFULL	0	R	RX Buffer Full Interrupt Flag
	Set when the recei	ive buffer becomes full.		
2	RXDATAV	0	R	RX Data Valid Interrupt Flag
	Set when data bed	comes available in the r	eceive buffer.	
1	TXBL	1	R	TX Buffer Level Interrupt Flag
	Set when the buffer if TXBIL is set.	er becomes empty if TX	(BIL is cleared, ar	nd is set whenever the transmit buffer goes from full to half-full or empty
0	TXC	0	R	TX Complete Interrupt Flag
	This interrupt is us	ed after a transmission	when both the TX	Cuffer and shift register are empty.

16.5.18 USARTn_IFS - Interrupt Flag Set Register

Offset				,				,							Bi	t Po	siti	on					,									
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	8	7	9	2	4	ю	2	1	0
Reset											•									0	0	0	0	0	0	0	0	0	0			0
Access																				W	W 1	W 1	W1	W1	W	W1	W	W	W			W1
Name																				CCF	SSM	MPAF	FERR	PERR	TXUF	TXOF	RXUF	RXOF	RXFULL			TXC

Bit	Name	Reset	Access	Description
31:13	Reserved	To ensure com	patibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
12	CCF	0	W1	Set Collision Check Fail Interrupt Flag
	Write to 1 to set the	CCF interrupt flag.		
11	SSM	0	W1	Set Slave-Select in Master mode Interrupt Flag
	Write to 1 to set the	SSM interrupt flag.		
10	MPAF	0	W1	Set Multi-Processor Address Frame Interrupt Flag
	Write to 1 to set the	MPAF interrupt flag.		
9	FERR	0	W1	Set Framing Error Interrupt Flag
	Write to 1 to set the	FERR interrupt flag.		
8	PERR	0	W1	Set Parity Error Interrupt Flag
	Write to 1 to set the	PERR interrupt flag.		
7	TXUF	0	W1	Set TX Underflow Interrupt Flag
	Write to 1 to set the	TXUF interrupt flag.		
6	TXOF	0	W1	Set TX Overflow Interrupt Flag
	Write to 1 to set the	TXOF interrupt flag.		
5	RXUF	0	W1	Set RX Underflow Interrupt Flag
	Write to 1 to set the	RXUF interrupt flag.		
4	RXOF	0	W1	Set RX Overflow Interrupt Flag
	Write to 1 to set the	RXOF interrupt flag.		
3	RXFULL	0	W1	Set RX Buffer Full Interrupt Flag
	Write to 1 to set the	RXFULL interrupt flag.		
2:1	Reserved	To ensure com	patibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	TXC	0	W1	Set TX Complete Interrupt Flag
	Write to 1 to set the	TXC interrupt flag.		



16.5.19 USARTn_IFC - Interrupt Flag Clear Register

Offset															Bi	t Po	siti	on														
0x048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	-	10	0	œ	7	9	2	4	က	2	-	0
Reset				•	•				•											0	0	0	0	0	0	0	0	0	0			0
Access																				W1	W1	W1	W1	W1	W1	W1	W1	W1	W1			W1
Name																				CCF	SSM	MPAF	FERR	PERR	TXUF	TXOF	RXUF	RXOF	RXFULL			TXC

Bit	Name	Reset	Access	Description
31:13	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
12	CCF	0	W1	Clear Collision Check Fail Interrupt Flag
	Write to 1 to clear the	he CCF interrupt flag.		
11	SSM	0	W1	Clear Slave-Select In Master Mode Interrupt Flag
	Write to 1 to clear the	he SSM interrupt flag.		
10	MPAF	0	W1	Clear Multi-Processor Address Frame Interrupt Flag
	Write to 1 to clear the	he MPAF interrupt flag.		
9	FERR	0	W1	Clear Framing Error Interrupt Flag
	Write to 1 to clear the	he FERR interrupt flag.		
8	PERR	0	W1	Clear Parity Error Interrupt Flag
	Write to 1 to clear the	he PERR interrupt flag.		
7	TXUF	0	W1	Clear TX Underflow Interrupt Flag
	Write to 1 to clear the	he TXUF interrupt flag.		
6	TXOF	0	W1	Clear TX Overflow Interrupt Flag
	Write to 1 to clear the	he TXOF interrupt flag.		
5	RXUF	0	W1	Clear RX Underflow Interrupt Flag
	Write to 1 to clear the	he RXUF interrupt flag.		
4	RXOF	0	W1	Clear RX Overflow Interrupt Flag
	Write to 1 to clear the	he RXOF interrupt flag.		
3	RXFULL	0	W1	Clear RX Buffer Full Interrupt Flag
	Write to 1 to clear the	he RXFULL interrupt fla	ag.	
2:1	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	TXC	0	W1	Clear TX Complete Interrupt Flag
	Write to 1 to clear the	he TXC interrupt flag.		

16.5.20 USARTn_IEN - Interrupt Enable Register

Offset															Bi	t Po	siti	on														
0x04C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	∞	7	9	2	4	ю	2	-	0
Reset												•	•		,					0	0	0	0	0	0	0	0	0	0	0	0	0
Access																				S.	RW W	R W	RW	RW	RW	RW	RW	W.	RW W	RW	W.	RW
Name																				CCF	SSM	MPAF	FERR	PERR	TXUF	TXOF	RXUF	RXOF	RXFULL	RXDATAV	TXBL	TXC



Bit	Name	Reset	Access	Description
31:13	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
12	CCF	0	RW	Collision Check Fail Interrupt Enable
	Enable interrupt on	collision check error d	etected.	
11	SSM	0	RW	Slave-Select In Master Mode Interrupt Enable
	Enable interrupt on	slave-select in master	mode.	
10	MPAF	0	RW	Multi-Processor Address Frame Interrupt Enable
	Enable interrupt on	multi-processor addre	ss frame.	
9	FERR	0	RW	Framing Error Interrupt Enable
	Enable interrupt on	framing error.		
8	PERR	0	RW	Parity Error Interrupt Enable
	Enable interrupt on	parity error (asynchro	nous mode only).	
7	TXUF	0	RW	TX Underflow Interrupt Enable
	Enable interrupt on	TX underflow.		
6	TXOF	0	RW	TX Overflow Interrupt Enable
	Enable interrupt on	TX overflow.		
5	RXUF	0	RW	RX Underflow Interrupt Enable
	Enable interrupt on	RX underflow.		
4	RXOF	0	RW	RX Overflow Interrupt Enable
	Enable interrupt on	RX overflow.		
3	RXFULL	0	RW	RX Buffer Full Interrupt Enable
	Enable interrupt on	RX Buffer full.		
2	RXDATAV	0	RW	RX Data Valid Interrupt Enable
	Enable interrupt on	RX data.		
1	TXBL	0	RW	TX Buffer Level Interrupt Enable
	Enable interrupt on	TX buffer level.		
0	TXC	0	RW	TX Complete Interrupt Enable
	Enable interrupt on	TX complete.		

16.5.21 USARTn_IRCTRL - IrDA Control Register

Offset															Bi	t Po	siti	on														
0x050	31	99	53	28	27	56	52	24	23	22	21	20	19	9	17	16	15	4	13	12	7	10	6	80	7	9	2	4	3	7	-	0
Reset															,										0		0x0		0	2	8	0
Access																									W.		ΑM		RW	<u> </u>		R ≷
Name																									IRPRSEN		IRPRSSEL		IRFILT	Waai		IREN

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	IRPRSEN	0	RW	IrDA PRS Channel Enable
	Enable the PRS channel s	selected by IRPRSS	EL as input to	IrDA module instead of TX.
6:4	IRPRSSEL	0x0	RW	IrDA PRS Channel Select
	A DDC can be used as inc	out to the pulse med	ulator instand	of TV. This value colocts the abannol to use



Bit	Name	Reset	Acce	ss Description
	Value	Mode		Description
	0	PRSCH0		PRS Channel 0 selected
	1	PRSCH1		PRS Channel 1 selected
	2	PRSCH2		PRS Channel 2 selected
	3	PRSCH3		PRS Channel 3 selected
	4	PRSCH4		PRS Channel 4 selected
	5	PRSCH5		PRS Channel 5 selected
	6	PRSCH6		PRS Channel 6 selected
	7	PRSCH7		PRS Channel 7 selected
3	IRFILT	0	RW	IrDA RX Filter
	Set to enable	filter on IrDA demodulator.		
	Value	Description		
	0	No filter enable	d	
	1	Filter enabled. I	rDA pulse must	be high for at least 4 consecutive clock cycles to be detected
2:1	IRPW	0x0	RW	IrDA TX Pulse Width
	Configure the	pulse width generated by t	the IrDA modu	ulator 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
0	IREN	0	RW	Enable IrDA Module
-				

16.5.22 USARTn_ROUTE - I/O Routing Register

Description

The USn_CLK pin is disabled

Offset															3it P	ositi	ion														
0x054	31	30	29	28	27	26	22	24	23	22	21	20	19	0 7	: 9	15	4	13	12	7	10	တ	∞	7	9	2	4	က	2	-	0
Reset																						2	Š					0	0	0	0
Access																						2	<u> </u>					RW	RW	RW	RW
Name																						NOITY						CLKPEN	CSPEN	TXPEN	RXPEN

					700		ਹ ਹ) -	<u>~</u>
Bit	Name	Rese	t Acces	ss Description					
31:10	Reserved	To en	sure compatibility w	vith future devices, always w	rite bits to 0. Mor	e information in S	Section	2.1 (µ	p. 3)
9:8	LOCATION	0x0	RW	I/O Location					
	Decides the loc	ation of the USART	I/O pins.						
	Value	Mode		Description					
	0	LOC0		Location 0					
	1	LOC1		Location 1					
	2	LOC2		Location 2					
	3	LOC3		Location 3					
7:4	Reserved	To en	sure compatibility w	vith future devices, always w	rite bits to 0. Mor	e information in S	Section	2.1 (p. 3)
3	CLKPEN	0	RW	CLK Pin Enable					
	When set, the	CLK pin of the USAF	RT is enabled.						

Value

0

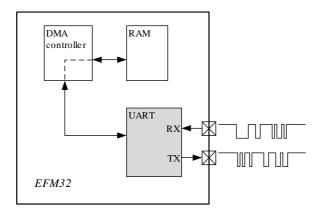


Bit	Name	Reset	Access	Description								
	Value	Description										
	1	The USn_CLK pi	n is enabled									
2	CSPEN	0	RW	CS Pin Enable								
	When set, the CS pin of the USART is enabled.											
	Value	Description										
	0 The USn_CS pin is disabled											
	1											
1	TXPEN	0	RW	TX Pin Enable								
	When set, the TX/MOSI pin of the USART is enabled											
	Value											
	0	The U(S)n_TX (MOSI) pin is disabled										
	1	The U(S)n_TX (M	IOSI) pin is enabled									
0	RXPEN	The U(S)n_TX (N	IOSI) pin is enabled	RX Pin Enable								
0			RW									
0		0	RW									
0	When set, the R	0 K/MISO pin of the USART	RW	RX Pin Enable								



17 UART - Universal Asynchronous Receiver/ Transmitter





Quick Facts

What?

The UART is capable of high-speed asynchronous serial communication.

Why?

Serial communication is frequently used in embedded systems and the UART allows efficient communication with a wide range of external devices.

How?

The UART has a wide selection of operating modes, frame formats and baud rates. The multi-processor mode allows the UART to remain idle when not addressed. Triple buffering and DMA support makes high datarates possible with minimal CPU intervention and it is possible to transmit and receive large frames while the MCU remains in EM1.

17.1 Introduction

The Universal Asynchronous serial Receiver and Transmitter (UART) is a very flexible serial I/O module. It supports full- and half-duplex asynchronous UART communication.

17.2 Features

- Full duplex and half duplex
- Separate TX / RX enable
- Separate receive / transmit 2-level buffers, with additional separate shift registers
- Programmable baud rate, generated as an fractional division from the peripheral clock (HFPERCLK)
- Max bit-rate
 - UART standard mode, peripheral clock rate / 16
 - UART FAST mode, peripheral clock rate / 8
- Asynchronous mode supports
 - Majority vote baud-reception
 - · False start-bit detection
 - Break generation/detection
 - Multi-processor mode
- 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
- Separate interrupt vectors for receive and transmit interrupts
- Loopback mode
 - Half duplex communication



- Communication debugging
- PRS can trigger transmissions
- Full DMA support

17.3 Functional Description

The UART is functionally equivalent to the USART with the exceptions defined in Table 17.1 (p. 223) . The register map and register descriptions are equal to those of the USART. See the USART chapter for detailed information on the operation of the UART.

Table 17.1. UART Limitations

Feature	Limitations
Synchronous operation	Not available. SYNC, CSMA, CSINV, CPOL and CPHA in USARTn_CTRL, and MASTEREN in USARTn_STATUS are always 0.
Transmission direction	Always LSB first. MSBF in USARTn_CTRL is always 0.
Chip-select	Not available. AUTOCS in USARTn_CTRL is always 0.
SmartCard mode	Not available. SCMODE in USARTn_CTRL is always 0.
Frame size	Limited to 8-9 databits. Other configurations of DATABITS in USARTn_FRAME are not possible.
IrDA	Not available. IREN in USARTn_IRCTRL is always 0.

17.4 Register Description

The register description of the UART is equivalent to the register description of the USART except the limitations mentioned in Table 17.1 (p. 223) . See the USART chapter for complete information.

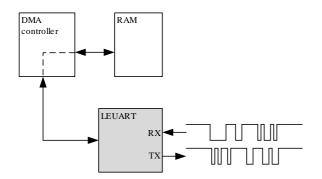
17.5 Register Map

The register map of the UART is equivalent to the register map of the USART. See the USART chapter for complete information.



18 LEUART - Low Energy Universal Asynchronous Receiver/Transmitter





Quick Facts

What?

The LEUART provides full UART communication using a low frequency 32.768 kHz clock, and has special features for communication without CPU intervention.

Why?

It allows UART communication to be performed in low energy modes, using only a few µA during active communication and only 150 nA when waiting for incoming data.

How?

A low frequency clock signal allows communication with less energy. Using DMA, the LEUART can transmit and receive data with minimal CPU intervention. Special UART-frames can be configured to help control the data flow, further automating data transmission.

18.1 Introduction

The unique LEUARTTM, the Low Energy UART, is a UART that allows two-way UART communication on a strict power budget. Only a 32.768 kHz clock is needed to allow UART communication at baud rates up to 9600.

Even when the EFM is in low energy mode EM2 (with most core functionality turned off), the LEUART can wait for an incoming UART frame while having an extremely low energy consumption. When a UART frame is completely received, 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.

Received data can optionally be blocked until a configurable start frame is detected. A signal frame can be configured to generate an interrupt to indicate e.g. the end of a data transmission. The start frame and signal frame can be used in combination for instance to handle higher level communication protocols.

Similarly, data can be transmitted in EM2 either on a frame-by-frame basis with data from the CPU or through use of the DMA.

The LEUART includes all necessary hardware support to make asynchronous serial communication possible with minimum of software intervention and energy consumption.

18.2 Features

- · Low energy asynchronous serial communications
- Full/half duplex communication
- Separate TX / RX enable
- Separate double buffered transmit buffer and receive buffer
- Programmable baud rate, generated as a fractional division of the LFBCLK
 - Supports baud rates from 300 baud/s to 9600 baud/s



- · Can use a high frequency clock source for even higher baud rates
- Configurable number of data bits: 8 or 9 (plus parity bit, if enabled)
- · Configurable parity: off, even or odd
 - · HW parity bit generation and check
- Configurable number of stop bits, 1 or 2
- · Capable of sleep-mode wake-up on received frame
 - Either wake-up on any received byte or
 - · Wake up only on specified start and signal frames
- · Supports transmission and reception in EM0, EM1 and EM2 with
 - Full DMA support
 - · Specified start-byte can start reception automatically
- IrDA modulator (pulse generator, pulse extender)
- Multi-processor mode
- Loopback mode
 - · Half duplex communication
 - · Communication debugging

18.3 Functional Description

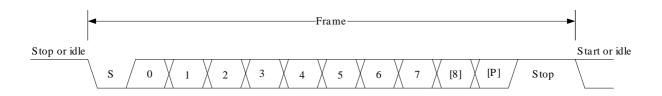
An overview of the LEUART module is shown in Figure 18.1 (p. 225).

Figure 18.1. LEUART Overview

18.3.1 Frame Format

The frame format used by the LEUART 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 8 or 9 data bits and an optional parity bit. The data is transmitted with the least significant bit first. Finally, a number of stop-bits, where the line is driven high, end the frame. The frame format is shown in Figure 18.2 (p. 225).

Figure 18.2. LEUART Asynchronous Frame Format



The number of data bits in a frame is set by DATABITS in LEUARTn_CTRL, and the number of stop-bits is set by STOPBITS in LEUARTn_CTRL. Whether or not a parity bit should be included, and whether it should be even or odd is defined by PARITY in LEUARTn_CTRL. For communication to be possible, all parties of an asynchronous transfer must agree on the frame format being used.

The frame format used by the LEUART can be inverted by setting INV in LEUARTn_CTRL. This affects the entire frame, resulting in a low idle state, a high start-bit, inverted data and parity bits, and low stop-bits. INV should only be changed while the receiver is disabled.

18.3.1.1 Parity Bit Calculation and Handling

Hardware automatically inserts parity bits into outgoing frames and checks the parity bits of incoming frames. The possible parity modes are defined in Table 18.1 (p. 226). 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 bits are disabled, which is the default configuration, the parity bit is omitted.

Table 18.1. LEUART Parity Bit

PARITY [1:0]	Description
00	No parity (default)
01	Reserved
10	Even parity
11	Odd parity

See Section 18.3.5.4 (p. 230) for more information on parity bit handling.

18.3.2 Clock Source

The LEUART clock source is selected by the LFB bit field the CMU_LFCLKSEL register. The clock is prescaled by the LEUARTn bitfield in the CMU_LFBPRESC0 register and enabled by the LEUARTn bit in the CMU_LFBCLKEN0.

To use this module, the LE interface clock must be enabled in CMU_HFCORECLKEN0, in addition to the module clock.

18.3.3 Clock Generation

The LEUART clock defines the transmission and reception data rate. The clock generator employs a fractional clock divider to allow baud rates that are not attainable by integral division of the 32.768 kHz clock that drives the LEUART.

The clock divider used in the LEUART is a 12-bit value, with a 7-bit integral part and a 5-bit fractional part. The baud rate of the LEUART is given by :

$$br = fLEUARTn/(1 + LEUARTn_CLKDIV/256)$$
 (18.1)

where fLEUARTn is the clock frequency supplied to the LEUART. The value of LEUARTn_CLKDIV thus defines the baud rate of the LEUART. The integral part of the divider is right-aligned in the upper 24 bits of LEUARTn_CLKDIV and the fractional part is left-aligned in the lower 8 bits. The divider is thus a 256th of LEUARTn_CLKDIV as seen in the equation.

For a desired baud rate br_{DESIRED}, LEUARTn_CLKDIV can be calculated by using:

LEUART CLKDIV Equation

$$LEUARTn_CLKDIV = 256 \times (fLEUARTn/br_{DESIRED} - 1)$$
 (18.2)

Table 18.2 (p. 227) lists a set of desired baud rates and the closest baud rates reachable by the LEUART with a 32.768 kHz clock source. It also shows the average baud rate error.



Table 18.2. LEUART Baud Rates

Desired baud rate [baud/s]	LEUARTn_CLKDIV	LEUARTn_CLKDIV/256	Actual baud rate [baud/s]	Error [%]
300	27704	108,21875	300,0217	0,01
600	13728	53,625	599,8719	-0,02
1200	6736	26,3125	1199,744	-0,02
2400	3240	12,65625	2399,487	-0,02
4800	1488	5,8125	4809,982	0,21
9600	616	2,40625	9619,963	0,21

18.3.4 Data Transmission

Data transmission is initiated by writing data to the transmit buffer using one of the methods described in Section 18.3.4.1 (p. 227). 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 LEUARTn_CMD by setting TXEN, and disabled by setting TXDIS. 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 LEUARTn_STATUS. After a transmission, when there is no more data in the shift register or transmit buffer, the TXC flag in LEUARTn_STATUS and the TXC interrupt flag in LEUARTn_IF are set, signaling that the transmitter is idle. The TXC status flag is cleared when a new byte becomes available for transmission, but the TXC interrupt flag must be cleared by software.

18.3.4.1 Transmit Buffer Operation

A frame can be loaded into the transmit buffer by writing to LEUARTn_TXDATA or LEUARTn_TXDATAX. Using LEUARTn_TXDATA allows 8 bits to be written to the buffer. If 9 bit frames are used, the 9th bit will in that case be set to the value of BIT8DV in LEUARTn_CTRL. To set the 9th bit directly and/or use transmission control, LEUARTn_TXDATAX must be used. When writing data to the transmit buffer using LEUARTn_TXDATAX, the 9th bit written to LEUARTn_TXDATAX overrides the value in BIT8DV, and alone defines the 9th bit that is transmitted if 9-bit frames are used.

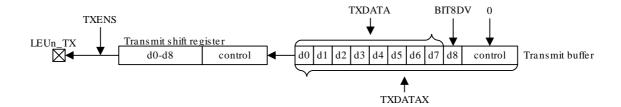
If a write is attempted to the transmit buffer when it is not empty, the TXOF interrupt flag in LEUARTn_IF is set, indicating the overflow. The data already in the buffer is in that case preserved, and no data is written.

In addition to the interrupt flag TXC in LEUARTn_IF and the status flag TXC in LEUARTn_STATUS which are set when the transmitter becomes idle, TXBL in LEUARTn_STATUS and the TXBL interrupt flag in LEUARTn_IF are used to indicate the level of the transmit buffer. Whenever the transmit buffer becomes empty, these flags are set high. Both the TXBL status flag and the TXBL interrupt flag are cleared automatically when data is written to the transmit buffer.

The transmit buffer, including the TX shift register can be cleared by setting command bit CLEARTX in LEUARTn_CMD. This will prevent the LEUART 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. An overview of the operation of the transmitter is shown in Figure 18.3 (p. 228) .



Figure 18.3. LEUART Transmitter Overview



18.3.4.2 Frame Transmission Control

The transmission control bits, which can be written using LEUARTn_TXDATAX, affect the transmission of the written frame. The following options are available:

- Generate break: By setting WBREAK, the output will be held low during the first 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 for one baud period before the next frame is transmitted so the next start condition can be identified correctly by the recipient. Continuous breaks lasting longer than an UART frame are thus not supported by the LEUART. GPIO can be used for this. Note that when AUTOTRI in LEUARTn_CTRL is used, the transmitter is not tristated before the high-bit after the break has been transmitted.
- 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.

The transmission control bits in the LEUART cannot tristate the transmitter. This is performed automatically by hardware however, if AUTOTRI in LEUARTn_CTRL is set. See Section 18.3.7 (p. 232) for more information on half duplex operation.

18.3.4.3 Jitter in Transmitted Data

Internally the LEUART module uses only the positive edges of the 32.768 kHz clock (LFBCLK) for transmission and reception. Transmitted data will thus have jitter equal to the difference between the optimal data set-up location and the closest positive edge on the 32.768 kHz clock. The jitter in on the location data is set up by the transmitter will thus be no more than half a clock period according to the optimal set-up location. The jitter in the period of a single baud output by the transmitter will never be more than one clock period.

18.3.5 Data Reception

Data reception is enabled by setting RXEN in LEUARTn_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 receive shift register is overwritten, and the RXOF interrupt flag in LEUARTn_IF is set to indicate the buffer overflow.

The receiver can be disabled by setting the command bit RXDIS in LEUARTn_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 LEUARTn_STATUS.

18.3.5.1 Receive Buffer Operation

When data becomes available in the receive buffer, the RXDATAV flag in LEUARTn_STATUS and the RXDATAV interrupt flag in LEUARTn_IF are set. Both the RXDATAV status flag and the RXDATAV interrupt flag are cleared by hardware when data is no longer available, i.e. when data has been read out of the buffer.

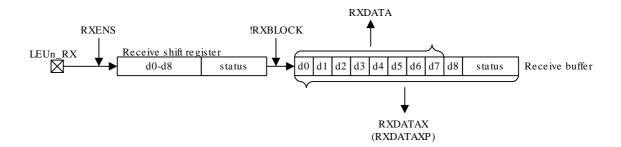
Data can be read from receive buffer using either LEUARTn_RXDATA or LEUARTn_RXDATAX. LEUARTn_RXDATA gives access to the 8 least significant bits of the received frame, while LEUARTn_RXDATAX must be used to get access to the 9th, most significant bit. The latter register also contains status information regarding the frame.

When a frame is read from the receive buffer using LEUARTn_RXDATA or LEUARTn_RXDATAX, the frame is removed from the buffer, making room for a new one. If an attempt is done to read more frames from the buffer than what is available, the RXUF interrupt flag in LEUARTn_IF is set to signal the underflow, and the data read from the buffer is undefined.

Frames can also be read from the receive buffer without removing the data by using LEUARTn_RXDATAXP, which gives access to the frame in the buffer including control bits. Data read from this register when the receive buffer is empty is undefined. No underflow interrupt is generated by a read using LEUARTn_RXDATAXP, i.e. the RXUF interrupt flag is never set as a result of reading from LEUARTn_RXDATAXP.

An overview of the operation of the receiver is shown in Figure 18.4 (p. 229).

Figure 18.4. LEUART Receiver Overview



18.3.5.2 Blocking Incoming Data

When using hardware frame recognition, as detailed in Section 18.3.5.6 (p. 231), Section 18.3.5.7 (p. 231), and Section 18.3.5.8 (p. 232), 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 LEUARTn_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 bit in LEUARTn_STATUS or the RXDATAV interrupt flag in LEUARTn_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 LEUARTn_CMD and disabled by setting RXBLOCKDIS also in LEUARTn_CMD. There are two exceptions where data is loaded into the receive buffer even when RXBLOCK is set. The first is when an address frame is received when in operating in multi-processor mode as shown in Section 18.3.5.8 (p. 232). The other case is when receiving a start-frame when SFUBRX in LEUARTn_CTRL is set; see Section 18.3.5.6 (p. 231)

Frames received containing framing or parity errors will not result in the FERR and PERR interrupt flags in LEUARTn_IF being set while RXBLOCK 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 LEUARTn_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 LEUARTn_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 is set.

18.3.5.3 Data Sampling

The receiver samples each incoming baud as close as possible to the middle of the baud-period. Except for the start-bit, only a single sample is taken of each of the incoming bauds.

The length of a baud-period is given by 1 + LEUARTn_CLKDIV/256, as a number of 32.768 kHz clock periods. Let the clock cycle where a start-bit is first detected be given the index 0. The optimal sampling point for each baud in the UART frame is then given by the following equation:

LEUART Optimal Sampling Point

$$S_{opt}(n) = n (1 + LEUARTn_CLKDIV/256) + CLKDIV/512$$
(18.3)

where n is the bit-index.

Since samples are only done on the positive edges of the 32.768 kHz clock, the actual samples are performed on the closest positive edge, i.e. the edge given by the following equation:

LEUART Actual Sampling Point

$$S(n) = floor(n \times (1 + LEUARTn_CLKDIV/256) + LEUARTn_CLKDIV/512)$$
(18.4)

The sampling location will thus have jitter according to difference between S_{opt} and S. The start-bit is found at n=0, then follows the data bits, any parity bit, and the stop bits.

If the value of the start-bit is found to be high, then the start-bit is discarded, and the receiver waits for a new start-bit.

18.3.5.4 Parity Error

When the parity bit is enabled, a parity check is automatically performed on incoming frames. When a parity error is detected in a frame, the data parity error bit PERR in the frame is set, as well as the interrupt flag PERR. 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 LEUARTn_RXDATAX register.

18.3.5.5 Framing Error and Break Detection

A framing error is the result of a received 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, the framing error bit FERR in the received frame is set. The interrupt flag FERR in LEUARTn_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 LEUARTn_RXDATAX or LEUARTn_RXDATAXP registers.

18.3.5.6 Programmable Start Frame

The LEUART 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 LEUART by transmitting a given frame.

When SFUBRX in LEUARTn_CTRL is set, an incoming frame matching the frame defined in LEUARTn_STARTFRAME will result in RXBLOCK in LEUARTn_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 LEUARTn_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 buffer. 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 buffer.

An incoming start frame results in the STARTF interrupt flag in LEUARTn_IF being set, regardless of the value of SFUBRX in LEUARTn_CTRL. This allows an interrupt to be made when the start frame is detected.

When 8 data-bit frame formats are used, only the 8 least significant bits of LEUARTn_STARTFRAME are compared to incoming frames. The full length of LEUARTn_STARTFRAME is used when operating with frames consisting of 9 data bits.

Note

The receiver must be enabled for start frames to be detected. In addition, a start frame with a parity error or framing error is not detected as a start frame.

18.3.5.7 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 LEUARTn_SIGFRAME is detected by the receiver, the SIGF interrupt flag in LEUARTn_IF is set. As for start frame detection, the receiver must be enabled for signal frames to be detected.

One use of the programmable signal frame is to signal the end of a multi-frame message transmitted to the LEUART. 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 LEUART 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. The device can thus wait for data packets in EM2, and only be woken up when a packet has been completely received.

A signal frame with a parity error or framing error is not detected as a signal frame.



18.3.5.8 Multi-Processor Mode

To simplify communication between multiple processors and maintain compatibility with the USART, the LEUART supports a 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 LEUARTn_CTRL is identified as an address frame. When an address frame is detected, the MPAF interrupt flag in LEUARTn_IF is set, and the address frame is loaded into the receive register. This happens regardless of the value of RXBLOCK in LEUARTn_STATUS.

Multi-processor mode is enabled by setting MPM in LEUARTn_CTRL. The mode can be used in buses with multiple slaves, allowing the slaves to be addressed using the special address frames. An addressed slave, which was previously blocking reception using RXBLOCK, would then unblock reception, receive a message from the bus master, and then block reception again, waiting for the next message. See the USART for a more detailed example.

Note

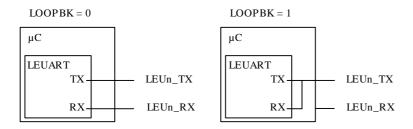
The programmable start frame functionality can be used for automatic address matching, enabling reception on a correctly configured incoming frame.

An address frame with a parity error or a framing error is not detected as an address frame.

18.3.6 Loopback

The LEUART receiver samples LEUn_RX by default, and the transmitter drives LEUn_TX by default. This is not the only configuration however. When LOOPBK in LEUARTn_CTRL is set, the receiver is connected to the LEUn_TX pin as shown in Figure 18.5 (p. 232). This is useful for debugging, as the LEUART can receive the data it transmits, but it is also used to allow the LEUART to read and write to the same pin, which is required for some half duplex communication modes. In this mode, the LEUn_TX pin must be enabled as an output in the GPIO.

Figure 18.5. LEUART Local Loopback



18.3.7 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.

18.3.7.1 Single Data-link

In this setup, the LEUART both receives and transmits data on the same pin. This is enabled by setting LOOPBK in LEUARTn_CTRL, which connects the receiver to the transmitter output. Because they are both connected to the same line, it is important that the LEUART 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. If AUTOTRI in LEUARTn_CTRL is set, the LEUART automatically tristates LEUn_TX whenever the transmitter is inactive. It is then the responsibility of the software protocol to make sure the transmitter is not transmitting data whenever incoming data is expected.

The transmitter can also be tristated from software by configuring the GPIO pin as an input and disabling the LEUART output on LEUn_TX.

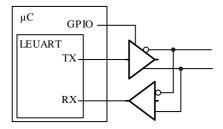
Note

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.

18.3.7.2 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. The USART has hardware support for automatically turning the driver on and off. When using the LEUART in such a setup, the driver must be controlled by a GPIO. Figure 18.6 (p. 233) shows an example configuration using an external driver.

Figure 18.6. LEUART Half Duplex Communication with External Driver



18.3.7.3 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.

18.3.8 Transmission Delay

By configuring TXDELAY in LEUARTn_CTRL, the transmitter can be forced to wait a number of bitperiods from it is ready to transmit data, to 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 baud 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.

18.3.9 DMA Support

The LEUART has full DMA support in energy modes EM0 – EM2. The DMA controller can write to the transmit buffer using the registers LEUARTn_TXDATA and LEUARTn_TXDATAX, and it can read from receive buffer using the registers LEUARTn_RXDATA and LEUARTn_RXDATAX. This enables single



byte transfers and 9 bit data + control/status bits transfers both to and from the LEUART. The DMA will start up the HFRCO and run from this when it is waken by the LEUART in EM2. The HFRCO is disabled once the transaction is done.

A request for the DMA controller to read from the receive buffer can come from one of the following sources:

· Receive buffer full

A write request can come from one of the following sources:

- Transmit buffer and shift register empty. No data to send.
- Transmit buffer empty

In some cases, it may be sensible to temporarily stop DMA access to the LEUART when a parity or framing error has occurred. This is enabled by setting ERRSDMA in LEUARTn_CTRL. When this bit is set, the DMA controller will not get requests from the receive buffer if a framing error or parity error is detected in the received byte. The ERRSDMA bit applies only to the RX DMA.

When operating in EM2, the DMA controller must be powered up in order to perform the transfer. This is automatically performed for read operations if RXDMAWU in LEUARTn_CTRL is set and for write operations if TXDMAWU in LEUARTn_CTRL is set. To make sure the DMA controller still transfers bits to and from the LEUART in low energy modes, these bits must thus be configured accordingly.

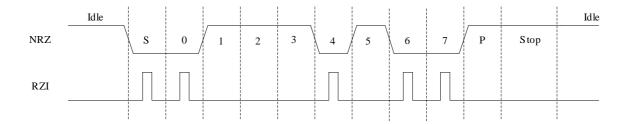
Note

When RXDMAWU or TXDMAWU is set, the system will not be able to go to EM2/EM3 before all related LEUART DMA requests have been processed. This means that if RXDMAWU is set and the LEUART receives a frame, the system will not be able to go to EM2/EM3 before the frame has been read from the LEUART. In order for the system to go to EM2 during the last byte transmission, LEUART_CTRL_TXDMAWU must be cleared in the DMA interrupt service routine. This is because TXBL will be high during that last byte transfer.

18.3.10 Pulse Generator/ Pulse Extender

The LEUART has an optional pulse generator for the transmitter output, and a pulse extender on the receiver input. These are enabled by setting PULSEEN in LEUARTn_PULSECTRL, and with INV in LEUARTn_CTRL set, they will change the output/input format of the LEUART from NRZ to RZI as shown in Figure 18.7 (p. 234) .

Figure 18.7. LEUART - NRZ vs. RZI



If PULSEEN in LEUARTn_PULSECTRL is set while INV in LEUARTn_CTRL is cleared, the output waveform will like RZI shown in Figure 18.7 (p. 234), only inverted.

The width of the pulses from the pulse generator can be configured using PULSEW in LEUARTn_PULSECTRL. The generated pulse width is PULSEW + 1 cycles of the 32.768 kHz clock, which makes pulse width from 31.25µs to 500µs possible.



Since the incoming signal is only sampled on positive clock edges, the width of the incoming pulses must be at least two 32.768 kHz clock periods wide for reliable detection by the LEUART receiver. They must also be shorter than half a UART baud period.

At 2400 baud/s or lower, the pulse generator is able to generate RZI pulses compatible with the IrDA physical layer specification. The external IrDA device must generate pulses of sufficient length for successful two-way communication.

18.3.10.1 Interrupts

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

18.3.11 Register access

Since this module is a Low Energy Peripheral, and runs off a clock which is asynchronous to the HFCORECLK, special considerations must be taken when accessing registers. Please refer to Section 5.3 (p. 19) for a description on how to perform register accesses to Low Energy Peripherals.



18.4 Register Map

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

Offset	Name	Туре	Description
0x000	LEUARTn_CTRL	RW	Control Register
0x004	LEUARTn_CMD	W1	Command Register
0x008	LEUARTn_STATUS	R	Status Register
0x00C	LEUARTn_CLKDIV	RW	Clock Control Register
0x010	LEUARTn_STARTFRAME	RW	Start Frame Register
0x014	LEUARTn_SIGFRAME	RW	Signal Frame Register
0x018	LEUARTn_RXDATAX	R	Receive Buffer Data Extended Register
0x01C	LEUARTn_RXDATA	R	Receive Buffer Data Register
0x020	LEUARTn_RXDATAXP	R	Receive Buffer Data Extended Peek Register
0x024	LEUARTn_TXDATAX	W	Transmit Buffer Data Extended Register
0x028	LEUARTn_TXDATA	W	Transmit Buffer Data Register
0x02C	LEUARTn_IF	R	Interrupt Flag Register
0x030	LEUARTn_IFS	W1	Interrupt Flag Set Register
0x034	LEUARTn_IFC	W1	Interrupt Flag Clear Register
0x038	LEUARTn_IEN	RW	Interrupt Enable Register
0x03C	LEUARTn_PULSECTRL	RW	Pulse Control Register
0x040	LEUARTn_FREEZE	RW	Freeze Register
0x044	LEUARTn_SYNCBUSY	R	Synchronization Busy Register
0x054	LEUARTn_ROUTE	RW	I/O Routing Register

18.5 Register Description

18.5.1 LEUARTn_CTRL - Control Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

Offset			Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	∞	7	9	2	4	က	2	-	0
Reset																	2	S S	0	0	0	0	0	0	0	0	0	0	2	OX O	0	0
Access																	3	<u>}</u>	RW	W.	R W	W.	RW	W.	W.	RW W	RW	N N	Š	<u>}</u>	W.W.	RW
Name																	2	ζ.	TXDMAWU	RXDMAWU	BIT8DV	MPAB	MPM	SFUBRX	LOOPBK	ERRSDMA	NI NI	STOPBITS	>Fig vg	- AAK	DATABITS	AUTOTRI

15:14	TYDEL AV	۸۷۸	D\M	TV Dolay Transmission
31:16	Reserved	To ensure compa	tibility with futu	re devices, always write bits to 0. More information in Section 2.1 (p. 3)
Bit	Name	Reset	Access	Description

Configurable delay before new transfers. Frames sent back-to-back are not delayed.

Value	Mode	Description
0	NONE	Frames are transmitted immediately
1	SINGLE	Transmission of new frames are delayed by a single baud period
2	DOUBLE	Transmission of new frames are delayed by two baud periods
3	TRIPLE	Transmission of new frames are delayed by three baud periods



Bit	Name	Reset	Access	Description								
13	TXDMAWU	0	RW	TX DMA Wakeup								
	Set to wake the	DMA controller up when in	n EM2 and space	e is available in the transmit buffer.								
	Value	Description										
	0	·	DMA controller wil	Il not get requests about space being available in the transmit buffer								
	1			lest about space available in the transmit buffer								
40	DYDMAMUL		-	·								
12	RXDMAWU	0	RW	RX DMA Wakeup								
	Set to wake the	JMA controller up when in	1 EM2 and data is	s available in the receive buffer.								
	Value	Description	·									
	0	While in EM2, the	DMA controller wil	Il not get requests about data being available in the receive buffer								
	1	DMA is available i	DMA is available in EM2 for the request about data in the receive buffer									
11	BIT8DV	0	RW	Bit 8 Default Value								
		is assigned to the 9th bit		9th bit is given by BIT8DV. If TXDATA is used to write a frame, then the frame. If a frame is written with TXDATAX however, the default value is								
10	MPAB	0	RW	Multi-Processor Address-Bit								
		e of the multi-processor ac	ddress bit. An inc	coming frame with its 9th bit equal to the value of this bit marks the frame								
9	MPM	0	RW	Multi-Processor Mode								
	Set to enable mu	ılti-processor mode.										
		· · · · · · · · · · · · · · · · · · ·										
	Value 0	Description The Oth hit of ince	amina framos bayo	no appoint function								
	1		The 9th bit of incoming frames have no special function An incoming frame with the 9th bit equal to MPAB will be loaded into the receive buffer regardless of RXBLOCK a									
			e with the attract e									
			IPAB interrupt flag I	being set								
8	SFUBRX		IPAB interrupt flag I	Start-Frame UnBlock RX								
8		will result in the M	RW	Start-Frame UnBlock RX								
8	Clears RXBLOC	will result in the M 0 K when the start-frame is	RW	-								
8	Clears RXBLOC	will result in the M 0 K when the start-frame is Description	RW found in the inco	Start-Frame UnBlock RX ming data. The start-frame is loaded into the receive buffer.								
8	Clears RXBLOC	will result in the M 0 K when the start-frame is Description Detected start-fram	RW found in the inco	Start-Frame UnBlock RX ming data. The start-frame is loaded into the receive buffer. on RXBLOCK								
	Clears RXBLOC Value 0 1	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram	RW found in the inco	Start-Frame UnBlock RX ming data. The start-frame is loaded into the receive buffer. I on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer								
7	Clears RXBLOC	will result in the M 0 K when the start-frame is Description Detected start-fram	RW found in the inco	Start-Frame UnBlock RX ming data. The start-frame is loaded into the receive buffer. on RXBLOCK								
	Clears RXBLOC Value 0 1 LOOPBK	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram	RW found in the inco	Start-Frame UnBlock RX ming data. The start-frame is loaded into the receive buffer. I on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer								
	Clears RXBLOC Value 0 1 LOOPBK	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0	RW found in the inco	Start-Frame UnBlock RX ming data. The start-frame is loaded into the receive buffer. I on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer								
	Clears RXBLOC Value 0 1 LOOPBK Set to connect re	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0 eceiver to LEUn_TX instea	RW found in the inco mes have no effect ne is detected, RXB RW ad of LEUn_RX.	Start-Frame UnBlock RX ming data. The start-frame is loaded into the receive buffer. I on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer								
	Clears RXBLOC Value 0 1 LOOPBK Set to connect re Value	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0 eceiver to LEUn_TX instead Description The receiver is co	RW found in the inco mes have no effect ne is detected, RXB RW ad of LEUn_RX.	Start-Frame UnBlock RX Iming data. The start-frame is loaded into the receive buffer. If on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer Loopback Enable								
	Clears RXBLOC Value 0 1 LOOPBK Set to connect re Value 0	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0 eceiver to LEUn_TX instead Description The receiver is co	RW found in the inco mes have no effect ne is detected, RXB RW ad of LEUn_RX.	Start-Frame UnBlock RX Iming data. The start-frame is loaded into the receive buffer. It on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer Loopback Enable Reives data from LEUn_RX								
7	Clears RXBLOC Value 0 1 LOOPBK Set to connect re Value 0 1 ERRSDMA	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0 eceiver to LEUn_TX instead Description The receiver is co The receiver is co	RW found in the inco mes have no effect ne is detected, RXB RW ad of LEUn_RX. onnected to and reconnected t	Start-Frame UnBlock RX uning data. The start-frame is loaded into the receive buffer. It on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer Loopback Enable Delives data from LEUn_RX Delives data from LEUn_TX Clear RX DMA On Error								
7	Clears RXBLOC Value 0 1 LOOPBK Set to connect re Value 0 1 ERRSDMA When set,RX DM	will result in the M 0 K when the start-frame is Description Detected start-fram 0 eceiver to LEUn_TX instead Description The receiver is co The receiver is co 0 MA requests will be cleared	RW found in the inco mes have no effect ne is detected, RXB RW ad of LEUn_RX. onnected to and reconnected t	Start-Frame UnBlock RX uning data. The start-frame is loaded into the receive buffer. It on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer Loopback Enable Delives data from LEUn_RX Delives data from LEUn_TX Clear RX DMA On Error								
7	Clears RXBLOC Value 0 1 LOOPBK Set to connect re Value 0 1 ERRSDMA When set,RX DM	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0 eceiver to LEUn_TX instead Description The receiver is co The receiver is co 0 MA requests will be cleared Description	RW found in the inco mes have no effect ne is detected, RXB RW ad of LEUn_RX. connected to and reco nnected to and reco nnected to and reco nnected to and reco nnected to and reco RW d on framing and	Start-Frame UnBlock RX Iming data. The start-frame is loaded into the receive buffer. If on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer Loopback Enable Relives data from LEUn_RX Relives data from LEUn_TX Clear RX DMA On Error I parity errors.								
7	Clears RXBLOC Value 0 1 LOOPBK Set to connect re Value 0 1 ERRSDMA When set,RX DM Value 0	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0 Description The receiver is co The receiver is co O MA requests will be cleared Description Framing and parit	RW found in the inco mes have no effect ne is detected, RXB RW ad of LEUn_RX. connected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco	Start-Frame UnBlock RX Iming data. The start-frame is loaded into the receive buffer. If on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer Loopback Enable Index derives data from LEUn_RX Index derives data from LEUn_TX Clear RX DMA On Error I parity errors. Iffect on DMA requests from the LEUART								
6	Clears RXBLOC Value 0 1 LOOPBK Set to connect re Value 0 1 ERRSDMA When set,RX DM Value 0 1	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0 eceiver to LEUn_TX instead Description The receiver is co The receiver is co 0 MA requests will be cleared Description Framing and parit RX DMA requests	RW found in the inco mes have no effect ne is detected, RXB RW ad of LEUn_RX. connected to and reconnected t	Start-Frame UnBlock RX Iming data. The start-frame is loaded into the receive buffer. It on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer Loopback Enable Relives data from LEUn_RX Relives data from LEUn_TX Clear RX DMA On Error I parity errors. Iffect on DMA requests from the LEUART Tare disabled if a framing error or parity error occurs.								
7	Clears RXBLOC Value 0 1 LOOPBK Set to connect re Value 0 1 ERRSDMA When set,RX DM Value 0	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0 Description The receiver is co The receiver is co O MA requests will be cleared Description Framing and parit	RW found in the inco mes have no effect ne is detected, RXB RW ad of LEUn_RX. connected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco nected to and reco	Start-Frame UnBlock RX Iming data. The start-frame is loaded into the receive buffer. If on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer Loopback Enable Index derives data from LEUn_RX Index derives data from LEUn_TX Clear RX DMA On Error I parity errors. Iffect on DMA requests from the LEUART								
6	Clears RXBLOC Value 0 1 LOOPBK Set to connect re Value 0 1 ERRSDMA When set,RX DM Value 0 1 INV	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0 eceiver to LEUn_TX instead Description The receiver is co The receiver is co 0 MA requests will be cleared Description Framing and parit RX DMA requests	RW found in the inco mes have no effect ne is detected, RXB RW ad of LEUn_RX. connected to and reconnected	Start-Frame UnBlock RX Iming data. The start-frame is loaded into the receive buffer. If on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer Loopback Enable Interest and Interest								
6	Clears RXBLOC Value 0 1 LOOPBK Set to connect re Value 0 1 ERRSDMA When set,RX DM Value 0 1 INV	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0 Description The receiver is co The receiver is co O MA requests will be cleared Description Framing and parit RX DMA requests 0	RW found in the inco mes have no effect ne is detected, RXB RW ad of LEUn_RX. connected to and reconnected	Start-Frame UnBlock RX Iming data. The start-frame is loaded into the receive buffer. It on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer Loopback Enable Ideives data from LEUn_RX Reives data from LEUn_TX Clear RX DMA On Error It parity errors. Iffect on DMA requests from the LEUART If are disabled if a framing error or parity error occurs. Invert Input And Output								
6	Clears RXBLOC Value 0 1 LOOPBK Set to connect re Value 0 1 ERRSDMA When set,RX DM Value 0 1 INV Set to invert the	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0 Description The receiver is co The receiver is co O MA requests will be cleared Description Framing and parit RX DMA requests 0 output on LEUn_TX and in Description	RW found in the inco mes have no effect ne is detected, RXB RW ad of LEUn_RX. connected to and reconnected	Start-Frame UnBlock RX Iming data. The start-frame is loaded into the receive buffer. If on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer Loopback Enable Interest and Interest								
6	Clears RXBLOC Value 0 1 LOOPBK Set to connect re Value 0 1 ERRSDMA When set,RX DM Value 0 1 INV Set to invert the Value	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0 Description The receiver is co The receiver is co O MA requests will be cleared Description Framing and parit RX DMA requests 0 output on LEUn_TX and in Description A high value on th	RW found in the inco mes have no effect ne is detected, RXB RW ad of LEUn_RX. connected to and reconnected t	Start-Frame UnBlock RX Iming data. The start-frame is loaded into the receive buffer. If on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer Loopback Enable Interest and Interest								
6	Clears RXBLOC Value 0 1 LOOPBK Set to connect re Value 0 1 ERRSDMA When set,RX DM Value 0 1 INV Set to invert the Value	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0 Description The receiver is co The receiver is co O MA requests will be cleared Description Framing and parit RX DMA requests 0 output on LEUn_TX and in Description A high value on th	RW found in the inco mes have no effect ne is detected, RXB RW ad of LEUn_RX. connected to and reconnected t	Start-Frame UnBlock RX Iming data. The start-frame is loaded into the receive buffer. It on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer Loopback Enable Interverse data from LEUn_RX Interverse data from LEUn_TX Clear RX DMA On Error I parity errors. Interverse data from the LEUART I are disabled if a framing error or parity error occurs. Invert Input And Output X. I and a low value is 0.								
7 6 5	Clears RXBLOC Value 0 1 LOOPBK Set to connect re Value 0 1 ERRSDMA When set,RX DM Value 0 1 INV Set to invert the Value 0 1 STOPBITS	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0 Description The receiver is co The receiver is co O MA requests will be cleared Description Framing and parit RX DMA requests 0 output on LEUn_TX and in Description A high value on the A low value on the	RW found in the inco mes have no effect ne is detected, RXB RW ad of LEUn_RX. connected to and reco nected	Start-Frame UnBlock RX Iming data. The start-frame is loaded into the receive buffer. It on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer Loopback Enable Interest and a from LEUn_TX Interest a framing error or parity error occurs. Invert Input And Output X. In and a low value is 0. In and a high value is 0. In and a high value is 0.								
7 6 5	Clears RXBLOC Value 0 1 LOOPBK Set to connect re Value 0 1 ERRSDMA When set,RX DM Value 0 1 INV Set to invert the Value 0 1 STOPBITS	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0 Description The receiver is co The receiver is co O MA requests will be cleared Description Framing and parit RX DMA requests 0 output on LEUn_TX and in Description A high value on the A low value on the	RW found in the inco mes have no effect ne is detected, RXB RW ad of LEUn_RX. connected to and reco needed	Start-Frame UnBlock RX Iming data. The start-frame is loaded into the receive buffer. It on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer Loopback Enable Relives data from LEUn_RX Relives data from LEUn_TX Clear RX DMA On Error I parity errors. Invert Input And Output X. and a low value is 0. Stop-Bit Mode								
7 6 5	Clears RXBLOC Value 0 1 LOOPBK Set to connect re Value 0 1 ERRSDMA When set,RX DM Value 0 1 INV Set to invert the Value 0 1 STOPBITS Determines the re	will result in the M 0 K when the start-frame is Description Detected start-fram When a start-fram 0 Description The receiver is co The receiver is co O MA requests will be cleared Description Framing and parit RX DMA requests 0 Description A high value on the A low value on the 0 number of stop-bits used.	RW found in the inco mes have no effect he is detected, RXB RW ad of LEUn_RX. connected to and reconnected	Start-Frame UnBlock RX Iming data. The start-frame is loaded into the receive buffer. It on RXBLOCK BLOCK is cleared and the start-frame is loaded into the receive buffer Loopback Enable Relives data from LEUn_RX Relives data from LEUn_TX Clear RX DMA On Error It parity errors. Invert Input And Output X. Invert Input And Output X. Stop-Bit Mode transmitting data. The receiver only verifies that one stop bit is present.								



Bit	Name	Reset	Access	Description								
Dit		Neset	Access	Description								
3:2	PARITY	0x0	RW	Parity-Bit Mode								
	Determines who	ether parity bits are enable	d, and whether	even or odd parity should be used.								
	Value	Mode	С	Description								
	0	NONE	P	Parity bits are not used								
	2	EVEN	E	even parity are used. Parity bits are automatically generated and checked by hardware.								
	3	ODD	C	odd parity is used. Parity bits are automatically generated and checked by hardware.								
1	DATABITS	0	RW	Data-Bit Mode								
	This register se	ets the number of data bits.										
	Value	Mode	С	Description								
	0	EIGHT	E	ach frame contains 8 data bits								
	1	NINE	E	ach frame contains 9 data bits								
0	AUTOTRI	0	RW	Automatic Transmitter Tristate								
	When set, LEU	n_TX is tristated whenever	the transmitter	r is inactive.								
	Value	Description										
	0	LEUn_TX is held	LEUn_TX is held high when the transmitter is inactive. INV inverts the inactive state.									
	1	LEUn_TX is trista	ated when the trai	nsmitter is inactive								

18.5.2 LEUARTn_CMD - Command Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19) .

Offset			Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	œ	7	9	2	4	က	2	-	0
Reset											•		•		,			•			•		,		0	0	0	0	0	0	0	0
Access																									W1	W1	W	W	X	M	W1	W1
Name																									CLEARRX	CLEARTX	RXBLOCKDIS	RXBLOCKEN	TXDIS	TXEN	RXDIS	RXEN

Bit	Name	Reset	Access	Description							
31:8	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)							
7	CLEARRX	0	W1	Clear RX							
	Set to clear receive b	uffer and the RX shif	ft register.								
6	CLEARTX	0	W1	Clear TX							
	Set to clear transmit I	buffer and the TX shi	ft register.								
5	RXBLOCKDIS	0	W1	Receiver Block Disable							
	Set to clear RXBLOCK, resulting in all incoming frames being loaded into the receive buffer.										
4	RXBLOCKEN	0	W1	Receiver Block Enable							
	Set to set RXBLOCK	, resulting in all incor	ming frames being	g discarded.							
3	TXDIS	0	W1	Transmitter Disable							
	Set to disable transm	ission.									
2	TXEN	0	W1	Transmitter Enable							
	Set to enable data tra	ansmission.									
1	RXDIS	0	W1	Receiver Disable							
	Set to disable data re	eception. If a frame is	under reception	when the receiver is disabled, the incoming frame is discarded.							
0	RXEN	0	W1	Receiver Enable							



4

TXENS

Bit	Name	Reset	Access	Description
	Set to activate data recept	ion on LEUn_RX.		

18.5.3 LEUARTn_STATUS - Status Register

Offset															Bi	t Po	siti	on														
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	0	80	7	9	2	4	က	2	-	0
Reset				•					•	•	•				•			•	•						,	•	0	-	0	0	0	0
Access																											œ	22	2	22	~	<u>~</u>
Name																											RXDATAV	TXBL	TXC	RXBLOCK	TXENS	RXENS
Bit	Na	ame						Re	eset			A	\cc	ess		De	scr	ipti	on													
31:6	Re	serv	ed					То	ens	ure c	comp	atib	ility	with	futu	ire d	evice	es, a	alwa	ays v	vrite	bits	to 0.	Mor	e int	orm	natio	n in (Sect	ion 2	2.1 (p	o. 3)
5	RX	DAT	AV					0				R				RX	Dat	a Va	alid													

TVC	Λ	D	TV Complete
Indicates the level of the tra	ansmit buffer. Set w	hen the transm	nit buffer is empty, and cleared when it is full.
IXBL	1	ĸ	IX Buffer Level

Set when a transmission has completed and no more data is available in the transmit buffer. Cleared when a new transmission starts.

RXBLOCK 0 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.

Set when the transmitter is enabled.

0 RXENS 0 R Receiver Enable Status

Set when data is available in the receive buffer. Cleared when the receive buffer is empty.

Set when the receiver is enabled. The receiver must be enabled for start frames, signal frames, and multi-processor address bit detection.

Transmitter Enable Status

18.5.4 LEUARTn_CLKDIV - Clock Control Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

Offset															Bi	t Po	siti	on														
0x00C	31	30	29	28	27	56	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	8	7	9	2	4	က	2	-	0
Reset																							0	00000								
Access																							i	À								
Name																							Ž	2								

Bit	Name	Reset	Access	Description
31:15	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
14:3	DIV	0x000	RW	Fractional Clock Divider
	Specifies the fractional c	lock divider for the	ELEUART.	
2:0	Reserved	To ensure co.	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)



18.5.5 LEUARTn_STARTFRAME - Start Frame Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

Offset								,	,						Bi	t Pc	siti	on					,									
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	∞	7	9	2	4	က	2	-	0
Reset																												000x0				
Access																												ΑŠ				
Name																												STARTFRAME				

Bit	Name	Reset	Access	Description
31:9	Reserved	To ensure co	ompatibility with f	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
8:0	STARTFRAME	0x000	RW	Start Frame
	When a frame matchin	0	,	receiver, STARTF interrupt flag is set, and if SFUBRX is set, RXBLOCK

18.5.6 LEUARTn_SIGFRAME - Signal Frame Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

Offset															Bi	t Po	siti	on														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	8	7	9	2	4	3	2	-	0
Reset																												000x0				
Access																												R				
Name																												SIGFRAME				

Bit	Name	Reset	Access	Description
31:9	Reserved	To ensure co	mpatibility with f	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
8:0	SIGFRAME	0x000	RW	Signal Frame
	When a frame matchi	ng SIGFRAME is de	tected by the rec	eiver, SIGF interrupt flag is set.

18.5.7 LEUARTn_RXDATAX - Receive Buffer Data Extended Register

Offset															Bi	t Po	siti	on														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	80	7	9	2	4	3	2	-	0
Reset																	0	0										000x0				
Access																	2	~										~				
Name																	FERR	PERR										RXDATA				



Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure comp	patibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15	FERR	0	R	Receive Data Framing Error
	Set if data in buffer has a	framing error. Can	be the result o	f a break condition.
14	PERR	0	R	Receive Data Parity Error
	Set if data in buffer has a	parity error.		
13:9	Reserved	To ensure comp	patibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
8:0	RXDATA	0x000	R	RX Data
	Use this register to access	s data read from the	e LEUART. Bu	ffer is cleared on read access.

18.5.8 LEUARTn_RXDATA - Receive Buffer Data Register

Offset															Bi	t Po	siti	on														
0x01C	31	30	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	41	13	12	7	10	6	∞	7	9	2	4	က	2	-	0
Reset																													0x00			
Access																													ď			
Name																													RXDATA			

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure compa	tibility with futu	re devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:0	RXDATA	0x00	R	RX Data
	Use this register to access	data read from LEU.	ART. Buffer is	cleared on read access. Only the 8 LSB can be read using this register.

18.5.9 LEUARTn_RXDATAXP - Receive Buffer Data Extended Peek Register

Offset															Bi	t Po	siti	on														
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	1	10	6	8	7	9	2	4	က	2	-	0
Reset																0	0										000x0					
Access																	œ	~										ď				
Name																	FERRP	PERRP										RXDATAP				

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15	FERRP	0	R	Receive Data Framing Error Peek
	Set if data in buffe	r has a framing error. C	an be the result o	f a break condition.
14	PERRP	0	R	Receive Data Parity Error Peek
	Set if data in buffe	r has a parity error.		



Bit	Name	Reset	Access	Description
13:9	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
8:0	RXDATAP	0x000	R	RX Data Peek
	Use this register to acces	ss data read from	the LEUART.	

18.5.10 LEUARTn_TXDATAX - Transmit Buffer Data Extended Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19) .

Offset															Bi	t Po	siti	on								-						
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	8	7	9	2	4	က	2	-	0
Reset																	0	0	0			•						000x0				
Access																	>	≥	>									≥				
Name																	RXENAT	TXDISAT	TXBREAK									TXDATA				

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15	RXENAT	0	W	Enable RX After Transmission
	Set to enable rece	ption after transmission	1.	
	Value	Description		
	0	-		
	1	The receiver is e	nabled, setting RXE	NS after the frame has been transmitted
14	TXDISAT	0	W	Disable TX After Transmission
	Set to disable trans	smitter directly after tra	nsmission has cor	npeted.
	Value	Description		
	0	-		
	1	The transmitter is	s disabled, clearing l	XENS after the frame has been transmitted
13	TXBREAK	0	W	Transmit Data As Break
	Set to send data a of TXDATA.	as a break. Recipient w	vill see a framing	error or a break condition depending on its configuration and the value
	Value	Description		
	0	The specified nu	mber of stop-bits are	transmitted
	1		dinary stop-bits, 0 is r to detect the start of	transmitted to generate a break. A single stop-bit is generated after the break to f the next frame
12:9	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3,
12.0				
8:0	TXDATA	0x000	W	TX Data

18.5.11 LEUARTn_TXDATA - Transmit Buffer Data Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).



Offset															Bi	t Po	siti	on														
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	8	7	9	2	4	3	2	-	0
Reset																													0x0			
Access																													}			
Name																													TXDATA			

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure comp	atibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:0	TXDATA	0x00	W	TX Data
	This frame will be added	to the transmit buffe	r. Only 8 LSB	can be written using this register. 9th bit and control bits will be cleared.

18.5.12 LEUARTn_IF - Interrupt Flag Register

Offset					·										Bi	t Pc	siti	on														
0x02C	31	30	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	41	13	12	=	10	6	8	7	9	2	4	ю	2	-	0
Reset																						0	0	0	0	0	0	0	0	0	-	0
Access																					-	~	~	~	~	œ	~	~	~	~	~	<u>~</u>
Name																						SIGF	STARTF	MPAF	FERR	PERR	TXOF	RXUF	RXOF	RXDATAV	TXBL	TXC

Bit	Name	Reset	Access	Description
31:11	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
10	SIGF	0	R	Signal Frame Interrupt Flag
	Set when a signal	frame is detected.		
9	STARTF	0	R	Start Frame Interrupt Flag
	Set when a start fi	rame is detected.		
8	MPAF	0	R	Multi-Processor Address Frame Interrupt Flag
	Set when a multi-p	processor address frame	e is detected.	
7	FERR	0	R	Framing Error Interrupt Flag
	Set when a frame	with a framing error is re	eceived while RX	BLOCK is cleared.
6	PERR	0	R	Parity Error Interrupt Flag
	Set when a frame	with a parity error is rec	eived while RXBL	LOCK is cleared.
5	TXOF	0	R	TX Overflow Interrupt Flag
	Set when a write i	s done to the transmit b	uffer while it is full	I. The data already in the transmit buffer is preserved.
4	RXUF	0	R	RX Underflow Interrupt Flag
	Set when trying to	read from the receive b	uffer when it is er	mpty.
3	RXOF	0	R	RX Overflow Interrupt Flag
	Set when data is i	ncoming while the receiv	ve shift register is	full. The data previously in shift register is overwritten by the new data.
2	RXDATAV	0	R	RX Data Valid Interrupt Flag
	Set when data bed	comes available in the re	eceive buffer.	
1	TXBL	1	R	TX Buffer Level Interrupt Flag
	Set when space b	ecomes available in the	transmit buffer fo	or a new frame.



Bit	Name	Reset	Access	Description
0	TXC	0	R	TX Complete Interrupt Flag
	Set after a transmission wh	en both the TX buf	fer and shift re	gister are empty.

18.5.13 LEUARTn_IFS - Interrupt Flag Set Register

Offset															Bi	t Po	siti	on														
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	8	7	9	2	4	က	2	-	0
Reset									•						•							0	0	0	0	0	0	0	0			0
Access																						٧ ا	W 1	W1	W1	W1	W1	W1	W1			W1
Name																						SIGF	STARTF	MPAF	FERR	PERR	TXOF	RXUF	RXOF			TXC

Bit	Name	Reset	Access	Description
31:11	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
10	SIGF	0	W1	Set Signal Frame Interrupt Flag
	Write to 1 to set the	SIGF interrupt flag.		
9	STARTF	0	W1	Set Start Frame Interrupt Flag
	Write to 1 to set the	STARTF interrupt flag	ı .	
8	MPAF	0	W1	Set Multi-Processor Address Frame Interrupt Flag
	Write to 1 to set the	MPAF interrupt flag.		
7	FERR	0	W1	Set Framing Error Interrupt Flag
	Write to 1 to set the	e FERR interrupt flag.		
6	PERR	0	W1	Set Parity Error Interrupt Flag
	Write to 1 to set the	e PERR interrupt flag.		
5	TXOF	0	W1	Set TX Overflow Interrupt Flag
	Write to 1 to set the	e TXOF interrupt flag.		
4	RXUF	0	W1	Set RX Underflow Interrupt Flag
	Write to 1 to set the	e RXUF interrupt flag.		
3	RXOF	0	W1	Set RX Overflow Interrupt Flag
	Write to 1 to set the	e RXOF interrupt flag.		
2:1	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	TXC	0	W1	Set TX Complete Interrupt Flag
	Write to 1 to set the	e TXC interrupt flag.		

18.5.14 LEUARTn_IFC - Interrupt Flag Clear Register

Offset															Bi	t Pc	siti	on														
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	-	10	6	œ	7	9	2	4	ю	2	-	0
Reset		•						•														0	0	0	0	0	0	0	0			0
Access																						M	W1	W1	W1	W1	W1	W1	W W			M
Name																						SIGF	STARTF	MPAF	FERR	PERR	TXOF	RXUF	RXOF			TXC



Bit	Name	Reset	Access	Description
31:11	Reserved	To ensure co	mpatibility with fu	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
10	SIGF	0	W1	Clear Signal-Frame Interrupt Flag
	Write to 1 to clear t	he SIGF interrupt flag.		
9	STARTF	0	W1	Clear Start-Frame Interrupt Flag
	Write to 1 to clear t	he STARTF interrupt fla	ag.	
8	MPAF	0	W1	Clear Multi-Processor Address Frame Interrupt Flag
	Write to 1 to clear t	he MPAF interrupt flag		
7	FERR	0	W1	Clear Framing Error Interrupt Flag
	Write to 1 to clear t	he FERR interrupt flag.		
6	PERR	0	W1	Clear Parity Error Interrupt Flag
	Write to 1 to clear t	he PERR interrupt flag.		
5	TXOF	0	W1	Clear TX Overflow Interrupt Flag
	Write to 1 to clear t	he TXOF interrupt flag.		
4	RXUF	0	W1	Clear RX Underflow Interrupt Flag
	Write to 1 to clear t	he RXUF interrupt flag.		
3	RXOF	0	W1	Clear RX Overflow Interrupt Flag
	Write to 1 to clear t	he RXOF interrupt flag.		
2:1	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	TXC	0	W1	Clear TX Complete Interrupt Flag
	Write to 1 to clear t	he TXC interrupt flag.		

18.5.15 LEUARTn_IEN - Interrupt Enable Register

Offset															Bi	t Po	siti	on														
0x038	31	98	59	78	27	26	52	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	æ	7	9	2	4	ю	2	-	0
Reset																						0	0	0	0	0	0	0	0	0	0	0
Access																						R₩	RW	RW	RW W	RW	RW	R W	₩ M	RW	W.	RW
Name																						SIGF	STARTF	MPAF	FERR	PERR	TXOF	RXUF	RXOF	RXDATAV	TXBL	TXC

Bit	Name	Reset	Access	Description
31:11	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
10	SIGF	0	RW	Signal Frame Interrupt Enable
	Enable interrupt on sign	nal frame.		
9	STARTF	0	RW	Start Frame Interrupt Enable
	Enable interrupt on star	t frame.		
8	MPAF	0	RW	Multi-Processor Address Frame Interrupt Enable
	Enable interrupt on mul	ti-processor addre	ss frame.	
7	FERR	0	RW	Framing Error Interrupt Enable
	Enable interrupt on fram	ning error.		
6	PERR	0	RW	Parity Error Interrupt Enable
	Enable interrupt on pari	ty error.		
5	TXOF	0	RW	TX Overflow Interrupt Enable
	Enable interrupt on TX	overflow.		



Bit	Name	Reset	Access	Description
4	RXUF	0	RW	RX Underflow Interrupt Enable
	Enable interrupt or	n RX underflow.		
3	RXOF	0	RW	RX Overflow Interrupt Enable
	Enable interrupt or	n RX overflow.		
2	RXDATAV	0	RW	RX Data Valid Interrupt Enable
	Enable interrupt or	n RX data.		
1	TXBL	0	RW	TX Buffer Level Interrupt Enable
	Enable interrupt or	n TX buffer level.		
0	TXC	0	RW	TX Complete Interrupt Enable
	Enable interrupt or	n TX complete.		

18.5.16 LEUARTn_PULSECTRL - Pulse Control Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19) .

Offset															Bi	t Pc	siti	on					,									
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	7	10	6	8	7	9	2	4	က	2	-	0
Reset			•		•										•						-		•				0	0			0×0	
Access																											RW	RW			χ ≷	
Name																											PULSEFILT	PULSEEN			PULSEW	

Bit	Name	Reset	Access	Description
31:6	Reserved	To ensure com	patibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
5	PULSEFILT	0	RW	Pulse Filter
	Enable a one-cycle pulse	e filter for pulse exte	ender	
	Value	Description		
	0	Filter is disabled. Po	ulses must be at le	east 2 cycles long for reliable detection.
	1	Filter is enabled. Pu	ilses must be at le	east 3 cycles long for reliable detection.
4	PULSEEN	0	RW	Pulse Generator/Extender Enable
	Filter LEUART output the	rough pulse generat	or and the LEU	ART input through the pulse extender.
3:0	PULSEW	0x0	RW	Pulse Width
	Configure the pulse widt	h of the pulse gener	rator as a numb	er of 32.768 kHz clock cycles.

18.5.17 LEUARTn_FREEZE - Freeze Register

Offset															Bi	t Po	siti	on														
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=======================================	10	6	∞	7	9	2	4	က	2	-	0
Reset																			•		•								•			0
Access																																RW
Name																																REGFREEZE



Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	REGFREEZE	0	RW	Register Update Freeze
	When set, the up	date of the LEUART is pe	ostponed until this	bit is cleared. Use this bit to update several registers simultaneously.
	Value	Mode	Des	cription
	0	UPDATE		h write access to a LEUART register is updated into the Low Frequency domain oon as possible.
	1	FREEZE	The	LEUART is not updated with the new written value.

18.5.18 LEUARTn_SYNCBUSY - Synchronization Busy Register

Offset													Bi	t Po	siti	on														
0x044	31	8	50 50	27	26	25 25	23	22	21	20	19	18	17	16	15	4	13	12	11	10	0	æ	7	9	2	4	က	2	-	0
Reset		-					·	-					•						•		•		0	0	0	0	0	0	0	0
Access																							œ	~	œ	22	œ	22	œ	~
Name																							PULSECTRL	TXDATA	TXDATAX	SIGFRAME	STARTFRAME	CLKDIV	CMD	CTRL

D:4	Nama	Decet	A	Description
Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7	PULSECTRL	0	R	PULSECTRL Register Busy
	Set when the value wri	tten to PULSECTR	L is being synchro	onized.
6	TXDATA	0	R	TXDATA Register Busy
	Set when the value wri	tten to TXDATA is I	being synchronize	ed.
5	TXDATAX	0	R	TXDATAX Register Busy
	Set when the value wri	tten to TXDATAX is	s being synchroniz	zed.
4	SIGFRAME	0	R	SIGFRAME Register Busy
	Set when the value wri	tten to SIGFRAME	is being synchror	nized.
3	STARTFRAME	0	R	STARTFRAME Register Busy
	Set when the value wri	tten to STARTFRA	ME is being synch	nronized.
2	CLKDIV	0	R	CLKDIV Register Busy
	Set when the value wri	tten to CLKDIV is b	eing synchronize	d.
1	CMD	0	R	CMD Register Busy
	Set when the value wri	tten to CMD is bein	g synchronized.	
0	CTRL	0	R	CTRL Register Busy
	Set when the value wri	tten to CTRL is bei	ng synchronized.	



18.5.19 LEUARTn_ROUTE - I/O Routing Register

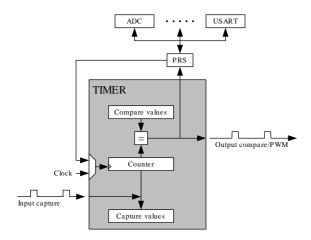
Offset	Bit Position																															
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	œ	7	9	2	4	က	7	-	0
Reset		,			•																•	•	3	OX O						,	0	0
Access																							3	} Ł							RW	RW
Name																							Q F	5							TXPEN	RXPEN

							LOC,		ř	2
Bit	Name	R	eset	Acce	ss Description					
31:10	Reserved	T	o ensure coi	mpatibility w	vith future devices, always	write bits to	0. Mor	e information in Sec	ction 2.1 (p.	. 3)
9:8	LOCATION	0x	0x0 RW I/O Location							
	Decides the loc	cation of the LEU	JART I/O pii	ns.						
	Value	Mode			Description					
	0	LOC0			Location 0					
	1	LOC1			Location 1					
	2	LOC2			Location 2					
7:2	Reserved	T	o ensure co	mpatibility w	vith future devices, always	write bits to	0. Mor	e information in Sec	ction 2.1 (p.	. 3)
7:2	Reserved TXPEN	0	o ensure col	mpatibility w	vith future devices, always TX Pin Enable	write bits to	0. Mor	e information in Sec	ction 2.1 (p.	. 3)
7:2	TXPEN			RW		write bits to	O. Mor	e information in Sec	ction 2.1 (p.	. 3)
7:2	TXPEN	0 TX pin of the LE		RW		write bits to	0. Mor	e information in Sec	ction 2.1 (p.	. 3)
7:2	TXPEN When set, the	0 TX pin of the LE	UART is en	RW abled.		write bits to	O. Mor	e information in Sec	ction 2.1 (p.	. 3)
7:2	TXPEN When set, the	0 TX pin of the LE Desc	UART is en	RW abled.		write bits to	0. Mor	e information in Sec	ction 2.1 (p.	. 3)
7:2	TXPEN When set, the	0 TX pin of the LE Desc	UART is endiction	RW abled.		write bits to	0. Mor	e information in Sec	ction 2.1 (p.	. 3)
1	TXPEN When set, the Value 0 1 RXPEN	0 TX pin of the LE Desc The	UART is endeription LEUn_TX pin LEUn_TX pin	RW abled. is disabled is enabled	TX Pin Enable	write bits to	0. Mor	e information in Sec	ction 2.1 (p.	. 3)
1	TXPEN When set, the Value 0 1 RXPEN	0 TX pin of the LE Desc The The 0 RX pin of the LE	UART is endeription LEUn_TX pin LEUn_TX pin	RW abled. is disabled is enabled	TX Pin Enable	write bits to	0. Mor	e information in Sec	ction 2.1 (p.	. 3)
1	TXPEN When set, the Value 0 1 RXPEN When set, the	0 TX pin of the LE Desc The The 0 RX pin of the LE	UART is enscription LEUn_TX pin LEUn_TX pin	RW abled. is disabled is enabled RW abled.	TX Pin Enable	write bits to	0. Mor	e information in Sec	ction 2.1 (p.	3)



19 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-bit TIMER can be configured to provide PWM waveforms with optional dead-time insertion for 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 reduce energy consumption.

19.1 Introduction

The 16-bit general purpose Timer has 3 compare/capture channels for input capture and compare/Pulse-Width Modulation (PWM) output. TIMER0 also includes a Dead-Time Insertion module suitable for motor control applications.

19.2 Features

- 16-bit auto reload up/down counter
 - Dedicated 16-bit reload register which serves as counter maximum
- 3 Compare/Capture channels
 - Individual 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
- 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



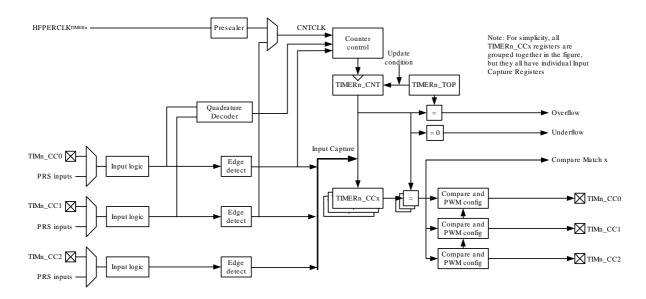
- · Pulse width measurement
- Two capture registers for each capture channel
 - Capture on either positive or negative edge
 - · Capture on both edges
- · Optional digital noise filtering on capture inputs
- Output Compare
 - Compare output toggle/pulse on compare match
 - Immediate update of compare 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
- · Clock sources
 - HFPERCLK_{TIMERn}
 - 10-bit Prescaler
 - External pin
 - Peripheral Reflex System
- Debug mode
 - Configurable to either run or stop when processor is stopped (break)
- Interrupts, PRS output and/or DMA request
 - Underflow
 - Overflow
 - Compare/Capture event
- Dead-Time Insertion Unit (TIMER0 only)
 - · 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
 - · Configuration lock

19.3 Functional Description

An overview of the TIMER module is shown in Figure 19.1 (p. 251). The Timer module consists of a 16 bit up/down counter with 3 Compare/Capture channels connected to pins TIMn_CC0, TIMn_CC1, and TIMn_CC2.



Figure 19.1. TIMER Block Overview



19.3.1 Counter Modes

The Timer consists of a counter that can be configured to the following modes:

- 1. Up-count: Counter counts up until it reaches the value in TIMERn_TOP, where it is reset to 0 before counting up again.
- 2. 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.
- 3. 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.
- 4. Quadrature Decoder: Two input channels where one determines the count direction, while the other pin triggers a clock event.

The counter value can be read or written by software at any time by accessing the CNT field in TIMERn_CNT.

19.3.1.1 Events

Overflow is set when the counter value shifts from TIMERn_TOP to the next value when counting up. In up-count mode the next value is 0. In up/down-count mode, the next value is TIMERn_TOP-1.

Underflow is set when the counter value shifts from 0 to the next value when counting down. In down-count mode, the next value is TIMERn_TOP. In up/down-count mode the next value is 1.

Update event is set on overflow in up-count mode and on underflow in down-count or up/down count mode. This event is used to time updates of buffered values.

19.3.1.2 Operation

Figure 19.2 (p. 252) shows the hardware Timer/Counter control. Software can start or stop the counter by writing a 1 to the START or STOP bits in TIMERn_CMD. The counter value (CNT in TIMERn_CNT) can always be written by software to any 16-bit value.

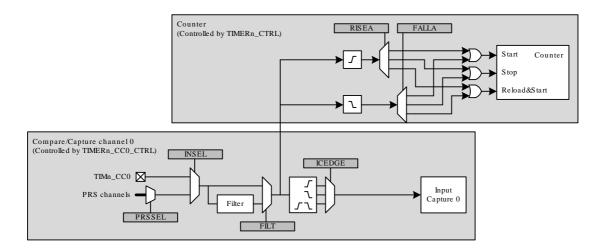
It is also possible to control the counter through either an external pin or PRS input. This is done through the input logic for the Compare/Capture Channel 0. The Timer/Counter allows individual actions (start, stop, reload) to be taken for rising and falling input edges. This is configured in the RISEA 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_CTRL 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.

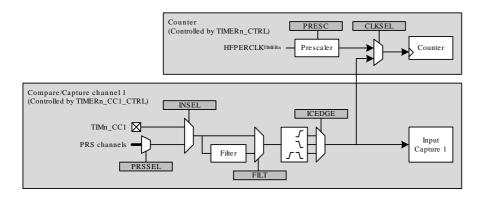
Figure 19.2. TIMER Hardware Timer/Counter Control



19.3.1.3 Clock Source

The counter can be clocked from several sources, which are all synchronized with the peripheral clock (HFPERCLK). See Figure 19.3 (p. 252) .

Figure 19.3. TIMER Clock Selection



19.3.1.3.1 Peripheral Clock (HFPERCLK)

The peripheral clock (HFPERCLK) can be used as a source with a configurable prescale factor of 2^PRESC, where PRESC is an integer between 0 and 10, which is set in PRESC in TIMERn_CTRL. The prescaler is stopped and reset when the timer is stopped.

19.3.1.3.2 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

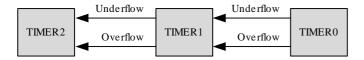


must not have a higher frequency than f_{HFPERCLK}/3 when running from a pin input or a PRS input with FILT enabled in TIMERn_CCx_CTRL. When running from PRS without FILT, the frequency can be as high as f_{HFPERCLK}. Note that when clocking the Timer from the same pulse that triggers a start (through RISEA/FALLA in TIMERn_CTRL), the starting pulse will not update the Counter Value.

19.3.1.3.3 Underflow/Overflow from Neighboring Timer

All Timers are linked together (see Figure 19.4 (p. 253)), allowing timers to count on overflow/underflow from the lower numbered neighbouring timers to form a 32-bit or 48-bit timer. Note that all timers must be set to same count direction and less significant timer(s) can only be set to count up or down.

Figure 19.4. TIMER Connections



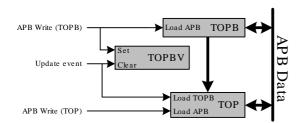
19.3.1.4 One-Shot Mode

By default, the counter counts continuously until it is stopped. If the OSMEN bit is set in the TIMERn_CTRL register, however, the counter is disabled by hardware on the first *update event*. Note that when the counter is running with CC1 as clock source (0b01 in CLKSEL in TIMERn_CTRL) and OSMEN is set, a CC1 capture event will not take place on the *update event* (CC1 rising edge) that stops the Timer.

19.3.1.5 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 have not yet been written to the TIMERn_TOP register (see Figure 19.5 (p. 253) .

Figure 19.5. TIMER TOP Value Update Functionality

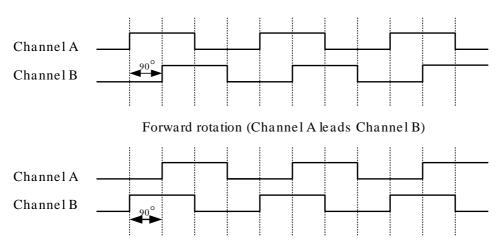


19.3.1.6 Quadrature Decoder

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 19.6 (p. 254)).



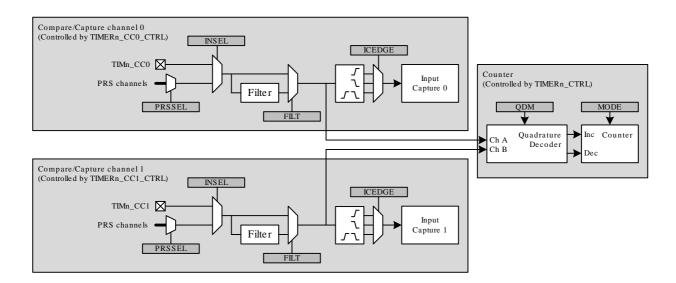
Figure 19.6. TIMER Quadrature Encoded Inputs



Backward rotation (Channel B leads Channel A)

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 Quadrature Decoder Mode 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.

Figure 19.7. 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_CTRL. See Figure 19.7 (p. 254)

19.3.1.6.1 X2 Decoding Mode

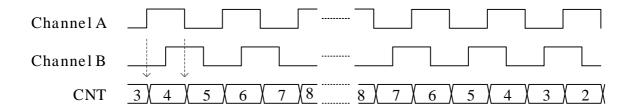
In X2 Decoding mode, the counter increments or decrements on every edge of Channel A, see Table 19.1 (p. 255) and Figure 19.8 (p. 255).



Table 19.1. TIMER Counter Response in X2 Decoding Mode

Channel B	Chan	nel A
Cildille	Rising	Falling
0	Increment	Decrement
1	Decrement	Increment

Figure 19.8. TIMER X2 Decoding Mode



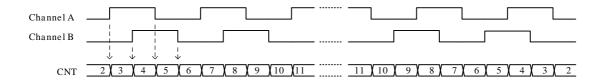
19.3.1.6.2 X4 Decoding Mode

In X4 Decoding mode, the counter increments or decrements on every edge of Channel A and Channel B, see Figure 19.9 (p. 255) and Table 19.2 (p. 255) .

Table 19.2. TIMER Counter Response in X4 Decoding Mode

Opposite Channel	Chan	nel A	Chan	nel 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 19.9. TIMER X4 Decoding Mode



19.3.1.6.3 TIMER Rotational Position

To calculate a position Equation 19.1 (p. 255) can be used.

TIMER Rotational Position Equation
$$pos^{\circ} = (CNT/X \times N) \times 360^{\circ}$$
 (19.1)

where X =Encoding type and N =Number of pulses per revolution.

19.3.2 Compare/Capture Channels

The Timer contains 3 Compare/Capture channels, which can be configured in the following modes:

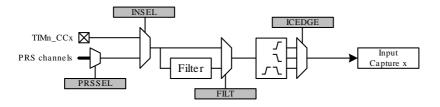


- 1. Input Capture
- 2. Output Compare
- 3. PWM

19.3.2.1 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 19.10 (p. 256)). Compare/Capture channels 0 and 1 are the inputs for the Quadrature Decoder Mode. The input channel can be filtered before it is used, which requires the input to remain stable for 5 cycles in a row before the input is propagated to the output.

Figure 19.10. TIMER Input Pin Logic



19.3.2.2 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), the behavior of the Compare/Capture registers (TIMERn_CCx_CCV) and buffer registers (TIMERn_CCx_CCVB) change depending on the mode the channel is set in.

19.3.2.2.1 Input Capture mode

When running in Input Capture mode, TIMERn_CCx_CCV and TIMERn_CCx_CCVB form a FIFO buffer, and new capture values are added on a capture event, see Figure 19.11 (p. 257). The first capture can always be read from TIMERn_CCx_CCV, and reading this address will load the next capture value into TIMERn_CCx_CCV from TIMERn_CCx_CCVB if it contains valid data. The CC value can be read without altering the FIFO contents by reading TIMERn_CCx_CCVP. TIMERn_CCx_CCVB can also be read without altering the FIFO contents. The ICV flag in TIMERn_STATUS indicates if there is a valid unread capture in TIMERn_CCx_CCV.

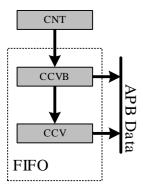
In case a capture is triggered while both CCV and CCVB contain unread capture values, the buffer overflow interrupt flag (ICBOF in TIMERn_IF) will be set. New capture values will on overflow overwrite the value in TIMERn_CCx_CCVB.

Note

In input capture mode, the timer will only trigger interrupts when it is running



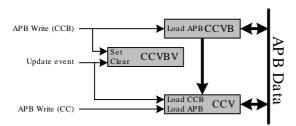
Figure 19.11. TIMER Input Capture Buffer Functionality



19.3.2.2.2 Compare and PWM Mode

When running in Output Compare or PWM mode, the value in TIMERn_CCx_CCV 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_CCV can be accessed directly or through the buffer register TIMERn_CCx_CCVB, see Figure 19.12 (p. 257). When writing to the buffer register, the value in TIMERn_CCx_CCVB will be written to TIMERn_CCx_CCV on the next update event. This functionality ensures glitch free PWM outputs. The CCVBV flag in TIMERn_STATUS indicates whether the TIMERn_CCx_CCVB register contains data that have not yet been written to the TIMERn_CCx_CCV register. Note that when writing 0 to TIMERn_CCx_CCVB the CCV 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 19.12. TIMER Output Compare/PWM Buffer Functionality

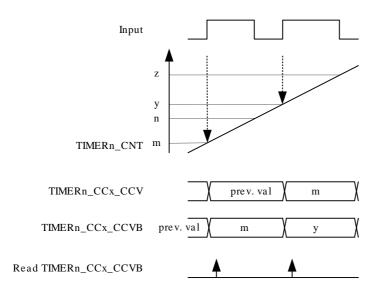


19.3.2.3 Input Capture

In Input Capture Mode, the counter value (TIMERn_CNT) can be captured in the Compare/Capture Register (TIMERn_CCx_CCV), see Figure 19.13 (p. 258). In this mode, TIMERn_CCx_CCV is read-only. Together with the Compare/Capture Buffer Register (TIMERn_CCx_CCVB) the TIMERn_CCx_CCV form a double-buffered capture registers allowing two subsequent capture events to take place before a read-out is required. The CCPOL bits in TIMERn_STATUS indicate the polarity the edge that triggered the capture in TIMERn_CCx_CCV.



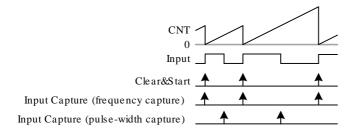
Figure 19.13. TIMER Input Capture



19.3.2.3.1 Period/Pulse-Width Capture

Period and/or pulse-width capture can be achieved by setting the RISEA field in TIMERn_CTRL to Clear&Start, and select the wanted input from either external pin or PRS, see Figure 19.14 (p. 258). For period capture, the Compare/Capture Channel 0 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 channel should be set to capture on a falling edge of the input signal. To start the measuring period on either a falling edge or measure the low pulse-width of a signal, opposite polarities should be chosen.

Figure 19.14. TIMER Period and/or Pulse width Capture

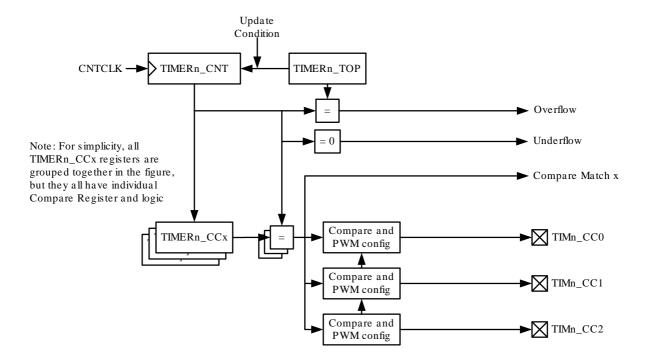


19.3.2.4 Compare

Each Compare/Capture channel contains a comparator which outputs a compare match if the contents of TIMERn_CCx_CCV matches the counter value, see Figure 19.15 (p. 259). 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 19.15. TIMER Block Diagram Showing Comparison Functionality

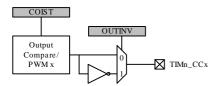


If occurring in the same cycle, match action will have priority over overflow or underflow action.

The input selected (through PRSSEL, INSEL and FILTSEL in TIMERn_CCx_CTRL) for the CC channel will also be sampled on compare match and the result is found in the CCPOL bits in TIMERn_STATUS.

The COIST bit in TIMERn_CCx_CTRL is the initial state of the compare/PWM output. 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 pulses on the output. The CC channel can be turned off by setting MODE to OFF in TIMER_CCx_CTRL.

Figure 19.16. TIMER Output Logic



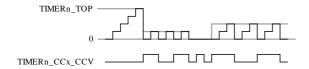
19.3.2.4.1 Frequency Generation (FRG)

Frequency generation (see Figure 19.17 (p. 260)) 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 19.17. TIMER Up-count Frequency Generation



The output frequency is given by Equation 19.2 (p. 260)

TIMER Up-count Frequency Generation Equation
$$f_{FRG} = f_{HFPERCLK} / (2^{(PRESC + 1) \times (TOP + 1) \times 2})$$
 (19.2)

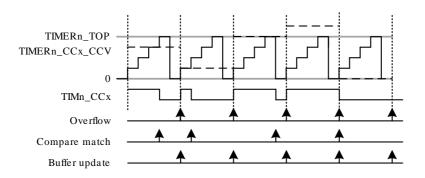
19.3.2.5 Pulse-Width Modulation (PWM)

In PWM mode, TIMERn_CCx_CCV 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.

19.3.2.6 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 19.18 (p. 260)). 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_CCV. This means that a constant high output is achieved by setting TIMER_CCx to TOP+1 or higher. The PWM resolution (in bits) is then given by Equation 19.3 (p. 260).

Figure 19.18. TIMER Up-count PWM Generation



TIMER Up-count PWM Resolution Equation

$$R_{PWM_{UD}} = log(TOP+1)/log(2)$$
 (19.3)

The PWM frequency is given by Equation 19.4 (p. 260):

TIMER Up-count PWM Frequency Equation
$$f_{PWM_{up/down}} = f_{HFPERCLK} / (2^{PRESC} \times (TOP + 1))$$
(19.4)

The high duty cycle is given by Equation 19.5 (p. 260)

TIMER Up-count Duty Cycle Equation

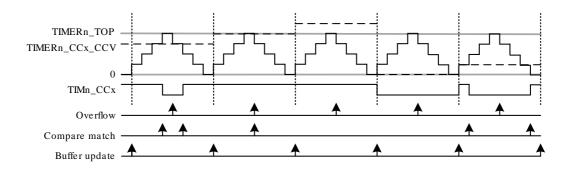


$$DS_{up} = CCVx/TOP (19.5)$$

19.3.2.7 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 19.19 (p. 261) .The resolution (in bits) is given by Equation 19.6 (p. 261) .

Figure 19.19. TIMER Up/Down-count PWM Generation



TIMER Up/Down-count PWM Resolution Equation

$$R_{PWM_{up/down}} = log(TOP+1)/log(2)$$
(19.6)

The PWM frequency is given by Equation 19.7 (p. 261):

TIMER Up/Down-count PWM Frequency Equation

$$f_{PWM_{up/down}} = f_{HFPERCLK} / (2^{PRESC+1}) \times TOP)$$
 (19.7)

The high duty cycle is given by Equation 19.8 (p. 261)

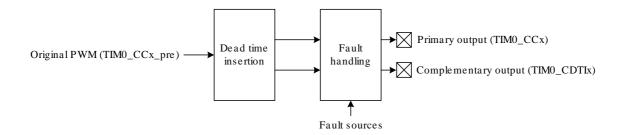
TIMER Up/Down-count Duty Cycle Equation

$$DS_{up/down} = CCVx/TOP (19.8)$$

19.3.3 Dead-Time Insertion Unit (TIMER0 only)

The Dead-Time Insertion Unit aims to make control of BLDC motors safer and more efficient by introducing complementary PWM outputs with dead-time insertion and fault handling, see Figure 19.20 (p. 261).

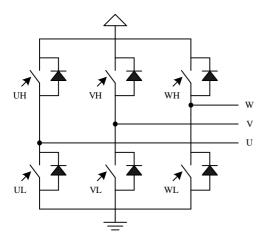
Figure 19.20. TIMER Dead-Time Insertion Unit Overview





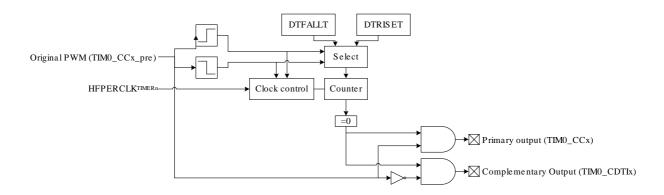
When used for motor control, the PWM outputs TIM0_CC0, TIM0_CC1 and TIM0_CC2 are often connected to the high-side transistors of a triple half-bridge setup (UH, VH and WH), and the complementary outputs connected to the respective low-side transistors (UL, VL, WL shown in Figure 19.21 (p. 262)). Transistors 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 19.21. 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 PMAC motors possible using only a single timer, see Figure 19.22 (p. 262).

Figure 19.22. TIMER Overview of Dead-Time Insertion Block for a Single PWM channel

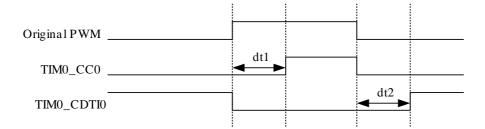


The DTI unit is enabled by setting DTEN in TIMER0_DTCTRL. In addition to providing the complementary outputs, the DTI unit then also overrides the compare match outputs from the timer.

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 19.23 (p. 263).



Figure 19.23. 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_{TIMERn} by a configurable factor between 1 and 1024, which is set in the DTPRESC field in TIMER0_DTTIME. The rising and falling edge dead-times are configured in DTRISET and DTFALLT in TIMER0_DTTIME to any number between 1-64 HFPERCLK_{TIMER0} cycles.

19.3.3.1 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 however, if this is required by the application. The active values of the primary and complementary outputs are set by two the TIMERO_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 19.24 (p. 264)

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 TIMERO_DTCTRL register, which inverts the polarity of the complementary outputs relative to the primary outputs.

Example 19.1. TIMER DTI Example 1

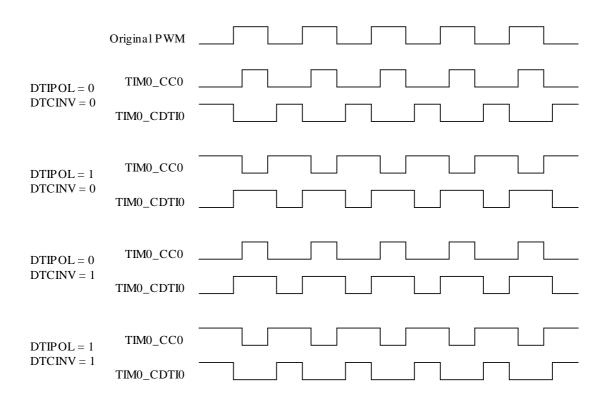
DTIPOL = 0 and DTCINV = 0 results in outputs with opposite phase and active-high states.

Example 19.2. TIMER DTI Example 2

DTIPOL = 1 and DTCINV = 1 results in outputs with equal phase. The primary output will be active-high, while the complementary will be active-low



Figure 19.24. 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, it will go to and stay in its inactive state.

19.3.3.2 PRS Channel as Source

A PRS channel can optionally be used as input to the DTI module instead of the PWM output from the timer. Setting DTPRSEN in TIMERO_DTCTRL will override the source of the first DTI channel, driving TIMO_CCO and TIMO_CDTIO, 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 PRS channel to use is chosen by configuring DTPRSSEL in TIMERO_DTCTRL. Note that the timer must be running even when PRS is used as DTI source.

The DTI prescaler, set by DTPRESC in TIMER0_DTTIME determines with which accuracy the DTI can insert dead-time into a PRS signal. The maximum dead-time error equals 2^{DTPRESC} clock cycles. With zero prescaling, the inserted dead-times are therefore accurate, but they may be inaccurate for larger prescaler settings.

19.3.3.3 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 makes a fast reaction to faults possible, reducing the possibility of damage to the system.

The fault sources which trigger a fault in the DTI module are determined by TIMER0_DTFSEN. Any combination of the available error sources can be selected:

- PRS source 0, determined by DTPRS0FSEL in TIMER0_DTFC
- PRS source 1, determined by DTPRS1FSEL in TIMER0 DTFC
- Debugger
- Core Lockup



One or two PRS channels can be used as an error source. When PRS source 0 is selected as an error source, DTPRS0FSEL determines which PRS channel is used for this source. DTPRS1FSEL determines which PRS channel is selected as PRS source 1. Please note that for Core Lockup, the LOCKUPRDIS in RMU_CTRL must be set. Otherwise this will generate a full reset of the EFM32.

19.3.3.3.1 Action on Fault

When a fault occurs, the bit representing the fault source is set in DTFS, 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 DTFACT in TIMERO_DTFC:

- · 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 of TIMER0_DTFS. TIMER0_DTFS is organized in the same way as DTFSEN, with one bit for each source.

19.3.3.3.2 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 TIMER0_DTFS. If the fault cause, determined by TIMER0_DTFS, is the debugger alone, the outputs can optionally be re-enabled when the debugger exits and the processor resumes normal operation. The corresponding bit in TIMER0_DTFS will in that case be cleared by hardware. The automatic start-up functionality can be enabled by setting DTDAS in TIMER0_DTCTRL. If more bits are still set in DTFS when the automatic start-up functionality has cleared the debugger bit, the DTI module does not exit the fault state. The fault state is only exited when all the bits in TIMER0_DTFS have been cleared.

19.3.3.4 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 results in TIMER0_DTFC, TIMER0_DTCTRL, TIMER0_DTTIME and TIMER0_ROUTE being locked for writing. To unlock the registers, write 0xCE80 to LOCKKEY in TIMER0_DTLOCK. The value of TIMER0_DTLOCK is 1 when the lock is active, and 0 when the registers are unlocked.

19.3.4 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 DBGHALT in TIMERn_CTRL.

19.3.5 Interrupts, DMA and PRS Output

The Timer has 5 output events:

- Counter Underflow
- · Counter Overflow
- Compare match or input capture (one per Compare/Capture channel)

Each of the events has its own interrupt flag. Also, there is one interrupt flag for each Compare/Capture channel which is set on buffer overflow in capture mode. Buffer overflow happens when a new capture pushes an old unread capture out of the TIMERn_CCx_CCV/TIMERn_CCx_CCVB register pair.



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 will also lead to a one HFPERCLK_{TIMERn} cycle high pulse on individual PRS outputs.

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 19.3 (p. 266). If DMACLRACT is set in TIMERn_CTRL, the DMA request is cleared when the triggered DMA channel is active, without having to access any timer registers.

Table 19.3. TIMER Events

Event	Acknowledge
Underflow/Overflow	Read or write to TIMERn_CNT or TIMERn_TOPB
CC 0	Read or write to TIMERn_CC0_CCV or TIMERn_CC0_CCVB
CC 1	Read or write to TIMERn_CC1_CCV or TIMERn_CC1_CCVB
CC 2	Read or write to TIMERn_CC2_CCV or TIMERn_CC2_CCVB

19.3.6 GPIO Input/Output

The TIMn_CCx inputs/outputs and TIM0_CDTIx outputs are accessible as alternate functions through GPIO. Each pin connection can be enabled/disabled separately by setting the corresponding CCxPEN or CDTIxPEN bits in TIMERn_ROUTE. The LOCATION bits in the same register can be used to move all enabled pins to alternate pins.



19.4 Register Map

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

Offset	Name	Туре	Description
0x000	TIMERn_CTRL	RW	Control Register
0x004	TIMERn_CMD	W1	Command Register
0x008	TIMERn_STATUS	R	Status Register
0x00C	TIMERn_IEN	RW	Interrupt Enable Register
0x010	TIMERn_IF	R	Interrupt Flag Register
0x014	TIMERn_IFS	W1	Interrupt Flag Set Register
0x018	TIMERn_IFC	W1	Interrupt Flag Clear Register
0x01C	TIMERn_TOP	RWH	Counter Top Value Register
0x020	TIMERn_TOPB	RW	Counter Top Value Buffer Register
0x024	TIMERn_CNT	RWH	Counter Value Register
0x028	TIMERn_ROUTE	RW	I/O Routing Register
0x030	TIMERn_CC0_CTRL	RW	CC Channel Control Register
0x034	TIMERn_CC0_CCV	RWH	CC Channel Value Register
0x038	TIMERn_CC0_CCVP	R	CC Channel Value Peek Register
0x03C	TIMERn_CC0_CCVB	RWH	CC Channel Buffer Register
0x040	TIMERn_CC1_CTRL	RW	CC Channel Control Register
0x044	TIMERn_CC1_CCV	RWH	CC Channel Value Register
0x048	TIMERn_CC1_CCVP	R	CC Channel Value Peek Register
0x04C	TIMERn_CC1_CCVB	RWH	CC Channel Buffer Register
0x050	TIMERn_CC2_CTRL	RW	CC Channel Control Register
0x054	TIMERn_CC2_CCV	RWH	CC Channel Value Register
0x058	TIMERn_CC2_CCVP	R	CC Channel Value Peek Register
0x05C	TIMERn_CC2_CCVB	RWH	CC Channel Buffer Register
0x070	TIMERn_DTCTRL	RW	DTI Control Register
0x074	TIMERn_DTTIME	RW	DTI Time Control Register
0x078	TIMERn_DTFC	RW	DTI Fault Configuration Register
0x07C	TIMERn_DTOGEN	RW	DTI Output Generation Enable Register
0x080	TIMERn_DTFAULT	R	DTI Fault Register
0x084	TIMERn_DTFAULTC	W1	DTI Fault Clear Register
0x088	TIMERn_DTLOCK	RW	DTI Configuration Lock Register



19.5 Register Description

19.5.1 TIMERn_CTRL - Control Register

Offset		Bit Position																														
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	ю	7	-	0
Reset				,			0x0						·	•	9	OX O		•		•	9	Š	6	OXO	0	0	0	0	0		0	OX OX
Access						-	≷ Y								7	} Ł					7	<u> </u>	ž	} ⊻	RW	RW	W.	W.	W.		i	 ≩ Ƴ
Name						() () ()	PRESC								100	2					<	`	Š.	KIOEA	DMACLRACT	DEBUGRUN	QDM	OSMEN	SYNC		L	MODI

Bit	Name	Reset	Access	Description
31:28	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
27:24	PRESC	0x0	RW	Prescaler Setting
	These bits sele	ect the prescaling factor.		
	Value	Mode	Des	scription
	0	DIV1	The	HFPERCLK is undivided
	1	DIV2	The	HFPERCLK is divided by 2
	2	DIV4	The	HFPERCLK is divided by 4
	3	DIV8	The	HFPERCLK is divided by 8
	4	DIV16	The	HFPERCLK is divided by 16
	5	DIV32	The	HFPERCLK is divided by 32

23:18 Reserved To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

17:16 CLKSEL 0x0 RW Clock Source Select

These bits select the clock source for the timer.

DIV64

DIV128

DIV256

DIV512

DIV1024

10

Value	Mode	Description
0	PRESCHFPERCLK	Prescaled HFPERCLK
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

The HFPERCLK is divided by 64

The HFPERCLK is divided by 128

The HFPERCLK is divided by 256

The HFPERCLK is divided by 512

The HFPERCLK is divided by 1024

15:12 Reserved To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

11:10 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

9:8 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



Bit	Name	Reset	Acces	s Description									
Dit.	Value	Mode	7,0000	Description									
	3	RELOADSTART		Reload and start counter									
				Noticed and start counter									
7	DMACLRACT	0	RW	DMA Request Clear on Active									
		is set, the DMA requests cleared without accessing		nen the corresponding DMA channel is active. This enables the timer DM									
6	DEBUGRUN	0	RW	Debug Mode Run Enable									
	Set this bit to e	enable timer to run in debu	ıg mode.										
	Value	Description											
	0	Timer is frozen	in debug mode										
	1	Timer is running	g in debug mode										
5	QDM	0	RW	Quadrature Decoder Mode Selection									
	This bit sets th	This bit sets the mode for the quadrature decoder.											
	Value	Mode		Description									
	0	X2		X2 mode selected									
	1	X4		X4 mode selected									
4	OSMEN	0	RW	One-shot Mode Enable									
	Enable/disable	e one shot mode.											
3	SYNC	0	RW	Timer Start/Stop/Reload Synchronization									
	When this bit i	s set, the Timer is started/	stopped/reloade	ed by start/stop/reload commands in the other timers									
	Value	Description											
	0	Timer is not sta	rted/stopped/reloa	aded by other timers									
	1	Timer is started	l/stopped/reloaded	d by other timers									
2	Reserved	To ensure	compatibility wit	th future devices, always write bits to 0. More information in Section 2.1 (p. 3)									
1:0	MODE	0x0	RW	Timer Mode									
		he counting mode for the Timer is clocked by the I		then Quadrature Decoder Mode is selected (MODE = 'b11), the CLKSEL is clock output.									
	Value	Mode		Description									
	0	UP		Up-count mode									
	1	DOWN		Down-count mode									
	2	UPDOWN		Up/down-count mode									
			1	· ·									

19.5.2 TIMERn_CMD - Command Register

Offset		Bit Position																														
0x004	31	30	29	28	27	56	25	24	23	22	21	20	19	9	17	16	15	4	13	12	=	10	6	80	7	9	2	4	က	2	-	0
Reset																															0	0
Access																															×	W W
Name																															STOP	START

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	STOP	0	W1	Stop Timer
	Write a 1 to this bit to stop	timer		
0	START	0	W1	Start Timer



Bit	Name	Reset	Access	Description
	Write a 1 to this bit to start	timer		

19.5.3 TIMERn_STATUS - Status Register

Offset															Bi	t Po	ositi	on														
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	8	7	9	2	4	ю	2	-	0
Reset					•	0	0	0			•	•		0	0	0					,	0	0	0					,	0	0	0
Access						ď	œ	œ						œ	~	~						œ	~	2						œ	œ	œ
Name						CCPOL2	CCPOL1	CCPOLO						ICV2	ICV1	ICV0						CCVBV2	CCVBV1	CCVBV0						TOPBV	DIR	RUNNING

B.14				
Bit	Name	Reset	Access	Description
31:27	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
26	CCPOL2	0	R	CC2 Polarity
				e edge that triggered capture in TIMERn_CC2_CCV. In Compare/PWM CC channel 2. These bits are cleared when CCMODE is written to 0b00
	mode, this bit indic		selected input to (

25 CCPOL1 0 R CC1 Polarity

HIGHFALL

In Input Capture mode, this bit indicates the polarity of the edge that triggered capture in TIMERn_CC1_CCV. In Compare/PWM mode, this bit indicates the polarity of the selected input to CC channel 1. These bits are cleared when CCMODE is written to 0b00 (Off).

CC2 polarity high level/falling edge

Value	Mode	Description
0	LOWRISE	CC1 polarity low level/rising edge
1	HIGHFALL	CC1 polarity high level/falling edge

24 CCPOL0 0 R CC0 Polarity

In Input Capture mode, this bit indicates the polarity of the edge that triggered capture in TIMERn_CCO_CCV. In Compare/PWM mode, this bit indicates the polarity of the selected input to CC channel 0. These bits are cleared when CCMODE is written to 0b00 (Off).

Value	Mode	Description
0	LOWRISE	CC0 polarity low level/rising edge
1	HIGHFALL	CC0 polarity high level/falling edge

23:19 Reserved To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

ICV2 0 R CC2 Input Capture Valid

This bit indicates that TIMERn_CC2_CCV contains a valid capture value. These bits are only used in input capture mode and are cleared when CCMODE is written to 0b00 (Off).

Value	Description
0	TIMERn_CC2_CCV does not contain a valid capture value(FIFO empty)
1	TIMERn_CC2_CCV contains a valid capture value(FIFO not empty)

17 ICV1 0 R CC1 Input Capture Valid

This bit indicates that TIMERn_CC1_CCV contains a valid capture value. These bits are only used in input capture mode and are cleared when CCMODE is written to 0b00 (Off).

Value	Description
0	TIMERn_CC1_CCV does not contain a valid capture value(FIFO empty)
1	TIMERn_CC1_CCV contains a valid capture value(FIFO not empty)

16 ICV0 0 R **CC0 Input Capture Valid**

18



Bit	Name	Reset	Access	Description
		es that TIMERn_CC0_CCCCMODE is written to 0b0		id capture value. These bits are only used in input capture mode and are
	Value	Description		
	0	TIMERn_CC0_	CCV does not conta	ain a valid capture value(FIFO empty)
	1	TIMERn_CC0_	CCV contains a vali	id capture value(FIFO not empty)
15:11	Reserved	To ensure o	compatibility with	future devices, always write bits to 0. More information in Section 2.1 (p. 3)
10	CCVBV2	0	R	CC2 CCVB Valid
				s contain data which have not been written to TIMERn_CC2_CCV. These cleared when CCMODE is written to 0b00 (Off).
	Value	Description		
	0	TIMERn_CC2_	CCVB does not con	otain valid data
	1	TIMERn_CC2_	CCVB contains vali	d data which will be written to TIMERn_CC2_CCV on the next update event
9	CCVBV1	0	R	CC1 CCVB Valid
				s contain data which have not been written to TIMERn_CC1_CCV. These cleared when CCMODE is written to 0b00 (Off).
	Value	Description		
	0	TIMERn_CC1_	CCVB does not con	ntain valid data
	1	TIMERn_CC1_	CCVB contains vali	d data which will be written to TIMERn_CC1_CCV on the next update event
8	CCVBV0	0	R	CC0 CCVB Valid
				s contain data which have not been written to TIMERn_CC0_CCV. These cleared when CCMODE is written to 0b00 (Off).
	Value	Description		
	0	TIMERn_CC0_	CCVB does not con	ntain valid data
	1	TIMERn_CC0_	CCVB contains vali	d data which will be written to TIMERn_CC0_CCV on the next update event
7:3	Reserved	To ensure of	compatibility with	future devices, always write bits to 0. More information in Section 2.1 (p. 3)
	TOPBV	0		
2		· ·	R	TOPB Valid
2	This indicates	that TIMERn_TOPB cont		TOPB Valid hat has not been written to TIMERn_TOP. This bit is also cleared when
2		that TIMERn_TOPB cont		
2	TIMERn_TOP	that TIMERn_TOPB cont s written.		nat has not been written to TIMERn_TOP. This bit is also cleared when
2	TIMERn_TOP	that TIMERn_TOPB conts written. Description TIMERn_TOPE	ains valid data the	nat has not been written to TIMERn_TOP. This bit is also cleared when
1	TIMERn_TOP	that TIMERn_TOPB conts written. Description TIMERn_TOPE	ains valid data the	hat has not been written to TIMERn_TOP. This bit is also cleared when alid data
	Value 0 1	that TIMERn_TOPB conts written. Description TIMERn_TOPE TIMERn_TOPE	ains valid data the	nat has not been written to TIMERn_TOP. This bit is also cleared when alid data which will be written to TIMERn_TOP on the next update event
	Value 0 1 DIR	that TIMERn_TOPB conts written. Description TIMERn_TOPE TIMERn_TOPE	ains valid data the does not contain valid data	nat has not been written to TIMERn_TOP. This bit is also cleared when alid data which will be written to TIMERn_TOP on the next update event
	Value 0 1 DIR Indicates count	that TIMERn_TOPB conts written. Description TIMERn_TOPE 0 direction.	ains valid data the does not contain valid data	hat has not been written to TIMERn_TOP. This bit is also cleared when alid data which will be written to TIMERn_TOP on the next update event Direction
	Value O Indicates count Value	that TIMERn_TOPB conts written. Description TIMERn_TOPE 0 direction. Mode	ains valid data the does not contain valid data R Do Co	hat has not been written to TIMERn_TOP. This bit is also cleared when alid data which will be written to TIMERn_TOP on the next update event Direction
	Value O Indicates count Value	that TIMERn_TOPB conts written. Description TIMERn_TOPE TIMERn_TOPE 0 direction. Mode UP	ains valid data the does not contain valid data R Do Co	nat has not been written to TIMERn_TOP. This bit is also cleared when alid data which will be written to TIMERn_TOP on the next update event Direction escription ounting up

19.5.4 TIMERn_IEN - Interrupt Enable Register

Offset															Bi	t Pc	siti	on														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	=	10	6	8	7	9	2	4	ю	2	1	0
Reset																						0	0	0		0	0	0			0	0
Access																						RW	RW	RW W		RW	RW	RW W			RW	RW
Name																						ICBOF2	ICBOF1	ICBOF0		CC2	CC1	000			UF	OF



Bit	Name	Reset	Access	Description
31:11	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
10	ICBOF2	0	RW	CC Channel 2 Input Capture Buffer Overflow Interrupt Enable
	Enable/disable Co	mpare/Capture ch 2 inp	out capture buffer	overflow interrupt.
9	ICBOF1	0	RW	CC Channel 1 Input Capture Buffer Overflow Interrupt Enable
	Enable/disable Co	mpare/Capture ch 1 inp	out capture buffer	overflow interrupt.
8	ICBOF0	0	RW	CC Channel 0 Input Capture Buffer Overflow Interrupt Enable
	Enable/disable Co	mpare/Capture ch 0 inp	out capture buffer	overflow interrupt.
7	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
6	CC2	0	RW	CC Channel 2 Interrupt Enable
	Enable/disable Co	mpare/Capture ch 2 into	errupt.	
5	CC1	0	RW	CC Channel 1 Interrupt Enable
	Enable/disable Co	mpare/Capture ch 1 inte	errupt.	
4	CC0	0	RW	CC Channel 0 Interrupt Enable
	Enable/disable Co	mpare/Capture ch 0 into	errupt.	
3:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	UF	0	RW	Underflow Interrupt Enable
	Enable/disable und	derflow interrupt.		
0	OF	0	RW	Overflow Interrupt Enable
	Enable/disable ove	erflow interrupt.		

19.5.5 TIMERn_IF - Interrupt Flag Register

Offset															Bi	t Po	siti	on														
0x010	31	30	53	28	27	56	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	က	7	-	0
Reset									•								•			•		0	0	0		0	0	0			0	0
Access																						~	~	~		~	~	~			œ	~
Name																						ICBOF2	ICBOF1	ICBOF0		CC2	CC1	000			JU	OF

Bit	Name	Reset	Access	Description
31:11	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
10	ICBOF2	0	R	CC Channel 2 Input Capture Buffer Overflow Interrupt Flag
	This bit indicates the pair.	hat a new capture value	e has pushed an ur	nread value out of the TIMERn_CC2_CCV/TIMERn_CC2_CCVB register
9	ICBOF1	0	R	CC Channel 1 Input Capture Buffer Overflow Interrupt Flag
	This bit indicates the pair.	hat a new capture value	e has pushed an ur	nread value out of the TIMERn_CC1_CCV/TIMERn_CC1_CCVB register
8	ICBOF0	0	R	CC Channel 0 Input Capture Buffer Overflow Interrupt Flag
	This bit indicates the pair.	hat a new capture value	e has pushed an ur	nread value out of the TIMERn_CC0_CCV/TIMERn_CC0_CCVB register
7	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
6	CC2	0	R	CC Channel 2 Interrupt Flag
	This bit indicates t	hat there has been an	interrupt event on	Compare/Capture channel 2.
5	CC1	0	R	CC Channel 1 Interrupt Flag



Bit	Name	Reset	Access	Description
	This bit indicates t	hat there has been an i	nterrupt event on	Compare/Capture channel 1.
4	CC0	0	R	CC Channel 0 Interrupt Flag
	This bit indicates t	hat there has been an i	nterrupt event on	Compare/Capture channel 0.
3:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	UF	0	R	Underflow Interrupt Flag
	This bit indicates t	hat there has been an ι	ınderflow.	
0	OF	0	R	Overflow Interrupt Flag
	This bit indicates t	hat there has been an o	overflow.	

19.5.6 TIMERn_IFS - Interrupt Flag Set Register

Offset															Bi	t Po	siti	on														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	=	10	6	æ	7	9	2	4	ю	7	-	0
Reset																						0	0	0		0	0	0		•	0	0
Access																						W 1	W1	W1		W1	W1	W1			W1	W 1
Name																						ICBOF2	ICB0F1	ICBOF0		CC2	CC1	000			JU	OF

Bit	Name	Reset	Access	Description
31:11	Reserved	To ensure c	ompatibility with fu	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
10	ICBOF2	0	W1	CC Channel 2 Input Capture Buffer Overflow Interrupt Flag Set
	Writing a 1 to this b	oit will set Compare/Ca	pture channel 2 ir	nput capture buffer overflow interrupt flag.
9	ICBOF1	0	W1	CC Channel 1 Input Capture Buffer Overflow Interrupt Flag Set
	Writing a 1 to this b	oit will set Compare/Ca	pture channel 1 ir	put capture buffer overflow interrupt flag.
8	ICBOF0	0	W1	CC Channel 0 Input Capture Buffer Overflow Interrupt Flag Set
	Writing a 1 to this b	oit will set Compare/Ca	pture channel 0 ir	put capture buffer overflow interrupt flag.
7	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3,
6	CC2	0	W1	CC Channel 2 Interrupt Flag Set
	Writing a 1 to this b	oit will set Compare/Ca	pture channel 2 ir	nterrupt flag.
5	CC1	0	W1	CC Channel 1 Interrupt Flag Set
	Writing a 1 to this b	oit will set Compare/Ca	pture channel 1 ir	nterrupt flag.
4	CC0	0	W1	CC Channel 0 Interrupt Flag Set
	Writing a 1 to this b	oit will set Compare/Ca	pture channel 0 ir	nterrupt flag.
3:2	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	UF	0	W1	Underflow Interrupt Flag Set
	Writing a 1 to this b	oit will set the underflow	v interrupt flag.	
0	OF	0	W1	Overflow Interrupt Flag Set
	Writing a 1 to this b	oit will set the overflow	interrupt flag.	



19.5.7 TIMERn_IFC - Interrupt Flag Clear Register

Offset															Bi	t Po	siti	on														
0x018	31	30	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	တ	∞	7	9	2	4	ო	2	-	0
Reset				,																		0	0	0		0	0	0			0	0
Access																						×	W 1	W 1		N N	W W	W			W V	W W
Name																						ICBOF2	ICBOF1	ICBOF0		CC2	CC1	000			UF	OF
Bit	Na	me						Re	set			Α	CC	ess		De	scri	iptio	on													
31:11	Re	serv	ed					То	ensi	ure c	comp	atib	ility	with	futu	re d	evice	es, a	lwa	уѕ и	rite l	bits t	to 0.	Mor	e inf	orm	atio	n in 🤅	Sect	ion 2	2.1 (p	o. 3)

DIL	Ivaille	Reset	Access	Description
31:11	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
10	ICBOF2	0	W1	CC Channel 2 Input Capture Buffer Overflow Interrupt Flag Clear
	Writing a 1 to this b	oit will clear Compare/C	Capture channel 2	input capture buffer overflow interrupt flag.
9	ICBOF1	0	W1	CC Channel 1 Input Capture Buffer Overflow Interrupt Flag Clear
	Writing a 1 to this b	oit will clear Compare/C	Capture channel 1	input capture buffer overflow interrupt flag.
8	ICBOF0	0	W1	CC Channel 0 Input Capture Buffer Overflow Interrupt Flag Clear
	Writing a 1 to this b	oit will clear Compare/C	capture channel 0	input capture buffer overflow interrupt flag.
7	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
6	CC2	0	W1	CC Channel 2 Interrupt Flag Clear
	Writing a 1 to this b	oit will clear Compare/C	apture interrupt fl	lag 2.
5	CC1	0	W1	CC Channel 1 Interrupt Flag Clear
	Writing a 1 to this b	oit will clear Compare/C	apture interrupt fl	ag 1.
4	CC0	0	W1	CC Channel 0 Interrupt Flag Clear
	Writing a 1 to this b	oit will clear Compare/C	apture interrupt fl	lag 0.
3:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	UF	0	W1	Underflow Interrupt Flag Clear
	Writing a 1 to this b	oit will clear the underflo	ow interrupt flag.	
0	OF	0	W1	Overflow Interrupt Flag Clear
	Writing a 1 to this b	oit will clear th overflow	interrupt flag.	

19.5.8 TIMERn_TOP - Counter Top Value Register

Offset				,											Bi	t Pc	siti	on														
0x01C	31	93	53	78	27	26	22	24	23	22	21	70	19	18	17	16	15	4	13	12	7	10	6	80	7	9	2	4	က	2	-	0
Reset																									- - - - -							
Access																								I MA								
Name																								a C L	5							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	TOP	0xFFFF	RWH	Counter Top Value



Bit	Name	Reset	Access	Description
	These bits hold the TOP va	alue for the counter		

19.5.9 TIMERn_TOPB - Counter Top Value Buffer Register

Offset															Bi	t Po	siti	on														
0x020	31	98	53	78	27	26	22	24	23	22	21	20	19	18	17	16	15	4	13	12	1	9	6	80	7	9	2	4	က	2	-	0
Reset																							•		0000x0	•						
Access																								i	¥ M							
Name																								1 1 1	TOPB							
Bit	Na	ıme						Re	set			Δ	\cc	ess		De	scr	iptic	on													
31:16	Re	serv	ed					То	ensı	ıre c	omp	atib	ility	with	futu	re de	evice	es, a	lwa _.	уѕ и	rite	bits	to 0.	Mor	e in	forn	natio	n in	Sect	ion 2	2.1 (p	. 3)
15:0	ТО	РВ						0x0	0000			R	W			Co	unte	r To	р۷	/alu	е Ви	ıffeı	r									
	The	ese b	oits h	old t	the 1	ГОР	bu'	ffer v	/alue	٠.																						

19.5.10 TIMERn_CNT - Counter Value Register

Offset															Bi	t Po	siti	on														
0x024	33	30	29	28	27	56	22	24	23	22	21	70	19	18	17	16	15	4	13	12	=	19	6	80	7	9	2	4	က	2	-	0
Reset																									000000							
Access																								į	I M Y							
Name																								Ė	200							
Bit	Na	ame						Re	set			A	\cc	ess		De	scri	iptic	on													
31:16	Re	serv	ed					То	ensı	ıre c	omp	atib	ility	with	futu	re de	evice	es, a	lwaj	уѕ и	/rite	bits t	0 0.	Mor	e inf	orm	natio	n in s	Sect	ion 2	.1 (p	o. 3)
15:0	CN	ΙΤ						0x0	000			R	WH	ı		Cou	ınte	r Va	llue													
	Th	ese b	oits h	old 1	the o	cour	nter	valu	e.																							

19.5.11 TIMERn_ROUTE - I/O Routing Register

Offset					·				·						Bi	t Po	siti	on														
0x028	31	30	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	41	13	12	7	10	6	8	7	9	2	4	ю	2	-	0
Reset				•						•				,	2	S S					•	0	0	0					,	0	0	0
Access															\ \ \	2						RW	RW	RW						RW	RW	RW
Name															NOITY	5						CDTI2PEN	CDTI1PEN	CDTIOPEN						CC2PEN	CC1PEN	CCOPEN



Bit Name Reset 31:18 Reserved To ensure com	Access	Description
31:18 Reserved To ensure com		Description
100/100/00/11	patibility with	future devices, always write bits to 0. More information in Section 2.1 (p. 3)
17:16 LOCATION 0x0	RW	I/O Location
Decides the location of the CC pins.		
Value Mode	De	escription
0 LOC0	Lo	ocation 0
1 LOC1	Lo	ocation 1
2 LOC2	Lo	ocation 2
3 LOC3	Lo	ocation 3
15:11 Reserved To ensure com	patibility with	future devices, always write bits to 0. More information in Section 2.1 (p. 3)
10 CDTI2PEN 0	RW	CC Channel 2 Complementary Dead-Time Insertion Pin Enable
Enable/disable CC channel 2 complementar	y dead-time ir	nsertion output connection to pin.
9 CDTI1PEN 0	RW	CC Channel 1 Complementary Dead-Time Insertion Pin Enable
Enable/disable CC channel 1 complementar	y dead-time ir	nsertion output connection to pin.
8 CDTIOPEN 0	RW	CC Channel 0 Complementary Dead-Time Insertion Pin Enable
Enable/disable CC channel 0 complementar	y dead-time ir	nsertion output connection to pin.
7:3 Reserved To ensure com	patibility with	future devices, always write bits to 0. More information in Section 2.1 (p. 3)
2 CC2PEN 0	RW	CC Channel 2 Pin Enable
Enable/disable CC channel 2 output/input co	onnection to p	in.
1 CC1PEN 0	RW	CC Channel 1 Pin Enable
Enable/disable CC channel 1 output/input co	onnection to p	in.
0 CCOPEN 0	RW	CC Channel 0 Pin Enable
Enable/disable CC Channel 0 output/input c	onnection to p	oin.

19.5.12 TIMERn_CCx_CTRL - CC Channel Control Register

Offset															Bi	t Po	siti	on														
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	=	10	6	8	7	9	2	4	3	7	-	0
Reset					000		Š	Š			0	0			0x0					0 0 0	2	Š	Ġ	OX OX				0		0	2	OXO
Access					% W		2	<u>}</u>			W.	W.			W.				i	S N	Š	<u> </u>	ž	≩ Y				W.		W.	7	 }
Name					ICEVCTRL		<u> </u>	2			FILT	INSEL			PRSSEL				(!	CUFOA	S C			CINCA				COIST		OUTINV	HCOM	

Bit	Name	Reset	Acces	s Description
31:28	Reserved	To ensure con	mpatibility wit	h future devices, always write bits to 0. More information in Section 2.1 (p. 3)
27:26	ICEVCTRL	0x0	RW	Input Capture Event Control
	These bits contr	ol when a Compare/Captur	e PRS outpu	at pulse, interrupt flag and DMA request is set.
	Value	Mode		Description
	0	EVERYEDGE		PRS output pulse, interrupt flag and DMA request set on every capture
	1	EVERYSECONDEDGE		PRS output pulse, interrupt flag and DMA request set on every second capture
	2	RISING		PRS output pulse, interrupt flag and DMA request set on rising edge only (if ICEDGE = BOTH)
	3	FALLING		PRS output pulse, interrupt flag and DMA request 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.

www.silabs.com



Bit	Name	Reset	Acces	ss Description
	Value	Mode		Description
	0	RISING		Rising edges detected
	1	FALLING		Falling edges detected
	2	вотн		Both edges detected
	3	NONE		No edge detection, signal is left as it is
23:22	Reserved	To ensure	compatibility w	ith future devices, always write bits to 0. More information in Section 2.1 (p. 3)
21	FILT	0	RW	Digital Filter
	Enable digital	filter.		
	Value	Mode		Description
	0	DISABLE		Digital filter disabled
	1	ENABLE		Digital filter enabled
20	INSEL Select Compa	0 are/Capture channel input.	RW	Input Selection
	Value	Mode		Description
	0	PIN		TIMERnCCx pin is selected
	1	PRS		PRS input (selected by PRSSEL) is selected
19	Reserved	To ensure	compatibility w	ith future devices, always write bits to 0. More information in Section 2.1 (p. 3)
18:16	PRSSEL	0x0	RW	Compare/Capture Channel PRS Input Channel Selection
	Select PRS in	put channel for Compare/0	Capture chann	el.
	Value	Mode		Description
	0	PRSCH0		PRS Channel 0 selected as input
	1	PRSCH1		PRS Channel 1 selected as input
	2	PRSCH2		PRS Channel 2 selected as input
	3	PRSCH3		PRS Channel 3 selected as input
	5	PRSCH4 PRSCH5		PRS Channel 4 selected as input PRS Channel 5 selected as input
	6	PRSCH6		PRS Channel 6 selected as input
	7	PRSCH7		PRS Channel 7 selected as input
15:14	Reserved	To ensure o	compatibility w	ith future devices, always write bits to 0. More information in Section 2.1 (p. 3)
13:12	CUFOA	0x0	RW	Counter Underflow Output Action
	Select output	action on counter underflo	w.	
	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
11:10	COFOA Select output	0x0 action on counter overflow	RW	Counter Overflow Output Action
	Value	Mode		Description
	0	NONE		Description 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



Bit	Name	Reset	Access	Description						
	Value	Mode	De	escription						
	3	SET	Se	et output on compare match						
7:5	Reserved	To ensure co	mpatibility with	future devices, always write bits to 0. More information in Section 2.1 (p. 3)						
4	COIST	0	RW	Compare Output Initial State						
	is disabled. Ŵ cleared when	hen counting resumes, this	value will repre PWM mode, the	When this bit is set in compare mode, the output is set high when the counter sent the initial value for the output. If the bit is cleared, the output will be output will always be low when disabled, regardless of this bit. However, is enabled.						
3	Reserved	To ensure co	To ensure compatibility with future devices, always write bits to 0. More information in Section 2							
2	OUTINV	0	RW	Output Invert						
	Setting this bi	t inverts the output from the	CC channel (Ou	utput compare,PWM).						
1:0	MODE	0x0	RW	CC Channel Mode						
	These bits se	lect the mode for Compare/C	the mode for Compare/Capture channel.							
	Value	Mode	De	escription						
	0	OFF	Co	ompare/Capture channel turned off						
	1	INPUTCAPTURE	Inj	put capture						
	2	OUTPUTCOMPARE	Ot	utput compare						

19.5.13 TIMERn_CCx_CCV - CC Channel Value Register

Offset															Bi	t Pc	siti	on														
0x034	33	30	53	28	27	26	22	24	23	22	21	20	19	18	17	16	15	4	5 5	7	1	19	6	8	7	9	2	4	က	2	-	0
Reset																								0000	OXOOOO							
Access																									E 2 2							
Name																								ć	<u>}</u>							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure comp	patibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	CCV	0x0000	RWH	CC Channel Value
				re value. When reading this register in input capture mode, then contents Rn_CCx_CCV in the next cycle. In compare mode, this fields holds the



19.5.14 TIMERn_CCx_CCVP - CC Channel Value Peek Register

Offset															Bit	t Po	siti	on														
0x038	31	99	53	28	27	26	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	ω	7	9	2	4	က	2	-	0
Reset																								000	000000							
Access																								C	צ							
Name																								G,	2)							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	mpatibility with fu	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	CCVP	0x0000	R	CC Channel Value Peek
	This field is used t	o read the CC value with	out pulling data	through the FIFO in capture mode.

19.5.15 TIMERn_CCx_CCVB - CC Channel Buffer Register

Offset															Bi	t Po	siti	on														
0x03C	33	30	53	78	27	56	22	24	23	72	21	70	19	18	17	16	15	4	13	12	7	9	6	8	7	9	2	4	က	2	-	0
Reset																									000000							
Access																								2	I M Y							
Name																								Q	a ^>							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	CCVB	0x0000	RWH	CC Channel Value Buffer
	' '	•	•	ue if the TIMERn_CCx_CCV register already contains an earlier unread d holds the CC buffer value which will be written to TIMERn_CCx_CCV

on an update event if TIMERn_CCx_CCVB contains valid data.

19.5.16 TIMERn_DTCTRL - DTI Control Register

Offset															Bi	t Po	siti	on														
0x070	33	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	8	7	9	2	4	ю	2	-	0
Reset				•				0					•		,			•					•				0x0		0	0	0	0
Access								RW																			RW		W.	W.	W.	RW
Name								DTPRSEN																			DTPRSSEL		DTCINV	DTIPOL	DTDAS	DTEN



Bit	Name	Reset	Access	Description
31:25	Reserved	To ensure c	ompatibility with	n future devices, always write bits to 0. More information in Section 2.1 (p. 3)
24	DTPRSEN	0	RW	DTI PRS Source Enable
	Enable/disable	PRS as DTI input.		
23:7	Reserved	To ensure c	ompatibility with	n future devices, always write bits to 0. More information in Section 2.1 (p. 3)
6:4	DTPRSSEL	0x0	RW	DTI PRS Source Channel Select
	Select which P	RS channel to listen to.		
	Value	Mode		Description
	0	PRSCH0	F	PRS Channel 0 selected as input
	1	PRSCH1	F	PRS Channel 1 selected as input
	2	PRSCH2	F	PRS Channel 2 selected as input
	3	PRSCH3	F	PRS Channel 3 selected as input
	4	PRSCH4	F	PRS Channel 4 selected as input
	5	PRSCH5	F	PRS Channel 5 selected as input
	6	PRSCH6	F	PRS Channel 6 selected as input
	7	PRSCH7	F	PRS Channel 7 selected as input
3	DTCINV	0	RW	DTI Complementary Output Invert.
	Set to invert co	emplementary outputs.		
2	DTIPOL	0	RW	DTI Inactive Polarity
	Set inactive po	larity for outputs.		
1	DTDAS	0	RW	DTI Automatic Start-up Functionality
	Configure DTI	restart on debugger exit.		
	Value	Mode		Description
	0	NORESTART	١	No DTI restart on debugger exit
	1	RESTART	0	OTI restart on debugger exit
0	DTEN	0	RW	DTI Enable
	Enable/disable	DTI.		

19.5.17 TIMERn_DTTIME - DTI Time Control Register

Offset															Bi	t Pc	siti	on														
0x074	33	30	29	28	27	56	22	24	23	22	21	20	19	9	17	16	15	4	13	12	=	10	6	∞	7	9	2	4	က	7	-	0
Reset													Ö	00X0								00X0								-	3	
Access													i	≥ Y								R ⊗									<u> </u>	
Name													- - - - -	DIFALLI								DTRISET								Condata	22	

Bit	Name	Reset	Access	Description
31:22	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
21:16	DTFALLT	0x00	RW	DTI Fall-time
	Set time span for the	he falling edge.		
	Value		Descripti	on
	DTFALLT		Fall time	of DTFALLT+1 prescaled HFPERCLK cycles
15:14	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
	DTRISET			



Bit	Name	Reset	Access	s Description
	Set time span f	or the rising edge.		
	Value		Desc	ription
	DTRISET		Rise	time of DTRISET+1 prescaled HFPERCLK cycles
7:4	Reserved	To ensure co	mpatibility witl	h future devices, always write bits to 0. More information in Section 2.1 (p. 3)
3:0	DTPRESC	0x0	RW	DTI Prescaler Setting
	Select prescale	er for DTI.		
	Value	Mode	ı	Description
	0	DIV1	-	The HFPERCLK is undivided
	1	DIV2	-	The HFPERCLK is divided by 2
	2	DIV4	-	The HFPERCLK is divided by 4
	3	DIV8	-	The HFPERCLK is divided by 8
	4	DIV16	-	The HFPERCLK is divided by 16
	5	DIV32	-	The HFPERCLK is divided by 32
	6	DIV64	-	The HFPERCLK is divided by 64
	7	DIV128	-	The HFPERCLK is divided by 128
	8	DIV256	-	The HFPERCLK is divided by 256
	9	DIV512	-	The HFPERCLK is divided by 512
	10	DIV1024	-	The HFPERCLK is divided by 1024

19.5.18 TIMERn_DTFC - DTI Fault Configuration Register

Offset															Bi	t Po	siti	on														
0x078	33	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	∞	7	9	2	4	က	2	-	0
Reset					0	0	0	0							2	Š							0x0								000	
Access					RW	RW	RW	RW W		-					2	<u> </u>							RW								S≪	
Name					DTLOCKUPFEN	DTDBGFEN	DTPRS1FEN	DTPRS0FEN							V U H	L							DTPRS1FSEL								DTPRS0FSEL	

			· · · · · · · · · · · · · · · · · · ·	
Bit	Name	Reset	Access	Description
31:28	Reserved	To ensure c	ompatibility with	future devices, always write bits to 0. More information in Section 2.1 (p. 3)
27	DTLOCKUPFEN	0	RW	DTI Lockup Fault Enable
	Set this bit to 1 t	o enable core lockup as a	a fault source	
26	DTDBGFEN	0	RW	DTI Debugger Fault Enable
	Set this bit to 1 t	o enable debugger as a f	ault source	
25	DTPRS1FEN	0	RW	DTI PRS 1 Fault Enable
	Set this bit to 1 t	o enable PRS source 1(F	PRS channel det	termined by DTPRS1FSEL) as a fault source
24	DTPRS0FEN	0	RW	DTI PRS 0 Fault Enable
	Set this bit to 1 t	o enable PRS source 0(F	PRS channel det	termined by DTPRS0FSEL) as a fault source
23:18	Reserved	To ensure co	ompatibility with	future devices, always write bits to 0. More information in Section 2.1 (p. 3)
17:16	DTFA	0x0	RW	DTI Fault Action
	Select fault action	n.		
	Value	Mode	D	escription
	0	NONE	N	lo action on fault
	1	INACTIVE	S	et outputs inactive
	2	CLEAR	С	lear outputs



Bit	Name	Reset	Acces	ss Description
	Value	Mode		Description
	3	TRISTATE		Tristate outputs
15:11	Reserved	To ensure co	ompatibility w	ith future devices, always write bits to 0. More information in Section 2.1 (p. 3)
10:8	DTPRS1FSEL	0x0	RW	DTI PRS Fault Source 1 Select
	Select PRS cha	nnel for fault source 1.		
	Value	Mode		Description
	0	PRSCH0		PRS Channel 0 selected as fault source 1
	1	PRSCH1		PRS Channel 1 selected as fault source 1
	2	PRSCH2		PRS Channel 2 selected as fault source 1
	3	PRSCH3		PRS Channel 3 selected as fault source 1
	4	PRSCH4		PRS Channel 4 selected as fault source 1
	5	PRSCH5		PRS Channel 5 selected as fault source 1
	6	PRSCH6		PRS Channel 6 selected as fault source 1
	7	PRSCH7		PRS Channel 7 selected as fault source 1
7:3	Reserved	To ensure co	ompatibility w	ith future devices, always write bits to 0. More information in Section 2.1 (p. 3)
2:0	DTPRS0FSEL	0x0	RW	DTI PRS Fault Source 0 Select
	Select PRS cha	innel for fault source 0.		
	Value	Mode		Description
	0	PRSCH0		PRS Channel 0 selected as fault source 0
	1	PRSCH1		PRS Channel 1 selected as fault source 0
	2	PRSCH2		PRS Channel 2 selected as fault source 0
	3	PRSCH3		PRS Channel 3 selected as fault source 0
	4	PRSCH4		PRS Channel 4 selected as fault source 0
	5	PRSCH5		PRS Channel 5 selected as fault source 0
	6	PRSCH6		PRS Channel 6 selected as fault source 0
	7	PRSCH7		PRS Channel 7 selected as fault source 0

19.5.19 TIMERn_DTOGEN - DTI Output Generation Enable Register

Offset															Bi	t Po	siti	on														
0x07C	31	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	ю	7	-	0
Reset			•	,						•		•						•				•					0	0	0	0	0	0
Access		-																									RW	W.	W.	RW W	RW	RW
Name																											DTOGCDTI2EN	DTOGCDTI1EN	DTOGCDTI0EN	DTOGCC2EN	DTOGCC1EN	DTOGCC0EN

Bit	Name	Reset	Access	Description
31:6	Reserved	To ensure co	mpatibility with fut	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
5	DTOGCDTI2EN	0	RW	DTI CDTI2 Output Generation Enable
	This bit enables/disables	output generatio	n for the CDTI2 o	utput from the DTI.
4	DTOGCDTI1EN	0	RW	DTI CDTI1 Output Generation Enable
	This bit enables/disables	output generatio	n for the CDTI1 or	utput from the DTI.
3	DTOGCDTI0EN	0	RW	DTI CDTI0 Output Generation Enable
	This bit enables/disables	output generatio	n for the CDTI0 o	utput from the DTI.
2	DTOGCC2EN	0	RW	DTI CC2 Output Generation Enable
	This bit enables/disables	output generatio	n for the CC2 outp	out from the DTI.



Bit	Name	Reset	Access	Description
1	DTOGCC1EN	0	RW	DTI CC1 Output Generation Enable
	This bit enables/disabl	es output generation	n for the CC1 out	out from the DTI.
0	DTOGCC0EN	0	RW	DTI CC0 Output Generation Enable
	This bit enables/disabl	es output generation	n for the CC0 out	out from the DTI.

19.5.20 TIMERn_DTFAULT - DTI Fault Register

Offset															Bi	t Po	siti	on														
0x080	31	30	59	78	27	26	52	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	80	7	9	2	4	က	7	-	0
Reset									•			•					•				•						•		0	0	0	0
Access																													œ	œ	~	~
Name																													DTLOCKUPF	DTDBGF	DTPRS1F	DTPRS0F

Bit	Name	Reset	Access	Description
31:4	Reserved	To ensure comp	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
3	DTLOCKUPF	0	R	DTI Lockup Fault
	This bit is set to 1 if a cor used to clear fault bits.	e lockup fault has	occurred and I	DTLOCKUPFEN is set to 1. The TIMER0_DTFAULTC register can be
2	DTDBGF	0	R	DTI Debugger Fault
	This bit is set to 1 if a deb clear fault bits.	ugger fault has occ	curred and DTI	DBGFEN is set to 1. The TIMER0_DTFAULTC register can be used to
1	DTPRS1F	0	R	DTI PRS 1 Fault
	This bit is set to 1 if a PR clear fault bits.	S 1 fault has occur	rred and DTPF	RS1FEN is set to 1. The TIMER0_DTFAULTC register can be used to
0	DTPRS0F	0	R	DTI PRS 0 Fault
	This bit is set to 1 if a PR clear fault bits.	S 0 fault has occur	rred and DTPF	RS0FEN is set to 1. The TIMER0_DTFAULTC register can be used to

19.5.21 TIMERn_DTFAULTC - DTI Fault Clear Register

Offset															Bi	t Pc	siti	on														
0x084	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	-	10	6	œ	7	9	2	4	3	7	-	0
Reset			•	,							•	•	•		,						•								0	0	0	0
Access																													W1	W1	W1	W
Name																													TLOCKUPFC	DTDBGFC	DTPRS1FC	DTPRS0FC

Bit	Name	Reset	Access	Description						
31:4	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)						
3	TLOCKUPFC	0	W1	DTI Lockup Fault Clear						
	Write 1 to this bit to	clear core lockup faul	core lockup fault.							
2	DTDBGFC	0	W1	DTI Debugger Fault Clear						
	Write 1 to this bit to clear debugger fault.									



Bit	Name	me Reset /		Description						
1	DTPRS1FC 0		W1	DTI PRS1 Fault Clear						
	Write 1 to this bit to cl	lear PRS 1 fault.								
0	DTPRS0FC	0	W1	DTI PRS0 Fault Clear						
	Write 1 to this bit to clear PRS 0 fault.									

19.5.22 TIMERn_DTLOCK - DTI Configuration Lock Register

Offset	Bit Posit													sition																		
0x088	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	8	7	9	2	4	3	2	1	0
Reset																								0000	000000							
Access																								<u> </u>	<u>}</u>							
Name																								\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	LOCKNET							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure com	patibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	LOCKKEY	0x0000	RW	DTI Lock Kev

Write any other value than the unlock code to lock TIMERO_ROUTE, TIMERO_DTCTRL, TIMERO_DTTIME and TIMERO_DTFC from editing. Write the unlock code to unlock. When reading the register, bit 0 is set when the lock is enabled.

Mode	Value	Description
Read Operation		
UNLOCKED	0	TIMER DTI registers are unlocked
LOCKED	1	TIMER DTI registers are locked
Write Operation		
LOCK	0	Lock TIMER DTI registers
UNLOCK	0xCE80	Unlock TIMER DTI registers



20 RTC - Real Time Counter





Quick Facts

What?

The Real Time Counter (RTC) ensures timekeeping in low energy modes. Combined with two low power oscillators (XTAL or RC), the RTC can run in EM2 with total current consumption less than 0.9 μ A.

Why?

Timekeeping over long time periods is required in many applications, while using as little power as possible.

How?

Selectable 32.768 Hz oscillators that can be used as clock source and two different compare registers that can trigger a wake-up. 24-bit resolution and selectable prescaling allow the system to stay in EM2 for a long time and still maintain reliable timekeeping.

20.1 Introduction

The Real Time Counter (RTC) contains a 24-bit counter and is clocked either by a 32.768 Hz crystal oscillator, a 32.768 Hz RC oscillator. In addition to energy modes EM0 and EM1, the RTC is also available in EM2. This makes it ideal for keeping track of time since the RTC is enabled in EM2 where most of the device is powered down.

Two compare channels are available in the RTC. These can be used to trigger interrupts and to wake the device up from a low energy mode. They can also be used with the LETIMER to generate various output waveforms.

20.2 Features

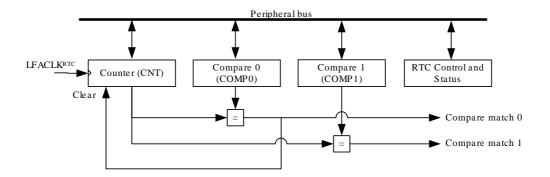
- 24-bit Real Time Counter.
- Prescaler
 - $32.768 \text{ kHz/2}^{\text{N}}$, N = 0 15.
 - Overflow @ 0.14 hours for prescaler setting = 0.
 - Overflow @ 4660 hours (194 days) for prescaler setting = 15 (1 s tick).
- Two compare registers
 - A compare match can potentially wake-up the device from low energy modes EM1 and EM2.
 - Second compare register can be top value for RTC.
 - Both compare channels can trigger LETIMER.
 - Compare match events are available to other peripherals through the Peripheral Reflex System (PRS).

20.3 Functional Description

The RTC is a 24-bit counter with two compare channels. The RTC is closely coupled with the LETIMER, and can be configured to trigger it on a compare match on one or both compare channels. An overview of the RTC module is shown in Figure 20.1 (p. 286) .



Figure 20.1. RTC Overview



20.3.1 Counter

The RTC is enabled by setting the EN bit in the RTC_CTRL register. It counts up as long as it is enabled, and will on an overflow simply wrap around and continue counting. The RTC is cleared when it is disabled. The timer value is both readable and writable and the RTC always starts counting from 0 when enabled. The value of the counter can be read or modified using the RTC_CNT register.

20.3.1.1 Clock Source

The RTC clock source and its prescaler value are defined in the Register Description section of the Clock Management Unit (CMU). The clock used by the RTC has a frequency given by Equation 20.1 (p. 286).

RTC Frequency Equation
$$f_{RTC} = f_{LFACLK}/2^{RTC_PRESC} \tag{20.1}$$

where f_{LFACLK} is the LFACLK frequency (32.768 kHz) and RTC_PRESC is a 4 bit value. Table 20.1 (p. 287) shows the time of overflow and resolution of the RTC at the available prescaler values.

To use this module, the LE interface clock must be enabled in CMU_HFCORECLKEN0 in addition to the module clock



Table 20.1. RTC Resolution Vs Overflow

RTC_PRESC	Resolution	Overflow
0	30,5 µs	512 s
1	61,0 µs	1024 s
2	122 μs	2048 s
3	244 μs	1,14 hours
4	488 μs	2,28 hours
5	977 μs	4,55 hours
6	1,95 ms	9,10 hours
7	3,91 ms	18,2 hours
8	7,81 ms	1,52 days
9	15,6 ms	3,03 days
10	31,25 ms	6,07 days
11	62,5 ms	12,1 days
12	0,125 s	24,3 days
13	0,25 s	48,5 days
14	0,5 s	97,1 days
15	1 s	194 days

20.3.2 Compare Channels

Two compare channels are available in the RTC. The compare values can be set by writing to the RTC compare channel registers RTC_COMPn, and when RTC_CNT is equal to one of these, the respective compare interrupt flag COMPn is set.

If COMP0TOP is set, the compare value set for compare channel 0 is used as a top value for the RTC, and the timer is cleared on a compare match with compare channel 0. If using the COMP0TOP setting, make sure to set this bit prior to or at the same time the EN bit is set. Setting COMP0TOP after the EN bit is set may cause unintended operation (i.e. if CNT > COMP0).

20.3.2.1 LETIMER Triggers

A compare event on either of the compare channels can start the LETIMER. See the LETIMER documentation for more information on this feature.

20.3.2.2 PRS Sources

Both the compare channels of the RTC can be used as PRS sources. They will generate a pulse lasting one RTC clock cycle on a compare match.

20.3.3 Interrupts

The interrupts generated by the RTC are combined into one interrupt vector. If interrupts for the RTC is enabled, an interrupt will be made if one or more of the interrupt flags in RTC_IF and their corresponding bits in RTC_IEN are set. Interrupt events are overflow and compare match on either compare channels. Clearing of an interrupt flag is performed by writing to the corresponding bit in the RTC_IFC register.



20.3.4 Debugrun

By default, the RTC is halted when code execution is halted from the debugger. By setting the DEBUGRUN bit in the RTC_CTRL register, the RTC will continue to run even when the debugger is halted.

20.3.5 Register access

Since this module is a Low Energy Peripheral, and runs off a clock which is asynchronous to the HFCORECLK, special considerations must be taken when accessing registers. Please refer to Section 5.3.1.1 (p. 19) for a description on how to perform register accesses to Low Energy Peripherals.



20.4 Register Map

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

Offset	Name	Туре	Description
0x000	RTC_CTRL	RW	Control Register
0x004	RTC_CNT	R	Counter Value Register
0x008	RTC_COMP0	RW	Compare Value Register 0
0x00C	RTC_COMP1	RW	Compare Value Register 1
0x010	RTC_IF	R	Interrupt Flag Register
0x014	RTC_IFS	W1	Interrupt Flag Set Register
0x018	RTC_IFC	W1	Interrupt Flag Clear Register
0x01C	RTC_IEN	RW	Interrupt Enable Register
0x020	RTC_FREEZE	RW	Freeze Register
0x024	RTC_SYNCBUSY	R	Synchronization Busy Register

20.5 Register Description

20.5.1 RTC_CTRL - Control Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19) .

Offset									,						Bi	t Pc	siti	on														
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	=======================================	10	6	∞	7	9	2	4	က	2	-	0
Reset									•												•								-	0	0	0
Access																														ΑW	RW	RW W
Name																														СОМРОТОР	DEBUGRUN	EN

Bit	Name		Reset	Access	s Description
31:3	Reserved		To ensure c	ompatibility witl	h future devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	COMP0TOP		0	RW	Compare Channel 0 is Top Value
	When set, the co	unter is cl	eared in the cl	ock cycle after	a compare match with compare channel 0.
	Value	Mode		ı	Description
	0	DISABLE		-	The top value of the RTC is 16777215 (0xFFFFFF)
	1	ENABLE		-	The top value of the RTC is given by COMP0
1	DEBUGRUN		0	RW	Debug Mode Run Enable
	Set this bit to ena	able the R	TC to keep rur	nning in debug.	
	Value		Description		
	0		RTC is frozen in	debug mode	
	1		RTC is running i	n debug mode	
0	EN		0	RW	RTC Enable
	When this bit is s	et, the RT	C is enabled a	and counts up. \	When cleared, the counter register CNT is reset.



20.5.2 RTC_CNT - Counter Value Register

Offset															Bit I	Posit	ion														
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	1 1	16	15	14	13	12	1	10	6	8	7	9	2	4	3	2	-	0
Reset																				000000×0											
Access																				ď											
Name																				CNT											

Bit	Name	Reset	Access	Description
31:24	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
23:0	CNT	0x000000	R	Counter Value
	Gives access to the counter	r value of the RTC.		

20.5.3 RTC_COMP0 - Compare Value Register 0 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

Offset															Bi	t Po	siti	on														
0x008	33	30	29	28	27	56	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	თ	∞	7	9	2	4	ю	2	-	0
Reset																					00000000											
Access																					R⊗											
Name																					COMPO											

Bit	Name	Reset	Access	Description
31:24	Reserved	To ensure comp	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
23:0	COMP0	0x000000	RW	Compare Value 0
	A compare match event oc the LETIMER. It is also ava			alue. This event sets the COMP0 interrupt flag, and can be used to start

20.5.4 RTC_COMP1 - Compare Value Register 1 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).



Offset									,						Bi	t Po	siti	on					,		,							
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	=	10	6	ω	7	9	2	4	က	2	-	0
Reset																					000000x0											
Access																					₩ N											
Name																					COMP1											

Bit	Name	Reset	Access	Description
31:24	Reserved	To ensure comp	atibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
23:0	COMP1	0x000000	RW	Compare Value 1
	A compare match event of the LETIMER. It is also available.		•	value. This event sets COMP1 interrupt flag, and can be used to start

20.5.5 RTC_IF - Interrupt Flag Register

Offset															Bi	t Pc	siti	on														
0x010	31	30	29	28	27	56	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	ю	2	-	0
Reset					•			•	•							•	•			•										0	0	0
Access		-																												œ	œ	2
Name																														COMP1	сомРо	OF

Bit	Name	Reset	Access	Description
31:3	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	COMP1	0	R	Compare Match 1 Interrupt Flag
	Set on a compare ma	atch between CNT a	nd COMP1.	
1	COMP0	0	R	Compare Match 0 Interrupt Flag
	Set on a compare ma	atch between CNT a	nd COMP0.	
0	OF	0	R	Overflow Interrupt Flag
	Set on a CNT value	overflow.		

20.5.6 RTC_IFS - Interrupt Flag Set Register

Offset								·	·						Bi	t Pc	siti	on														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	ю	2	-	0
Reset				,						•			•		,														,	0	0	0
Access																														W	W1	W N
Name																														COMP1	COMPO	OF



Bit	Name	Reset	Access	Description
31:3	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	COMP1	0	W1	Set Compare match 1 Interrupt Flag
	Write to 1 to set the C	OMP1 interrupt flag	J.	
1	COMP0	0	W1	Set Compare match 0 Interrupt Flag
	Write to 1 to set the C	OMP0 interrupt flag	J.	
0	OF	0	W1	Set Overflow Interrupt Flag
	Write to 1 to set the O	F interrupt flag.		

20.5.7 RTC_IFC - Interrupt Flag Clear Register

Offset															Bi	t Po	siti	on														
0x018	31	98	59	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	80	7	9	2	4	က	2	-	0
Reset			•	•						•	•	•	•		,				•							•			•	0	0	0
Access																														ž	×	×
Name																														COMP1	COMPO	OF

Bit	Name	Reset	Access	Description
31:3	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	COMP1	0	W1	Clear Compare match 1 Interrupt Flag
	Write to 1 to clear the C	OMP1 interrupt fla	ıg.	
1	COMP0	0	W1	Clear Compare match 0 Interrupt Flag
	Write to 1 to clear the C	OMP0 interrupt fla	ıg.	
0	OF	0	W1	Clear Overflow Interrupt Flag
	Write to 1 to clear the O	F interrupt flag.		

20.5.8 RTC_IEN - Interrupt Enable Register

Offset															Bi	t Pc	siti	on														
0x01C	31	30	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	8	7	9	2	4	ю	2	-	0
Reset															•														,	0	0	0
Access																														RW	RW	RW
Name																														COMP1	COMPO	OF

Bit	Name	Reset	Access	Description
31:3	Reserved	To ensure co.	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	COMP1	0	RW	Compare Match 1 Interrupt Enable
	Enable interrupt or	n compare match 1.		
1	COMP0	0	RW	Compare Match 0 Interrupt Enable
	Enable interrupt or	n compare match 0.		
0	OF	0	RW	Overflow Interrupt Enable



Bit	Name	Reset	Access	Description
	Enable interrupt on overflo	w.		

20.5.9 RTC_FREEZE - Freeze Register

Offset															Bi	t Pc	siti	on														
0x020	31	30	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	8	7	9	2	4	ю	2	-	0
Reset																							•									0
Access																																RW
Name																																REGFREEZE

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure co	ompatibility with fut	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	REGFREEZE	0	RW	Register Update Freeze
	When set, the up	date of the RTC is postpo	oned until this bit i	s cleared. Use this bit to update several registers simultaneously.
	When set, the up	date of the RTC is postpo		s cleared. Use this bit to update several registers simultaneously.
			Desc Each	

20.5.10 RTC_SYNCBUSY - Synchronization Busy Register

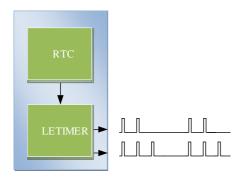
Offset															Bi	t Pc	siti	on														
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	œ	7	9	2	4	ю	2	-	0
Reset															,														,	0	0	0
Access																														~	ď	~
Name																														COMP1	COMPO	CTRL

Bit	Name	Reset	Access	Description
31:3	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	COMP1	0	R	COMP1 Register Busy
	Set when the value	written to COMP1 is b	eing synchronized	d.
1	COMP0	0	R	COMP0 Register Busy
	Set when the value	written to COMP0 is b	eing synchronized	d.
0	CTRL	0	R	CTRL Register Busy
	Set when the value	written to CTRL is bei	ng synchronized.	



21 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 32.768 Hz clock the LETIMER is available in EM2 with sub μ A current consumption.

Why?

The LETIMER can be used to provide repeatable waveforms to external components while remaining in EM2. It is well suited for e.g. metering systems or to provide more compare values than available in the RTC.

How?

With buffered repeat and top value registers, the LETIMER can provide glitch-free waveforms at frequencies up to 16 kHz. It is tightly coupled to the RTC, which allows advanced time-keeping and wake-up functions in EM2.

21.1 Introduction

The unique LETIMERTM, the Low Energy Timer, is a 16-bit timer that is available in energy mode EM2, in addition to EM1 and EM0. 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.

The LETIMER can be used to output a variety of waveforms with minimal software intervention. It is also connected to the Real Time Counter (RTC), and can be configured to start counting on compare matches from the RTC.

21.2 Features

- 16-bit down count timer
- · 2 Compare match registers
- Compare register 0 can be top timer top value
- · Compare registers can be double buffered
- Double buffered 8-bit Repeat Register
- Same clock source as the Real Time Counter
- LETIMER can be triggered (started) by an RTC event or by software
- 2 output pins can optionally be configured to provide different waveforms on timer underflow:
 - Toggle output pin
 - Apply a positive pulse (pulse width of one LFACLK_{I ETIMER} period)
 - PWM
- Interrupt on:
 - Compare matches
 - · Timer underflow

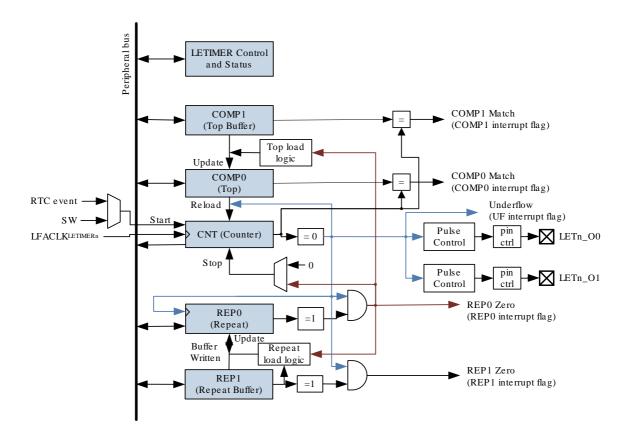


- Repeat done
- · Optionally runs during debug

21.3 Functional Description

An overview of the LETIMER module is shown in Figure 21.1 (p. 295). The LETIMER is a 16-bit down-counter with two compare registers, LETIMERn_COMP0 and LETIMERn_COMP1. The LETIMERn_COMP0 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_COMP0 and LETIMERn_REP0 registers can be double buffered by the LETIMERn_COMP1 and LETIMERn_REP1 registers to allow continuous operation. The timer can generate a single pin output, or two linked outputs.

Figure 21.1. LETIMER Overview



21.3.1 Timer

The timer is started by setting command bit START in LETIMERn_CMD, and stopped by setting the STOP command bit in the same register. RUNNING in LETIMERn_STATUS is set as long as the timer is running. The timer can also be started on external signals, such as a compare match from the Real Time Counter. If START and STOP are set at the same time, STOP has priority, and the timer will be stopped.

The timer value can be read using the LETIMERn_CNT register. The value cannot be written, but it can 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.

21.3.2 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 becomes equal to their value. When LETIMERn_CNT becomes equal to the value of LETIMERn_COMP0, the interrupt flag COMP0 in LETIMERn_IF is set, and when LETIMERn_CNT becomes equal to the value of LETIMERn COMP1, the interrupt flag COMP1 in LETIMERn IF is set.

21.3.3 Top Value

If COMP0TOP in LETIMERn_CTRL is set, the value of LETIMERn_COMP0 acts as the top value of the timer, and LETIMERn_COMP0 is loaded into LETIMERn_CNT on timer underflow. Else, the timer wraps around to 0xFFFF. The underflow interrupt flag UF in LETIMERn_IF is set when the timer reaches zero.

21.3.3.1 Buffered Top Value

If BUFTOP in LETIMERn_CTRL is set, the value of LETIMERn_COMP0 is buffered by LETIMERn_COMP1. In this mode, the value of LETIMERn_COMP1 is loaded into LETIMERn_COMP0 every time LETIMERn_REP0 is about to decrement to 0. This can for instance be used in conjunction with the buffered repeat mode to generate continually changing output waveforms.

Write operations to LETIMERn_COMP0 have priority over buffer loads.

21.3.3.2 Repeat Modes

By default, the timer wraps around to the top value or 0xFFFF 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 21.1 (p. 296).

Table 21.1. LETIMER Repeat Modes

REPMODE	Mode	Description
00	Free	The timer runs until it is stopped
01	One-shot	The timer runs as long as LETIMERn_REP0 != 0. LETIMERn_REP0 is decremented at each timer underflow.
10	Buffered	The timer runs as long as LETIMERn_REP0 != 0. LETIMERn_REP0 is decremented on each timer underflow. If LETIMERn_REP1 has been written, it is loaded into LETIMERn_REP0 when LETIMERn_REP0 is about to be decremented to 0.
11	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.

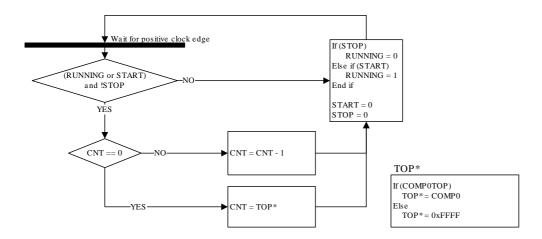
The interrupt flags REP0 and REP1 in LETIMERn_IF are set whenever LETIMERn_REP0 or LETIMERn_REP1 are decremented to 0 respectively. REP0 is also set when the value of LETIMERn_REP1 is loaded into LETIMERn_REP0 in buffered mode.

21.3.3.2.1 Free Mode

In the free running mode, the LETIMER acts as a regular timer, and the repeat counter is disabled. When started, the timer runs until it is stopped using the STOP command bit in LETIMERn_CMD. A state machine for this mode is shown in Figure 21.2 (p. 297) .



Figure 21.2. LETIMER State Machine for Free-running Mode



Note that the CLEAR command bit in LETIMERn_CMD always has priority over other changes 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 0xFFFF. Since no underflow event is generated, no output action is performed. LETIMERn_REP0, LETIMERn_REP1, LETIMERn_COMP0 and LETIMERn_COMP1 are also left untouched.

21.3.3.2.2 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.

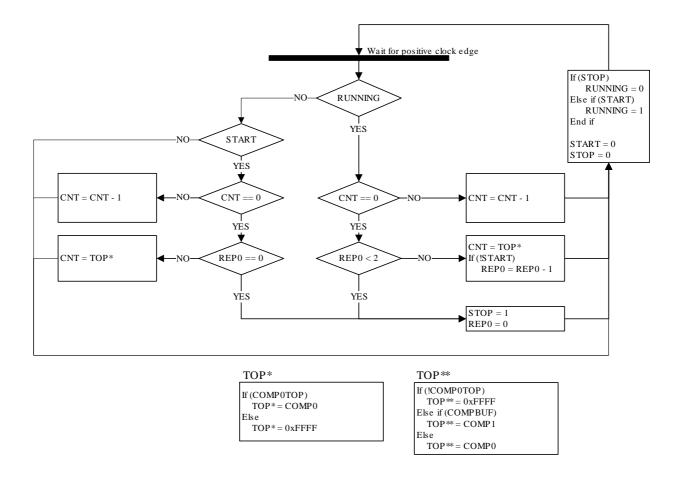
Note

Note that write operations to LETIMERn_REP0 have priority over the decrementation operation. So if LETIMERn_REP0 is assigned a new value in the same cycle it was supposed to be decremented, it is assigned the new value instead of being decremented.

LETIMERn_REP0 can be written while the timer is running to allow the timer to run for longer periods at a time without stopping. Figure 21.3 (p. 298) .



Figure 21.3. LETIMER One-shot Repeat State Machine



21.3.3.2.3 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 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.

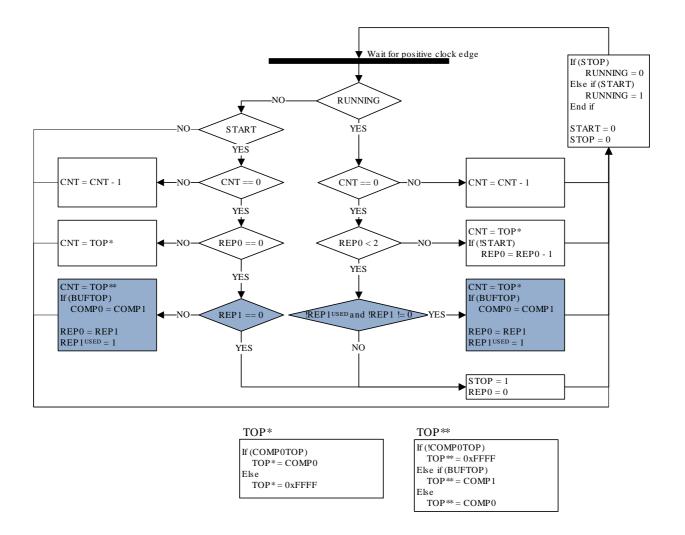
If the timer is started when both LETIMERn_CNT and LETIMERn_REP0 are zero but LETIMERn_REP1 is non-zero, LETIMERn_REP1 is loaded into LETIMERn_REP0, and the counter counts the loaded number of times. The state machine for the one-shot repeat mode is shown in Figure 21.3 (p. 298).

Used in conjunction with a buffered top value, enabled by setting BUFTOP in LETIMERn_CTRL, the buffered mode allows buffered values of both the top and repeat values of the timer, 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 21.4 (p. 299). 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 written.



Figure 21.4. LETIMER Buffered Repeat State Machine

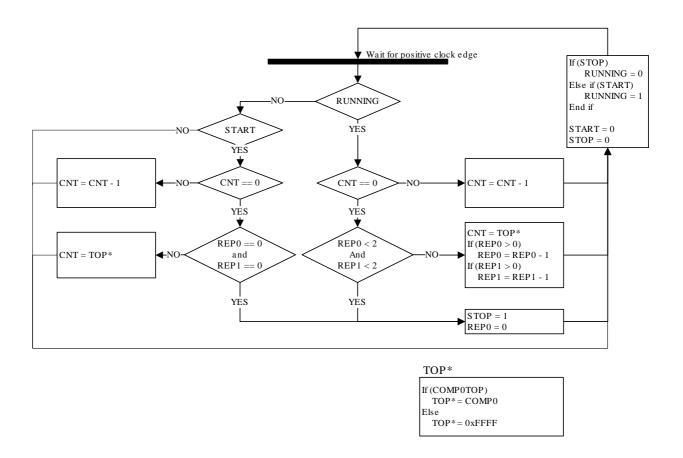


21.3.3.2.4 Double Mode

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 21.5 (p. 300) .



Figure 21.5. LETIMER Double Repeat State Machine



21.3.3.3 Clock Source

The LETIMER clock source and its prescaler value are defined in the Clock Management Unit (CMU). The LFACLK_{LETIMERn} has a frequency given by Equation 21.1 (p. 300).

LETIMER Clock Frequency
$$f_{LFACKL_LETIMERn} = 32.768/2^{LETIMERn}$$
 (21.1)

where the exponent LETIMERn is a 4 bit value in the CMU_LFAPRESC0 register.

To use this module, the LE interface clock must be enabled in CMU_HFCORECLKEN0, in addition to the module clock.

21.3.3.4 RTC Trigger

The LETIMER can be configured to start on compare match events from the Real Time Counter (RTC). If RTCC0TEN in LETIMERn_CTRL is set, the LETIMER will start on a compare match on RTC compare channel 0. In the same way, RTCC1TEN in LETIMERn_CTRL enables the LETIMER to start on a compare match with RTC compare channel 1.

Note

The LETIMER can only use compare match events from the RTC if the LETIMER runs at a higher than or equal frequency than the RTC. Also, if the LETIMER runs at twice the frequency of the RTC, a compare match event in the RTC will trigger the LETIMER twice. Four times the frequency gives four consecutive triggers, etc. The LETIMER will only



continue running if triggered while it is running, so the multiple-triggering will only have an effect if you try to disable the RTC when it is being triggered.

21.3.3.5 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.

21.3.4 Underflow Output Action

For each of the repeat registers, an underflow output action can be set. The configured output action is performed every time the counter underflows while the respective repeat register is nonzero. In PWM mode, the output is similarly only changed on COMP1 match if the repeat register is nonzero. As an example, the timer will perform 7 output actions if LETIMERn_REP0 is set to 7 when starting the timer in one-shot mode and leaving it untouched for a while.

The output actions can be set by configuring UFOA0 and UFOA1 in LETIMERn_CTRL. UFOA0 defines the action on output 0, and is connected to LETIMERn_REP0, while UFOA1 defines the action on output 1 and is connected to LETIMERn_REP1. The possible actions are defined in Table 21.2 (p. 301).

Table 21.2. LETIMER Underflow Output Actions	Table 21.2.	LETIMER	Underflow	Output	Actions
--	-------------	----------------	------------------	---------------	---------

UF0A0/UF0A1	Mode	Description
00	Idle	The output is held at its idle value
01	Toggle	The output is toggled on LETIMERn_CNT underflow if LEIMERn_REPx is nonzero
10	Pulse	The output is held active for one clock cycle on LETIMERn_CNT underflow if LETIMERn_REPx is nonzero. It then returns to its idle value
11	PWM	The output is set idle on LETIMERn_CNT underflow and active on compare match with LETIMERn_COMP1 if LETIMERn_REPx is nonzero.

Note

For the Pulse and PWM modes, the outputs will return to their idle states regardless of the state of the corresponding LETIMERn_REPx registers. They will only be set active if the LETIMERn_REPx registers are nonzero however.

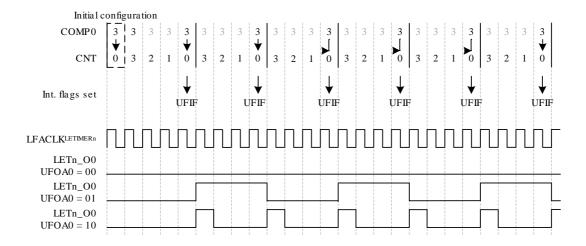
The polarity of the outputs can be set individually by configuring OPOL0 and OPOL1 in LETIMERn_CTRL. When these are cleared, their respective outputs have a low idle value and a high active value. When they are set, the idle value is high, and the active value is low.

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.



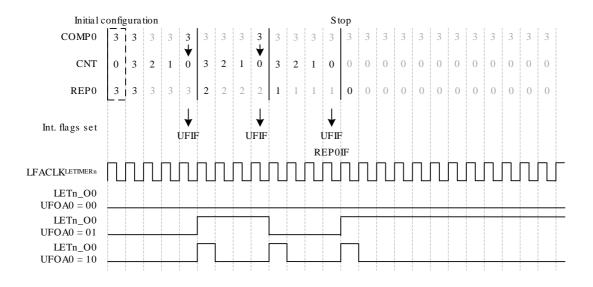
Some simple waveforms generated with the different output modes are shown in Figure 21.6 (p. 302) . For the example, REPMODE in LETIMERn_CTRL has been cleared, COMP0TOP also in LETIMERn_CTRL has been set and LETIMERn_COMP0 has been written to 3. As seen in the figure, LETIMERn_COMP0 now decides the length of the signal periods. For the toggle mode, the period of the output signal is 2(LETIMERn_COMP0 + 1), and for the pulse modes, the periods of the output signals are LETIMERn_COMP0+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 21.6. LETIMER Simple Waveforms Output



For the example in Figure 21.7 (p. 302), the One-shot repeat mode has been selected, 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.

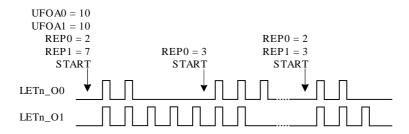
Figure 21.7. LETIMER Repeated Counting



Using the Double repeat mode, output can be generated on both the LETIMER outputs. Figure 21.8 (p. 303) shows an example of this. 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 21.8. LETIMER Dual Output



21.3.5 Examples

This section presents a couple of usage examples for the LETIMER.

21.3.5.1 Triggered Output Generation

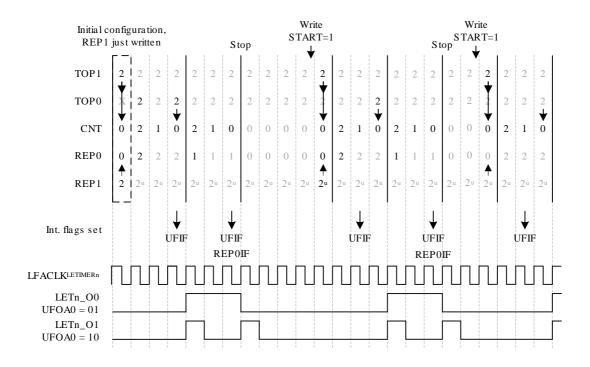
Example 21.1. LETIMER Triggered Output Generation

If both LETIMERn_CNT and LETIMERn_REP0 are 0 in buffered mode, and COMP0TOP and BUFTOP in LETIMERn_CTRL are set, the values of LETIMERn_COMP1 and LETIMERn_REP1 are loaded into LETIMERn_CNT 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_COMP1 determines the period lengths.

As the RTC can be used to start the LETIMER, the RTC and LETIMER can thus be combined to generate specific pulse-trains at given intervals. Software can update LETIMERn_COMP1 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 21.9 (p. 303), 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 21.9. LETIMER Triggered Operation





21.3.5.2 Continuous Output Generation

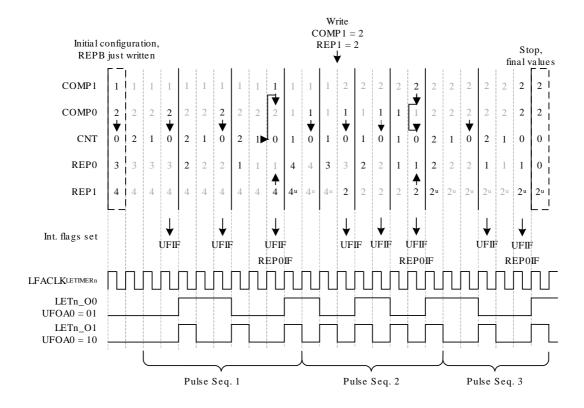
Example 21.2. LETIMER Continuous Output Generation

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 21.6 (p. 302), but to generate changing waveforms, using the repeat counter and buffer registers can prove advantageous.

For the example in Figure 21.10 (p. 304), 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 21.10. LETIMER Continuous Operation



The first two sequences are loaded into the LETIMER before the timer is started.

LETIMERn_COMP0 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. COMP1 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.

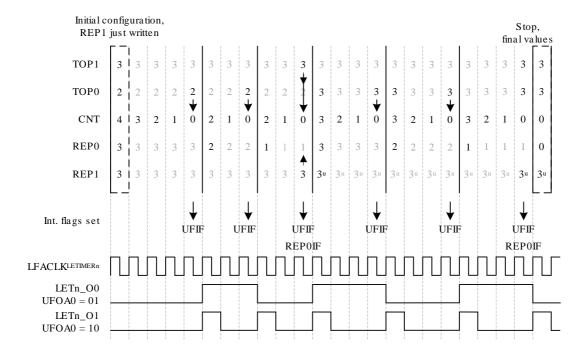
The final result is shown in Figure 21.10 (p. 304). The pulse output is grouped to show which sequence generated which output. Toggle output is also shown in the figure. Note that the toggle output is not aligned with the pulse outputs



Multiple LETIMER cycles are required to write a value to the LETIMER registers. The example in Figure 21.10 (p. 304) assumes that writes are done in advance so they arrive in the LETIMER as described in the figure.

Figure 21.11 (p. 305) 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 21.11. LETIMER LETIMERn_CNT Not Initialized to 0



21.3.5.3 PWM Output

Example 21.3. LETIMER PWM Output

There are several ways of generating PWM output with the LETIMER, but the most straight-forward way is using the PWM output mode. This mode is enabled by setting UFOA0 or OFUA1 in LETIMERn_CTRL to 3. In PWM mode, the output is set idle on timer underflow, and active on LETIMERn_COMP1 match, so if for instance COMP0TOP = 1 and OPOL0 = 0 in LETIMERn_CTRL, LETIMERn_COMP0 determines the PWM period, and LETIMERn_LETIMERn_COMP1 determines the active period.

The PWM period in PWM mode is LETIMERn_COMP0 + 1. There is no special handling of the case where LETIMERn_COMP1 > LETIMERn_COMP0, so if LETIMERn_COMP1 > LETIMERn_COMP0, 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_COMP1 is set to a value larger than LETIMERn_COMP0.

21.3.5.4 Interrupts

Example 21.4. LETIMER PWM Output

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.



21.3.6 Register access

Since this module is a Low Energy Peripheral, and runs off a clock which is asynchronous to the HFCORECLK, special considerations must be taken when accessing registers. Please refer to Section 5.3.1.1 (p. 19) for a description on how to perform register accesses to Low Energy Peripherals.



21.4 Register Map

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

Offset	Name	Туре	Description
0x000	LETIMERn_CTRL	RW	Control Register
0x004	LETIMERn_CMD	W1	Command Register
0x008	LETIMERn_STATUS	R	Status Register
0x00C	LETIMERn_CNT	R	Counter Value Register
0x010	LETIMERn_COMP0	RW	Compare Value Register 0
0x014	LETIMERn_COMP1	RW	Compare Value Register 1
0x018	LETIMERn_REP0	RW	Repeat Counter Register 0
0x01C	LETIMERn_REP1	RW	Repeat Counter Register 1
0x020	LETIMERn_IF	R	Interrupt Flag Register
0x024	LETIMERn_IFS	W1	Interrupt Flag Set Register
0x028	LETIMERn_IFC	W1	Interrupt Flag Clear Register
0x02C	LETIMERn_IEN	RW	Interrupt Enable Register
0x030	LETIMERn_FREEZE	RW	Freeze Register
0x034	LETIMERn_SYNCBUSY	R	Synchronization Busy Register
0x040	LETIMERn_ROUTE	RW	I/O Routing Register

21.5 Register Description

21.5.1 LETIMERn_CTRL - Control Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

Offset	Bit Position							
0x000	33 34 35 36 37 38 39 30 30 31 32 33 34 35 36 37 38 39 30 30 30 30 41 41 41 41 42 43 44 44 44 44 44 44 44 45 46 47 48 49 40 40 40 40 40 40 40 40 40 41 42 43 44 44 44 45 46 47 48 48 49 40 40 40 40 40 <th>11 11</th> <th>10</th> <th>8</th> <th>7</th> <th>0 0 4</th> <th>2 3</th> <th>- 0</th>	11 11	10	8	7	0 0 4	2 3	- 0
Reset		0 0	0 0	0	0	0x0	0x0	0x0
Access		RW WW	RW W	RW	RW S	X X	RW	RW
Name		DEBUGRUN RTCC1TEN	RTCC0TEN COMP0TOP	BUFTOP	OPOL1	UFOA1	UFOA0	REPMODE

Bit	Name	Reset	Access	Description
31:13	Reserved	To ensure co	ompatibility with fu	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
12	DEBUGRUN	0	RW	Debug Mode Run Enable
	Set to keep the LE	ΓΙΜΕR running in debu	ıg mode.	
	Value	Description		
	0	LETIMER is froze	en in debug mode	
	1	LETIMER is runn	ing in debug mode	
11	RTCC1TEN	0	RW	RTC Compare 1 Trigger Enable
	Allows the LETIME	R to be started on a co	mpare match on	RTC compare channel 1.
	Value	Description		
	0	npare channel 1		



Bit	Name	Reset	Access	Description
	Value	Description		
	1	A compare match	on RTC compare	e channel 1 starts the LETIMER if the LETIMER is not already started
10	RTCC0TEN	0	RW	RTC Compare 0 Trigger Enable
	Allows the LET	TIMER to be started on a cor	mpare match or	n RTC compare channel 0.
	Value	Description		
	0	LETIMER is not a	fected by RTC co	ompare channel 0
	1	A compare match	on RTC compare	e channel 0 starts the LETIMER if the LETIMER is not already started
9	COMPOTOP	0	RW	Compare Value 0 Is Top Value
	When set, the	counter is cleared in the clo	ck cycle after a	compare match with compare channel 0.
	Value	Description		
	0	The top value of the	ne LETIMER is 6	5535 (0xFFFF)
	1	The top value of the		
8	BUFTOP	0	RW	Buffered Top
	Set to load CO	MP1 into COMP0 when RE	P0 reaches 0. a	allowing a buffered top value
	Value			
	0	Description COMP0 is only wr	itten hv software	
	1	COMP0 is set to 0	•	P0 reaches 0
7	OPOL1	0	RW	Output 1 Polarity
,		-	IXVV	Output I Folianty
		e value of output 1.	5111	
6	OPOL0	0	RW	Output 0 Polarity
	Defines the idle	e value of output 0.		
5:4	UFOA1	0x0	RW	Underflow Output Action 1
	Defines the ac	tion on LETn_O1 on a LETII	MER underflow	
	Value	Mode	De	escription
	0	NONE	LE	ETn_O1 is held at its idle value as defined by OPOL1.
	1	TOGGLE		ETn_O1 is toggled on CNT underflow.
	2	PULSE		ETn_O1 is held active for one LFACLK _{LETIMERO} clock cycle on CNT underflow. The atput then returns to its idle value as defined by OPOL1.
	3	PWM	LE	ETn_O1 is set idle on CNT underflow, and active on compare match with COMP1
3:2	UFOA0	0x0	RW	Underflow Output Action 0
	Defines the ac	tion on LETn_O0 on a LETII		·
	Value	Mode		
	0	NONE		escription ETn_O0 is held at its idle value as defined by OPOL0.
	1	TOGGLE		ETn_O0 is toggled on CNT underflow.
	2	PULSE	LE	ETn_O0 is held active for one LFACLK _{LETIMER0} clock cycle on CNT underflow. The
	_			utput then returns to its idle value as defined by OPOL0.
	3	PWM	LE	ETn_O0 is set idle on CNT underflow, and active on compare match with COMP1
1:0	REPMODE	0x0	RW	Repeat Mode
	Allows the repe	eat counter to be enabled ar	nd disabled.	
	Value	Mode	De	escription
	0	FREE	W	hen started, the LETIMER counts down until it is stopped by software.
	1	ONESHOT	Th	ne counter counts REP0 times. When REP0 reaches zero, the counter stops.
	2	BUFFERED		ne counter counts REP0 times. If REP1 has been written, it is loaded into REP0 when EP0 reaches zero. Else the counter stops
	3	DOUBLE		oth REP0 and REP1 are decremented when the LETIMER wraps around. The



21.5.2 LETIMERn_CMD - Command Register

Offset	Bit Position	
0x004	3 3 4 5 5 6 6 7 7 8 8 8 6 7 7 8 8 8 7 7 8 8 8 8 7 9 9 9 9 9 9 9 9	- 0
Reset		0 0
Access	\times \	× ×
Name		STOP

Bit	Name	Reset	Access	Description
31:5	Reserved	To ensure comp	atibility with fut	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
4	CTO1	0	W1	Clear Toggle Output 1
	Set to drive toggle output	1 to its idle value		
3	CTO0	0	W1	Clear Toggle Output 0
	Set to drive toggle output	0 to its idle value		
2	CLEAR	0	W1	Clear LETIMER
	Set to clear LETIMER			
1	STOP	0	W1	Stop LETIMER
	Set to stop LETIMER			
0	START	0	W1	Start LETIMER
	Set to start LETIMER			

21.5.3 LETIMERn_STATUS - Status Register

Offset	Bit Position																															
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	တ	∞	7	9	2	4	က	2	-	0
Reset										•																						0
Access																																~
Name																																RUNNING

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure compa	atibility with futu	re devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	RUNNING	0	R	LETIMER Running
	Set when LETIMER is run	ning.		



21.5.4 LETIMERn_CNT - Counter Value Register

Offset															Bi	t Pc	siti	on														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	1	10	6	8	7	9	5	4	3	2	-	0
Reset																								0000	000000							
Access																								۵	Ľ							
Name																								Ė	5							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure comp	atibility with fut	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	CNT	0x0000	R	Counter Value
	Use to read the current va	lue of the LETIMER	₹.	

21.5.5 LETIMERn_COMP0 - Compare Value Register 0 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

Offset																osition																
0x010	31	98	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	80	7	9	2	4	က	2	-	0
Reset																								0000	000000							
Access																								74.0	^ Ƴ							
Name																																

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure compa	tibility with futu	re devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	COMP0	0x0000	RW	Compare Value 0
	Compare and optionally top	value for LETIMER	₹	

21.5.6 LETIMERn_COMP1 - Compare Value Register 1 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19) .



Offset			I	Bit Position													
0x014	30 30 28 27	25 23 23 24 25 25 25 23 24 24 25 25 25 25 25 25 25 25 25 25 25 25 25	19 18 17	2 9 5 7 7 1 2 7 1 0 0 8 7 9 9 7 1 0 0													
Reset				0000×0													
Access				% ≥													
Name				COMP1													
Bit	Name	Reset	Access	Description													
31:16	Reserved	To ensure com	patibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)													
15:0	COMP1	0x0000	RW	Compare Value 1													

21.5.7 LETIMERn_REP0 - Repeat Counter Register 0 (Async Reg)

Compare and optionally buffered top value for LETIMER

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

									•					·		•								٧.		,						
Offset				,											Bi	t Pc	siti	on														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	8	7	9	5	4	3	2	-	0
Reset																													0x00			
Access																													S.			
Name																													REP0			
Bit	Na	me						Re	set			A	\CC	ess		De	scri	iptic	on													

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure compa	atibility with futu	re devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:0	REP0	0x00	RW	Repeat Counter 0
	Optional repeat counter.			

21.5.8 LETIMERn_REP1 - Repeat Counter Register 1 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

														_																		
Offset															Bi	t Po	siti	on														
0x01C	31	90	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	80	7	9	2	4	က	2	1	0
Reset																													00×0			
Access																													χ ≷			
Name																													REP1			
Bit	Nar	ne						Re	set			Α	\cc	ess		De	scr	iptic	on													

Reserved

31:8

To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)



Bit	Name	Reset	Access	Description
7:0	REP1	0x00	RW	Repeat Counter 1
	Optional repeat	counter or buffer for REP0		

21.5.9 LETIMERn_IF - Interrupt Flag Register

Offset															Bi	t Pc	siti	on														
0x020	31	30	29	28	27	26	22	24	23	22	21	20	19	9	17	16	15	4	13	12	7	10	6	80	7	9	2	4	ю	2	-	0
Reset					•			•	•			•					•				•							0	0	0	0	0
Access																												~	~	~	~	<u>~</u>
Name																												REP1	REP0	UF	COMP1	COMPO

Bit	Name	Reset	Access	Description											
31:5	Reserved	To ensure comp	atibility with fut	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)											
4	REP1	0	R	Repeat Counter 1 Interrupt Flag											
	Set when repeat counter 1	reaches zero.													
3	REP0	0 R Repeat Counter 0 Interrupt Flag at counter 0 reaches zero or when the REP1 interrupt flag is loaded into the REP0 interrupt flag.													
	Set when repeat counter 0	reaches zero or wi	hen the REP1	interrupt flag is loaded into the REP0 interrupt flag.											
2	UF	0	R	Underflow Interrupt Flag											
	Set on LETIMER underflow	٧.													
1	COMP1	0	R	Compare Match 1 Interrupt Flag											
	Set when LETIMER reache	es the value of CON	MP1												
0	COMP0	0	R	Compare Match 0 Interrupt Flag											
	Set when LETIMER reache	es the value of CO	MP0												

21.5.10 LETIMERn_IFS - Interrupt Flag Set Register

Offset															Bi	t Pc	siti	on													,	
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	=======================================	10	6	∞	7	9	2	4	ю	2	-	0
Reset																•					•							0	0	0	0	0
Access																												W	Ž.	W	W1	X
Name																												REP1	REP0	UF	COMP1	COMPO

Bit	Name	Reset	Access	Description
31:5	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
4	REP1	0	W1	Set Repeat Counter 1 Interrupt Flag
	Write to 1 to set the	e REP1 interrupt flag.		
3	REP0	0	W1	Set Repeat Counter 0 Interrupt Flag
	Write to 1 to set the	e REP0 interrupt flag.		
2	UF	0	W1	Set Underflow Interrupt Flag
	Write to 1 to set the	e UF interrupt flag.		



Bit	Name	Reset	Access	Description
1	COMP1	0	W1	Set Compare Match 1 Interrupt Flag
	Write to 1 to set th	e COMP1 interrupt flag.		
0	COMP0	0	W1	Set Compare Match 0 Interrupt Flag
	Write to 1 to set th	e COMP0 interrupt flag.		

21.5.11 LETIMERn_IFC - Interrupt Flag Clear Register

Offset															Bi	t Po	ositi	on	·							-					,	
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	=======================================	10	6	∞	7	9	2	4	ю	2	-	0
Reset																					•							0	0	0	0	0
Access																												W1	W W	W1	W1	W W
Name																												REP1	REP0	UF	COMP1	COMPO

Bit	Name	Reset	Access	Description
31:5	Reserved	To ensure comp	atibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
4	REP1	0	W1	Clear Repeat Counter 1 Interrupt Flag
	Write to 1 to clear the REF	P1 interrupt flag.		
3	REP0	0	W1	Clear Repeat Counter 0 Interrupt Flag
	Write to 1 to clear the REF	0 interrupt flag.		
2	UF	0	W1	Clear Underflow Interrupt Flag
	Write to 1 to clear the UF	interrupt flag.		
1	COMP1	0	W1	Clear Compare Match 1 Interrupt Flag
	Write to 1 to clear the CO	MP1 interrupt flag.		
0	COMP0	0	W1	Clear Compare Match 0 Interrupt Flag
	Write to 1 to clear the CO	MP0 interrupt flag.		

21.5.12 LETIMERn_IEN - Interrupt Enable Register

Offset				,	,										Bi	t Pc	siti	on														
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	8	7	9	2	4	က	7	-	0
Reset						•		•	*						*							*	•					0	0	0	0	0
Access																												RW	RW	RW	RW	RW
Name																												REP1	REP0	Ą	COMP1	COMPO

Bit	Name	Reset	Access	Description
31:5	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
4	REP1	0	RW	Repeat Counter 1 Interrupt Enable
	Set to enable inter	rupt on the REP1 interru	upt flag.	
3	REP0	0	RW	Repeat Counter 0 Interrupt Enable
	Set to enable inter	rupt on the REP0 interru	upt flag.	



Bit	Name	Reset	Access	Description
2	UF	0	RW	Underflow Interrupt Enable
	Set to enable interrupt on	the UF interrupt flag].	
1	COMP1	0	RW	Compare Match 1 Interrupt Enable
	Set to enable interrupt on	the COMP1 interrup	ot flag.	
0	COMP0	0	RW	Compare Match 0 Interrupt Enable
	Set to enable interrupt on	the COMP0 interrup	ot flag.	

21.5.13 LETIMERn_FREEZE - Freeze Register

Offset															Bi	t Po	siti	on														
0x030	31	30	59	28	27	56	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	ო	7	-	0
Reset					•																											0
Access																																RW
Name																																REGFREEZE

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	REGFREEZE	0	RW	Register Update Freeze
	When set, the up	date of the LETIMER is p	postponed until th	is bit is cleared. Use this bit to update several registers simultaneously.
	When set, the up	date of the LETIMER is p	· ·	is bit is cleared. Use this bit to update several registers simultaneously.
	,	·	Des Eac	

21.5.14 LETIMERn_SYNCBUSY - Synchronization Busy Register

Offset															Bi	t Po	siti	on														
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	∞	7	9	2	4	ю	2	-	0
Reset																											0	0	0	0	0	0
Access																											~	~	~	~	22	œ
Name																											REP1	REP0	COMP1	COMPO	CMD	CTRL

Bit	Name	Reset	Access	Description
31:6	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
5	REP1	0	R	REP1 Register Busy
	Set when the value	written to REP1 is bei	ng synchronized.	
4	REP0	0	R	REP0 Register Busy
	Set when the value	written to REP0 is bei	ng synchronized.	
3	COMP1	0	R	COMP1 Register Busy
	Set when the value	written to COMP1 is b	eing synchronized	d.
2	COMP0	0	R	COMP0 Register Busy



Bit	Name	Reset	Access	Description
	Set when the value	e written to COMP0 is b	eing synchronize	d.
1	CMD	0	R	CMD Register Busy
	Set when the value	e written to CMD is bein	g synchronized.	
0	CTRL	0	R	CTRL Register Busy
	Set when the value	e written to CTRL is bei	ng synchronized.	

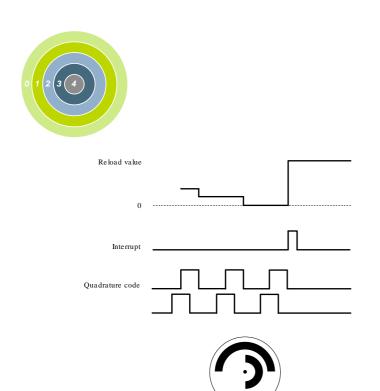
21.5.15 LETIMERn_ROUTE - I/O Routing Register

Offset															Bi	t Po	siti	on														
0x040	31	30	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	∞	7	9	2	4	က	2	-	0
Reset																							3	OXO							0	0
Access																							2	<u>}</u>							RW	RW
Name																							I S								OUT1PEN	OUTOPEN

				OUT
Name		Reset	Acce	ss Description
Reserved		To ensure co	ompatibility w	vith future devices, always write bits to 0. More information in Section 2.1 (p. 3)
LOCATION		0x0	RW	I/O Location
Decides the lo	cation of th	e LETIMER I/O	pins	
Value	Mode			Description
0	LOC0			Location 0
1	LOC1			Location 1
2	LOC2			Location 2
3	LOC3			Location 3
Reserved		To ensure co	ompatibility w	rith future devices, always write bits to 0. More information in Section 2.1 (p. 3)
OUT1PEN		0	RW	Output 1 Pin Enable
When set, outp	out 1 of the	LETIMER is ena	abled	
Value		Description		
0		The LETn_O1 pi	n is disabled	
1		The LETn_O1 pi	n is enabled	
OUT0PEN		0	RW	Output 0 Pin Enable
When set, outp	out 0 of the	LETIMER is ena	abled	
17.1		Description		
value		Description		
0		The LETn_O0 pi	n is disabled	
	Reserved LOCATION Decides the lo Value 0 1 2 3 Reserved OUT1PEN When set, output Value 0 1 OUT0PEN When set, output When set, output When set, output OUT0PEN	Reserved LOCATION Decides the location of the Value Mode 0 LOC0 1 LOC1 2 LOC2 3 LOC3 Reserved OUT1PEN When set, output 1 of the Value 0 1 OUT0PEN When set, output 0 of the	Reserved LOCATION Decides the location of the LETIMER I/O Value Mode LOC0 LOC0 LOC1 LOC2 LOC2 LOC3 Reserved To ensure companies and the LETIMER is ensured to the	Reserved To ensure compatibility with the LOCATION Decides the location of the LETIMER I/O pins Value Mode UCC0 LOC0 LOC1 LOC2 LOC2 LOC3 Reserved To ensure compatibility with the LETIMER is enabled Value Description UCC1 Description The LETIMER is enabled The LETIMER is enabled



22 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-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 LFACLK or may be externally clocked from a pin. The module incorporates an 8-bit up/down-counter to keep track of incoming pulses or rotations.

22.1 Introduction

The Pulse Counter (PCNT) can be used for counting incoming pulses on a single input or to decode quadrature encoded inputs. It can run from the internal LFACLK (EM0-EM2) while counting pulses on the PCNTn_S0IN pin or using this pin as an external clock source (EM0-EM3) that runs both the PCNT counter and register access.

22.2 Features

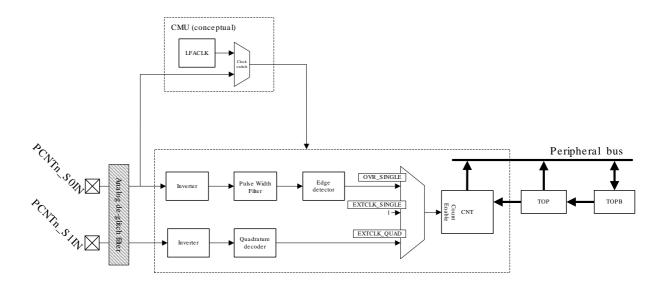
- · 8-bit counter with reload register
- Single input oversampling up/down counter mode (EM0-EM2)
- Externally clocked single input pulse up/down counter mode (EM0-EM3)
- Externally clocked quadrature decoder mode (EM0-EM3)
- · Interrupt on counter underflow and overflow
- Interrupt when a direction change is detected (quadrature decoder mode only)
- Optional pulse width filter
- · Optional input inversion/edge detect select

22.3 Functional Description

An overview of the PCNT module is shown in Figure 22.1 (p. 317) .



Figure 22.1. PCNT Overview



22.3.1 Pulse Counter Modes

The pulse counter can operate in single input oversampling mode (OVSSINGLE), externally clocked single input counter mode (EXTCLKSINGLE) and externally clocked quadrature decoder mode (EXTCLKQUAD). The following sections describe operation of each of the three modes and how they are enabled. Input timing constraints are described in Section 22.3.3 (p. 319) and Section 22.3.4 (p. 320).

22.3.1.1 Single Input Oversampling Mode

This mode is enabled by writing OVSSINGLE to the MODE field in the PCNTn_CTRL register and disabled by writing DISABLE to the same field. LFACLK is configured from the registers in the Clock Management Unit (CMU), Chapter 11 (p. 94).

The optional pulse width filter is enabled by setting the FILT bit in the PCNTn_CTRL 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.

PCNTn_S0IN is the only observed input in this mode. This input is sampled by the LFACLK and the number of detected positive or negative edges on PCNTn_S0IN appears in PCNTn_CNT. The counter may be configured to count down by setting the CNTDIR bit in PCNTn_CTRL. Default is to count up.

Only the underflow (UF) and overflow (OF) interrupt flags are set in this mode.

22.3.1.2 Externally Clocked Single Input Counter Mode

This mode is enabled by writing EXTCLKSINGLE to the MODE field in the PCNTn_CTRL register and disabled by writing DISABLE to the same field. The external pin clock source must be configured from the registers in the CMU (Chapter 11 (p. 94)).

Positive edges on PCNTn_S0IN are used to clock the counter. PCNTn_S1IN is ignored in this mode. As the LFACLK is not used in this mode, the PCNT module can operate in EM3. Like in the oversampling mode, the counter may be configured to count down by writing 1 to the CNTDIR bit in the PCNTn_CTRL register. Default is to count up.

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.

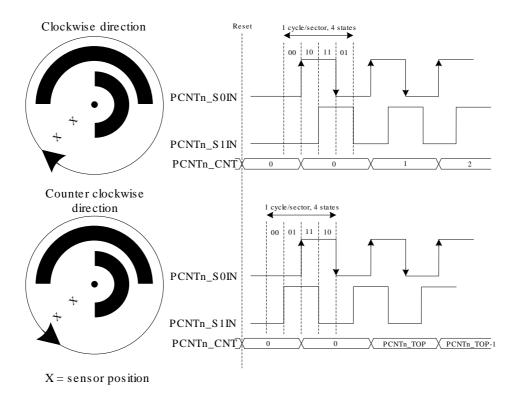
Only the underflow (UF) and overflow (OF) interrupt flags are set in this mode.

22.3.1.3 Externally Clocked Quadrature Decoder Mode

This mode is enabled by writing EXTCLKQUAD to the MODE field in PCNTn_CTRL and disabled by writing DISABLE to the same field. The external pin clock source must be configured from the registers in the CMU, (Chapter 11 (p. 94)).

Both edges on PCNTn_S0IN pin are used to sample PCNTn_S1IN pin to decode the quadrature code. Consequently, this mode does not depend on the internal LFACLK and may be operated in EM3. A quadrature coded signal contains information about the relative speed and direction of a rotating shaft as illustrated by Figure 22.2 (p. 318), hence the direction of the counter register PCNTn_CNT is controlled automatically.

Figure 22.2. 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. Although the direction is automatically detected, the detected direction may be inverted by writing 1 to the EDGE bit in the PCNTn_CTRL register. Default behavior is illustrated by Figure 22.2 (p. 318) .

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 22.1 (p. 319). Note that this function also filters some invalid inputs that may occur when the shaft changes direction or temporarily toggles direction.

Table 22.1. PCNT QUAD Mode Counter Control Function

Inputs		Control/Status	
S1IN posedge	S1IN negedge	Count Enable	CNTDIR status bit
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	0

Note

PCNTn_S1IN is sampled on both edges of PCNTn_S0IN.

22.3.2 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 (Chapter 11 (p. 94)).

When the RSTEN bit in the PCNTn_CTRL register is set to 1, the PCNT clock domain is asynchronously held in reset. The reset is synchronously released two PCNT clock edges after the RSTEN bit in the PCNTn_CTRL register is cleared by software. This asynchronous reset restores the reset values in PCNTn_TOP, PCNTn_CNT and other control registers in the PCNT clock domain.

Since this module is a Low Energy Peripheral, and runs off a clock which is asynchronous to the HFCORECLK, special considerations must be taken when accessing registers. Please refer to Section 5.3 (p. 19) for a description on how to perform register accesses to Low Energy Peripherals.

Note

PCNTn_TOP and PCNTn_CNT are read-only registers. When writing to PCNTn_TOPB, make sure that the counter value, PCNTn_CNT, can not exceed the value written to PCNTn_TOPB within two clock cycles.

22.3.3 Clock Sources

The 32 kHz LFACLK is one of two possible clock sources. The clock select register is described in Chapter 11 (p. 94). The default clock source is the LFACLK.

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 datasheet.

To use this module, the LE interface clock must be enabled in CMU_HFCORECLKEN0, in addition to the module clock.

Note

PCNT Clock Domain Reset, RSTEN, should be set when changing clock source for PCNT. In addition to this, the PCNTn_SYNCBUSY value should be zero. If changing to an external clock source, the clock pin has to be enabled as input prior to de-asserting RSTEN. Changing clock source without asserting RSTEN results in undefined behaviour.



22.3.4 Input Filter

An optional pulse width filter is available in OVSSINGLE mode. The filter is enabled by writing 1 to the FILT bit in the PCNTn_CTRL register. When enabled, the high and low periods of PCNTn_S0IN must be stable for 5 consecutive clock cycles before the edge is passed to the edge detector.

In EXTCLKSINGLE and EXTCLKQUAD mode, there is no digital pulse width filter available.

22.3.5 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 in OVSSINGLE mode and negative edges if the bit is set.

In EXTCLKQUAD mode, the EDGE bit in PCNTn_CTRL inverts the direction of the counter (which is automatically detected).

Note

The EDGE bit in PCNTn_CTRL has no effect in EXTCLKSINGLE mode.

22.3.6 PRS Sources

The PCNT module does not generate or receive any PRS events.

22.3.7 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.

22.3.7.1 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.

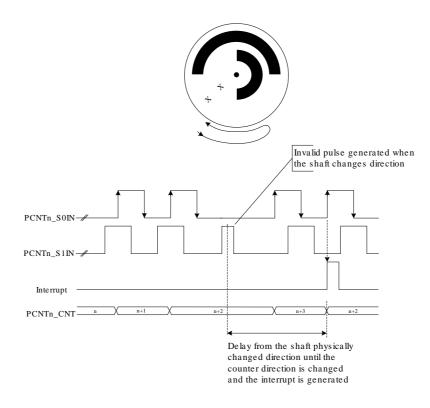
The overflow interrupt flag (OF) is set when the counter counts up from the PCNTn_TOP (reload) value. I.e. if PCNTn_CNT = PCNTn_TOP and a new pulse is received. The PCNTn_CNT register is loaded with the value 0 after this event.

22.3.7.2 Direction Change Interrupt

The PCNTn_PCNT module sets the DIRCNG interrupt flag (PCNTn_IF register) when the direction of the quadrature code changes. The behavior of this interrupt is illustrated by Figure 22.3 (p. 321).



Figure 22.3. PCNT Direction Change Interrupt (DIRCNG) Generation





22.4 Register Map

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

Offset	Name	Туре	Description
0x000	PCNTn_CTRL	RW	Control Register
0x004	PCNTn_CMD	W1	Command Register
0x008	PCNTn_STATUS	R	Status Register
0x00C	PCNTn_CNT	R	Counter Value Register
0x010	PCNTn_TOP	R	Top Value Register
0x014	PCNTn_TOPB	RW	Top Value Buffer Register
0x018	PCNTn_IF	R	Interrupt Flag Register
0x01C	PCNTn_IFS	W1	Interrupt Flag Set Register
0x020	PCNTn_IFC	W1	Interrupt Flag Clear Register
0x024	PCNTn_IEN	RW	Interrupt Enable Register
0x028	PCNTn_ROUTE	RW	I/O Routing Register
0x02C	PCNTn_FREEZE	RW	Freeze Register
0x030	PCNTn_SYNCBUSY	R	Synchronization Busy Register

22.5 Register Description

22.5.1 PCNTn_CTRL - Control Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

Offset															Bi	t Pc	siti	on														
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	8	7	9	2	4	ю	2	-	0
Reset				•							•		•								•		,				0	0	0	0	3	OX O
Access																											RW	RW W	W.	RW	3	—— ≩
Name																											RSTEN	FILT	EDGE	CNTDIR	LI C	<u></u>

Bit	Name	Reset	Access	Description
31:6	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
5	RSTEN	0	RW	Enable PCNT Clock Domain Reset
		bit is cleared. If external		nen this bit is set. The reset is synchronously released two PCNT clock set should be performed by setting and clearing the bit without pending
4	FILT	0	RW	Enable Digital Pulse Width Filter
	The filter passe	s all high and low periods	that are at least 5	clock cycles long. This filter is only available in OVSSINGLE mode.
3	EDGE	s all high and low periods 0	that are at least 5	clock cycles long. This filter is only available in OVSSINGLE mode. Edge Select
3	EDGE Determines the	0	RW dges. This bit sho	Edge Select uld be written when PCNT is in DISABLE mode, otherwise the behavior
3	EDGE Determines the	0 polarity of the incoming e	RW dges. This bit sho TCLKSINGLE mod	Edge Select uld be written when PCNT is in DISABLE mode, otherwise the behavior
3	EDGE Determines the is unpredictable	0 polarity of the incoming e This bit is ignored in EX	RW dges. This bit sho TCLKSINGLE mod	Edge Select uld be written when PCNT is in DISABLE mode, otherwise the behavior de.



Bit	Name	Reset	Access	Description
2	CNTDIR	0	RW	Non-Quadrature Mode Counter Direction Control
		f the counter must be set in a sautomatically detected.	the OVSSINGLE	and EXTCLKSINGLE modes. This bit is ignored in EXTCLKQUAD mode
	Value	Mode	Des	scription
	0	UP	Up	counter mode.
	1	DOWN	Dov	wn counter mode.
1:0	MODE	0x0	RW	Mode Select
	Selects the mo	de of operation. The corres	ponding clock so	ource must be selected from the CMU.
	Value	Mode	Des	scription
	0	DISABLE	The	e module is disabled.
	1	OVSSINGLE	Sin	gle input LFACLK oversampling mode (available in EM0-EM2).
	2	EXTCLKSINGLE	Ext	ernally clocked single input counter mode (available in EM0-EM3).
	3	EXTCLKQUAD	Ext	ernally clocked quadrature decoder mode (available in EM0-EM3).

22.5.2 PCNTn_CMD - Command Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19) .

Offset								,							Bi	t Po	siti	on						,								
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	8	7	9	2	4	8	2	-	0
Reset				,						•	•												,								0	0
Access																															M1	M
Name																															LTOPBIM	LCNTIM

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	LTOPBIM	0	W1	Load TOPB Immediately
				ed directly into TOP. For EFM32G revisions A and B: Load PCNTn_TOPB escription on how to extract the chip revision.
0	LCNTIM	0	W1	Load CNT Immediately
	Load PCNTn_TOF	into PCNTn_CNT on t	he next counter cl	lock cycle.

22.5.3 PCNTn_STATUS - Status Register

Offset															Bi	t Pc	siti	on														
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	တ	8	7	9	2	4	က	2	-	0
Reset			•																		•				•				•			0
Access																																~
Name																																DIR

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure con	npatibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)



Bit	Name	Reset	Access	Description
0	DIR	0	R	Current Counter Direction
	Current direction	n status of the counter. This	s bit is valid in EX	TCLKQUAD mode only.
	Value	Mode	Desc	pription
	0	UP		counter mode (clockwise in EXTCLKQUAD mode with the NEDGE bit in Tn_CTRL set to 0).
	1	DOWN	Dow	n counter mode.

22.5.4 PCNTn_CNT - Counter Value Register

Offset															Bi	t Pc	siti	on														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	9	17	16	15	4	13	12	=	10	6	∞	7	9	2	4	က	2	-	0
Reset																								0000	000000							
Access																								٥	צ							
Name																								Ė	<u>-</u>							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure compa	atibility with futu	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	CNT	0x0000	R	Counter Value
	Gives read access to the co	ounter.		

22.5.5 PCNTn_TOP - Top Value Register

Offset															Bi	t Po	siti	on														
0x010	31	99	53	78	27	26	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	80	7	9	2	4	က	2	-	0
Reset																									OXOOFF							
Access																								٥	צ							
Name																								C F	2							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)		
15:0	TOP	0x00FF	R	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.			

22.5.6 PCNTn_TOPB - Top Value Buffer Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19) .



Offset					,										Bi	t Po	siti	on														
0x014	33	30	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	∞	7	9	2	4	က	2	-	0
Reset																								L	0X00FF							
Access																								2	≥ Y							
Name																								C	IOPB							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	ТОРВ	0x00FF	RW	Counter Top Buffer
	,			evisions A and B: Loaded into TOP when LTOPBIM in PCNTn_CMD oftion on how to extract the chip revision.

22.5.7 PCNTn_IF - Interrupt Flag Register

Offset															Bi	t Pc	siti	on														
0x018	31	98	59	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	11	10	6	ω	7	9	2	4	3	2	-	0
Reset													·	,											,					0	0	0
Access																														2	œ	~
Name																														DIRCNG	OF	UF

Bit	Name	Reset	Access	Description
31:3	Reserved	To ensure con	npatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	DIRCNG	0	R	Direction Change Detect Interrupt Flag
	Set when the count direction	on changes. Set i	n EXTCLKQUA	D mode only.
1	OF	0	R	Overflow Interrupt Read Flag
	Set when a CNT overflow	occurs		
0	UF	0	R	Underflow Interrupt Read Flag
	Set when a CNT underflow	occurs		

22.5.8 PCNTn_IFS - Interrupt Flag Set Register

Offset															Bi	t Pc	siti	on														
0x01C	33	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	œ	7	9	2	4	ю	2	-	0
Reset															,								•						,	0	0	0
Access																														W1	W	M
Name																														DIRCNG	OF	UF



Bit	Name	Reset	Access	Description
31:3	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	DIRCNG	0	W1	Direction Change Detect Interrupt Set
	Write to 1 to set the	e direction change inter	rrupt flag	
1	OF	0	W1	Overflow Interrupt Set
	Write to 1 to set the	e overflow interrupt flag	3	
0	UF	0	W1	Underflow interrupt set
	Write to 1 to set the	underflow interrupt fla	ag	

22.5.9 PCNTn_IFC - Interrupt Flag Clear Register

Offset															Bi	t Pc	siti	on														
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	œ	7	9	2	4	က	2	-	0
Reset					•			•								•					•									0	0	0
Access																														W1	W	W N
Name																														DIRCNG	P	UF

Bit	Name	Reset	Access	Description
31:3	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	DIRCNG	0	W1	Direction Change Detect Interrupt Clear
	Write to 1 to clear t	he direction change de	tect interrupt flag	
1	OF	0	W1	Overflow Interrupt Clear
	Write to 1 to clear t	he overflow interrupt fla	ag	
0	UF	0	W1	Underflow Interrupt Clear
	Write to 1 to clear t	he underflow interrupt	flag	

22.5.10 PCNTn_IEN - Interrupt Enable Register

Offset															Bi	t Po	siti	on														
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	ω	7	9	2	4	က	2	-	0
Reset																														0	0	0
Access																														RW	RW	R
Name																														DIRCNG	OF	UF

Bit	Name	Reset	Access	Description
31:3	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	DIRCNG	0	RW	Direction Change Detect Interrupt Enable
	Enable the direction chang	e detect interrupt.		
1	OF	0	RW	Overflow Interrupt Enable



Bit	Name	Reset	Access	Description	
	Enable the overflow in	nterrupt			
0	UF	0	RW	Underflow Interrupt Enable	
	Enable the underflow	interrupt			

22.5.11 PCNTn_ROUTE - I/O Routing Register

Offset															Bi	t Po	siti	on														
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	=	10	6	8	7	9	2	4	ю	2	1	0
Reset																							2	Š								
Access																							, A	2								
Name																							F									

Bit	Name	Reset	Access	Description
31:10	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
9:8	LOCATION	0x0	RW	I/O Location
	Defines the loc	ation of the PCNT input pin	is. E.g. PCNTn_S	0#0, #1 or #2.
	Value	Mode	Des	cription
	0	LOC0	Loca	ation 0
	1	LOC1	Loca	ation 1
	2	LOC2	Loca	ation 2
7:0	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)

22.5.12 PCNTn_FREEZE - Freeze Register

Offset															Bi	t Po	siti	on														
0x02C	33	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	9	6	∞	7	9	2	4	ю	7	-	0
Reset								•	•								•				•			•	•							0
Access																																R W
Name																																REGFREEZE

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	REGFREEZE	0	RW	Register Update Freeze
	When set, the up simultaneously.	date of the PCNT clos	ck domain is postp	poned until this bit is cleared. Use this bit to update several registers
	Value	Mode	Des	cription
	0	UPDATE		h write access to a PCNT register is updated into the Low Frequency domain as n as possible.
				n do possible.



22.5.13 PCNTn_SYNCBUSY - Synchronization Busy Register

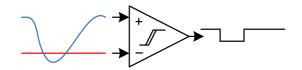
Offset															Bi	it Po	siti	on														
0x030	31	98	53	28	27	26	22	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	ω	7	9	2	4	က	7	_	0
Reset				,						•	•	•						•			•		,							0	0	0
Access																													-	~	ď	~
Name																														TOPB	CMD	CTRL

Bit	Name	Reset	Access	Description
31:3	Reserved	To ensure c	ompatibility with fut	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	ТОРВ	0	R	TOPB Register Busy
	Set when the value writ	ten to TOPB is be	ing synchronized.	
1	CMD	0	R	CMD Register Busy
	Set when the value writ	ten to CMD is beir	ng synchronized.	
0	CTRL	0	R	CTRL Register Busy
	Set when the value writ	ten to CTRL is bei	ng synchronized.	



23 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 and using as little as 100 nA, 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.

23.1 Introduction

The Analog Comparator is used to compare the voltage of two analog inputs, with a digital output indicating which input voltage is higher. Inputs can either be one of the selectable internal references or from external pins. Response time and thereby also the current consumption can be configured by altering the current supply to the comparator.

23.2 Features

- 8 selectable external positive inputs
- 8 selectable external negative inputs
- · 3 selectable internal negative inputs
 - Internal 1.25 V bandgap
 - Internal 2.5 V bandgap
 - V_{DD} scaled by 64 selectable factors
- Low power mode for internal V_{DD} and bandgap references
- Selectable hysteresis
 - 8 levels between 0 and ±70 mV
- Selectable response time
- Asynchronous interrupt generation on selectable edges
 - Rising edge
 - · Falling edge
 - Both edges
- Operational in EM0-EM3
- Dedicated capacitive sense mode with up to 8 inputs
 - Adjustable internal resistor
- Configurable inversion of comparator output
- Configurable output when inactive

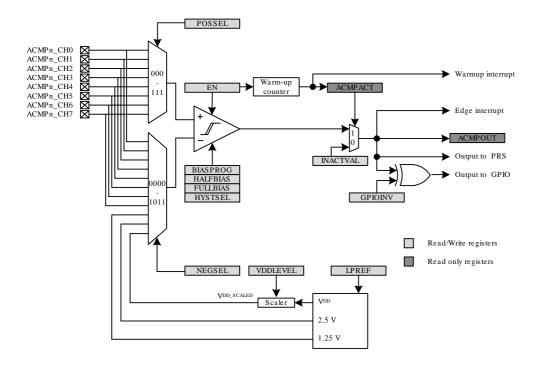


- · Comparator output direct on PRS
- · Comparator output on GPIO through alternate functionality
 - · Output inversion available

23.3 Functional Description

An overview of the ACMP is shown in Figure 23.1 (p. 330).

Figure 23.1. 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.

The output of the comparator can be read in the ACMPOUT bit in ACMPn_STATUS. It is possible to switch inputs while the comparator is enabled, but all other configuration should only be changed while the comparator is disabled.

23.3.1 Warm-up Time

The analog comparator is enabled by setting the EN bit in ACMPn_CTRL. When this bit is set, the comparator must stabilize before becoming active and the outputs can be used. This time period is called the warm-up time. The warm-up time is a configurable number of peripheral clock (HFPERCLK) cycles, set in WARMTIME, which should be set to at least 10 µs but lengthens to up to 1ms if LPREF is enabled. The ACMP should always start in active mode and then enable the LPREF after warm-up time. When the comparator is enabled and warmed up, the ACMPACT bit in ACMPn_STATUS will indicate that the comparator is active. The output value when the comparator is inactive is set to the value in INACTVAL in ACMPn_CTRL (see Figure 23.1 (p. 330)).

An edge interrupt will be generated after the warm-up time if edge interrupt is enabled and the value set in INACTVAL is different from ACMPOUT after warm-up.

One should wait until the warm-up period is over before entering EM2 or EM3, otherwise no comparator interrupts will be detected. EM1 can still be entered during warm-up. After the warm-up period is completed, interrupts will be detected in EM2 and EM3.



23.3.2 Response Time

There is a delay from when the actual input voltage changes polarity, to when the output toggles. This period is called the response time and can be altered by increasing or decreasing the bias current to the comparator through the BIASPROG, FULLBIASPROG and HALFBIAS fields in the ACMPn_CTRL register, as illustrated in Table 23.1 (p. 331) Setting the HALFBIAS bit in ACMPn_CTRL effectively halves the current. Setting a lower bias current will result in lower power consumption, but a longer response time.

If the FULLBIAS bit is set, the highest hysteresis level should be used to avoid glitches on the output.

Table 23.1. Bias Configuration

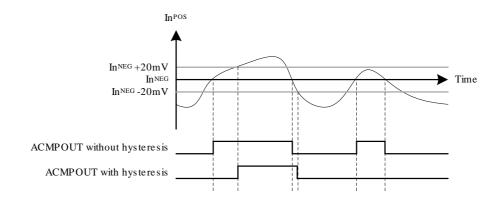
BIASPROG		Bias Current (µ	A), HYSTSEL=0	
	FULLBIAS=0, HALFBIAS=1	FULLBIAS=0, HALFBIAS=0	FULLBIAS=1, HALFBIAS=1	FULLBIAS=1, HALFBIAS=0
0b0000	0.05	0.1	3.3	6.5
0b0001	0.1	0.2	6.5	13
0b0010	0.2	0.4	13	26
0b0011	0.3	0.6	20	39
0b0100	0.4	0.8	26	52
0b0101	0.5	1.0	33	65
0b0110	0.6	1.2	39	78
0b0111	0.7	1.4	46	91
0b1000	1.0	2.0	65	130
0b1001	1.1	2.2	72	143
0b1010	1.2	2.4	78	156
0b1011	1.3	2.6	85	169
0b1100	1.4	2.8	91	182
0b1101	1.5	3.0	98	195
0b1110	1.6	3.2	104	208
0b1111	1.7	3.4	111	221

23.3.3 Hysteresis

In the analog comparator, hysteresis can be configured to 8 different levels, including off which is level 0, through the HYSTSEL field in ACMPn_CTRL. When the hysteresis level is set above 0, 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 23.2 (p. 332)). This feature can be used to filter out uninteresting input fluctuations around zero and only show changes that are big enough to breach the hysteresis threshold. Note that the ACMP current consumption will be influenced by the selected hysteresis level and in general decrease with increasing HYSTSEL values.



Figure 23.2. 20 mV Hysteresis Selected



23.3.4 Input Selection

The POSSEL and NEGSEL fields in ACMPn_INPUTSEL controls which signals are connected to the two inputs of the comparator. 8 external pins are available for both the negative and positive input. For the negative input, 3 additional internal reference sources are available; 1.25 V bandgap, 2.5V bandgap and V_{DD} . The V_{DD} reference can be scaled by a configurable factor, which is set in VDDLEVEL (in ACMPn_INPUTSEL) according to the following formula:

$$V_{DD}$$
 Scaled
$$V_{DD_SCALED} = V_{DD} \times VDDLEVEL/63 \tag{23.1}$$

A low power reference mode can be enabled by setting the LPREF bit in ACMPn_INPUTSEL. In this mode, the power consumption in the reference buffer (V_{DD} and bandgap) is lowered at the cost of accuracy. Low power mode will only save power if V_{DD} with VDDLEVEL higher than 0 or a bandgap reference is selected.

Normally the analog comparator input mux is disabled when the EN (in ACMPn_CTRL) bit is set low. However if the MUXEN bit in ACMPn_CTRL is set, the mux is enabled regardless of the EN bit. This will minimize kickback noise on the mux inputs when the EN bit is toggled.

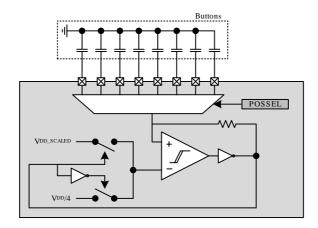
23.3.5 Capacitive Sense Mode

The analog comparator includes specialized hardware for capacitive sensing of passive push buttons. Such buttons are traces on PCB laid out in a way that creates a parasitic capacitor between the button and the ground node. Because a human finger will have a small intrinsic capacitance to ground, the capacitance of the button will increase when the button is touched. The capacitance is measured by including the capacitor in a free-running RC oscillator (see Figure 23.3 (p. 333)). The frequency produced will decrease when the button is touched compared to when it is not touched. By measuring the output frequency with a timer (e.g. through PRS), the change in capacitance can be calculated.

The analog comparator contains a complete feedback loop including an optional internal resistor. This resistor is enabled by setting the CSRESEN bit in ACMPn_INPUTSEL. The resistance can be set to one of four values by configuring the CSRESSEL bits in ACMPn_INPUTSEL. If the internal resistor is not enabled, the circuit will be open. The capacitive sense mode is enabled by setting the NEGSEL field in ACMPn_INPUTSEL to CAPSENSE. The input pin is selected through the POSSEL bits in ACMPn_INPUTSEL. The scaled V_{DD} in Figure 23.3 (p. 333) can be altered by configuring the VDDLEVEL in ACMPn_INPUTSEL. It is recommended to set the hysteresis (HYSTSEL in ACMPn_CTRL) higher than the lowest level when using the analog comparator in capacitive sense mode.



Figure 23.3. Capacitive Sensing Set-up



23.3.6 Interrupts and PRS Output

The analog comparator includes an edge triggered interrupt flag (EDGE in ACMPn_IF). If either IRISE and/or IFALL in ACMPn_CTRL is set, the EDGE interrupt flag will be set on rising and/or falling edge of the comparator output, respectively. An interrupt request will be sent if the EDGE interrupt flag in ACMPn_IF is set and enabled through the EDGE bit in ACMPn_IEN. The edge interrupt can also be used to wake up the device from EM3-EM1.

The analog comparator also includes an interrupt flag, WARMUP in ACMPn_IF, which is set when a warm-up sequence has finished. An interrupt request will be sent if the WARMUP interrupt flag in ACMPn_IF is set and enabled through the WARMUP bit in ACMPn_IEN.

The comparator output is also available as a PRS signal.

23.3.7 Output to GPIO

The output from the comparator is available as alternate function to the GPIO pins. Set the ACMPPEN bit in ACMPn_ROUTE to enable output to pin, and the LOCATION bits to select output location. The GPIO-pin must also be set as output. The output to the GPIO can be inverted by setting the GPIOINV bit in ACMPn_CTRL.



23.4 Register Map

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

Offset	Name	Туре	Description
0x000	ACMPn_CTRL	RW	Control Register
0x004	ACMPn_INPUTSEL	RW	Input Selection Register
0x008	ACMPn_STATUS	R	Status Register
0x00C	ACMPn_IEN	RW	Interrupt Enable Register
0x010	ACMPn_IF	R	Interrupt Flag Register
0x014	ACMPn_IFS	W1	Interrupt Flag Set Register
0x018	ACMPn_IFC	W1	Interrupt Flag Clear Register
0x01C	ACMPn_ROUTE	RW	I/O Routing Register

23.5 Register Description

23.5.1 ACMPn_CTRL - Control Register

Offset														Bi	t Po	siti	on														
0x000	33	30	29	28	27	26	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	3	2	-	0
Reset	0	-		•		0x7								0	0							0x0				0x0		0	0	0	0
Access	R W	W.				RW								W.	RW		-					RW				R W		RW	RW	RW	RW
Name	FULLBIAS	HALFBIAS				BIASPROG								IFALL	IRISE							WARMTIME				HYSTSEL		GPIOINV	INACTVAL	MUXEN	Z

Bit	Name	Reset	Access	Description
31	FULLBIAS	0	RW	Full Bias Current
	Set this bit to 1	for full bias current in acco	ordance with Tab	le 23.1 (p. 331) .
30	HALFBIAS	1	RW	Half Bias Current
	Set this bit to 1	to halve the bias current in	n accordance with	h Table 23.1 (p. 331) .
29:28	Reserved	To ensure co	ompatibility with f	uture devices, always write bits to 0. More information in Section 2.1 (p. 3,
27:24	BIASPROG	0x7	RW	Bias Configuration
	These bits cont	rol the bias current level ir	accordance with	n Table 23.1 (p. 331) .
23:18	Reserved	To ensure co	ompatibility with f	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
17	IFALL	0	RW	Falling Edge Interrupt Sense
	Set this bit to 1	to set the EDGE interrupt	flag on falling ed	ges of comparator output.
	Set this bit to 1	to set the EDGE interrupt Mode		ges of comparator output.
		·	De	
	Value	Mode	De Inte	scription
16	Value	Mode DISABLED	De Inte	scription errupt flag is not set on falling edges.
16	Value 0 1 IRISE	Mode DISABLED ENABLED	De Inte	scription errupt flag is not set on falling edges. errupt flag is set on falling edges. Rising Edge Interrupt Sense
16	Value 0 1 IRISE	Mode DISABLED ENABLED	Poe Into Into RW flag on rising edg	scription errupt flag is not set on falling edges. errupt flag is set on falling edges. Rising Edge Interrupt Sense
16	Value 0 1 IRISE Set this bit to 1	Mode DISABLED ENABLED 0 to set the EDGE interrupt	Poe International Internationa	scription errupt flag is not set on falling edges. errupt flag is set on falling edges. Rising Edge Interrupt Sense ges of comparator output.



Bit	Name	Reset	Access	Description
10:8	WARMTIME	0x0	RW	Warm-up Time
	Set analog com	parator warm-up time.		
	Value	Mode		Description
	0	4CYCLES		HFPERCLK cycles.
	1	8CYCLES	8	HFPERCLK cycles.
	2	16CYCLES	1	6 HFPERCLK cycles.
	3	32CYCLES	3	32 HFPERCLK cycles.
	4	64CYCLES	6	4 HFPERCLK cycles.
	5	128CYCLES	1	28 HFPERCLK cycles.
	6	256CYCLES	2	56 HFPERCLK cycles.
	7	512CYCLES	5	i12 HFPERCLK cycles.
7	Reserved	To ensure co	mpatibility with	n future devices, always write bits to 0. More information in Section 2.1 (p. 3
6:4	HYSTSEL	0x0	RW	Hysteresis Select
	Select hysteresi	s level. The hysteresis lev	els can vary, p	lease see the electrical characteristics for the device for more information
	Value	Mode		Description
	0	HYST0	N	No hysteresis.
	1	HYST1	-	15 mV hysteresis.
	2	HYST2	~	-22 mV hysteresis.
	3	HYST3	~	-29 mV hysteresis.
	4	HYST4	~	36 mV hysteresis.
	5	HYST5	-	43 mV hysteresis.
	6	HYST6	-	50 mV hysteresis.
	7	HYST7	~	-57 mV hysteresis.
3	GPIOINV	0	RW	Comparator GPIO Output Invert
	Set this bit to 1	to invert the comparator al	ternate function	n output to GPIO.
	Value	Mode		Description
	0	NOTINV	Т	The comparator output to GPIO is not inverted.
	1	INV	Т	The comparator output to GPIO is inverted.
2	INACTVAL	0	RW	Inactive Value
	The value of this	s bit is used as the compar	rator output wh	en the comparator is inactive.
	Value	Mode	С	Description
	0	LOW	Т	he inactive value is 0.
	1	HIGH	Т	The inactive state is 1.
1	MUXEN	0	RW	Input Mux Enable
	Enable Input Mu	ux. Setting the EN bit will a	lso enable the	input mux.
)	EN	0	RW	Analog Comparator Enable

23.5.2 ACMPn_INPUTSEL - Input Selection Register

Offset									,				·		Bi	t Po	ositi	on							,							
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	1	10	6	∞	7	9	2	4	က	2	1	0
Reset			3	OXO				0								-						0x0				ç	8 0 0				0x0	
Access			Š	≩				RW								RW						RW				2	≩				RW	
Name			0000	COREODEL				CSRESEN								LPREF						VDDLEVEL				L C C	NEGSEL				POSSEL	



Bit	Name	Reset	Acces	s Description
31:30	Reserved	To ensure co	ompatibility wit	th future devices, always write bits to 0. More information in Section 2.1 (p. 3)
29:28	CSRESSEL	0x0	RW	Capacitive Sense Mode Internal Resistor Select
	These bits sele device datashe		or the internal	capacitive sense resistor. Resulting actual resistor values are given in the
	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.
27:25	Reserved	To ensure co	ompatibility wit	th future devices, always write bits to 0. More information in Section 2.1 (p. 3)
24	CSRESEN	0	RW	Capacitive Sense Mode Internal Resistor Enable
	Enable/disable	the internal capacitive sen	se resistor.	·
23:17	Reserved			th future devices always write hits to 0. More information in Section 2.1 (n. 2)
				th future devices, always write bits to 0. More information in Section 2.1 (p. 3)
16	LPREF	1	RW	Low Power Reference Mode
	Enable low pov	wer mode for VDD and ban	idgap referenc	ees.
	Value	Description		
	0	Low power mode	e disabled.	
	1	Low power mode	enabled.	
15:14	Reserved	To ensure co	ompatibility wit	th future devices, always write bits to 0. More information in Section 2.1 (p. 3)
13:8	VDDLEVEL	0x00	RW	VDD Reference Level
	Select scaling f	factor for VDD reference le	vel.Vpp scale	$c_D = V_{DD} \times VDDLE \times VEL/63$.
7:4	NEGSEL	0x8	RW	Negative Input Select
7.4	Select negative		IXVV	regative input delect
	Value	Mode		Description
	0	CH0		Channel 0 as negative input.
	1	CH1		Channel 1 as negative input.
	2	CH2		Channel 2 as negative input.
	3	CH3		Channel 3 as negative input.
	4	CH4		Channel 4 as negative input.
	5	CH4 CH5		Channel 4 as negative input. Channel 5 as negative input.
				Channel 5 as negative input. Channel 6 as negative input.
	5	CH5		Channel 5 as negative input. Channel 6 as negative input. Channel 7 as negative input.
	5 6 7 8	CH5 CH6 CH7 1V25		Channel 5 as negative input. Channel 6 as negative input. Channel 7 as negative input. 1.25 V as negative input.
	5 6 7 8 9	CH5 CH6 CH7 1V25 2V5		Channel 5 as negative input. Channel 6 as negative input. Channel 7 as negative input. 1.25 V as negative input. 2.5 V as negative input.
	5 6 7 8 9 10	CH5 CH6 CH7 1V25 2V5 VDD		Channel 5 as negative input. Channel 6 as negative input. Channel 7 as negative input. 1.25 V as negative input. 2.5 V as negative input. Scaled VDD as negative input.
	5 6 7 8 9	CH5 CH6 CH7 1V25 2V5		Channel 5 as negative input. Channel 6 as negative input. Channel 7 as negative input. 1.25 V as negative input. 2.5 V as negative input.
3	5 6 7 8 9 10	CH5 CH6 CH7 1V25 2V5 VDD CAPSENSE		Channel 5 as negative input. Channel 6 as negative input. Channel 7 as negative input. 1.25 V as negative input. 2.5 V as negative input. Scaled VDD as negative input. Capacitive sense mode.
3 2:0	5 6 7 8 9 10	CH5 CH6 CH7 1V25 2V5 VDD CAPSENSE		Channel 5 as negative input. Channel 6 as negative input. Channel 7 as negative input. 1.25 V as negative input. 2.5 V as negative input. Scaled VDD as negative input. Capacitive sense mode.
	5 6 7 8 9 10 11 Reserved	CH5 CH6 CH7 1V25 2V5 VDD CAPSENSE To ensure co	ompatibility wit	Channel 5 as negative input. Channel 6 as negative input. Channel 7 as negative input. 1.25 V as negative input. 2.5 V as negative input. Scaled VDD as negative input. Capacitive sense mode. th future devices, always write bits to 0. More information in Section 2.1 (p. 3)
	5 6 7 8 9 10 11 Reserved POSSEL Select positive	CH5 CH6 CH7 1V25 2V5 VDD CAPSENSE To ensure co	ompatibility wit	Channel 5 as negative input. Channel 6 as negative input. Channel 7 as negative input. 1.25 V as negative input. 2.5 V as negative input. Scaled VDD as negative input. Capacitive sense mode. th future devices, always write bits to 0. More information in Section 2.1 (p. 3) Positive Input Select
	5 6 7 8 9 10 11 Reserved	CH5 CH6 CH7 1V25 2V5 VDD CAPSENSE To ensure co	ompatibility wit	Channel 5 as negative input. Channel 6 as negative input. Channel 7 as negative input. 1.25 V as negative input. 2.5 V as negative input. Scaled VDD as negative input. Capacitive sense mode. th future devices, always write bits to 0. More information in Section 2.1 (p. 3) Positive Input Select
	5 6 7 8 9 10 11 Reserved POSSEL Select positive Value	CH5 CH6 CH7 1V25 2V5 VDD CAPSENSE To ensure coolinput. Mode	ompatibility wit	Channel 5 as negative input. Channel 6 as negative input. Channel 7 as negative input. 1.25 V as negative input. 2.5 V as negative input. Scaled VDD as negative input. Capacitive sense mode. th future devices, always write bits to 0. More information in Section 2.1 (p. 3) Positive Input Select Description Channel 0 as positive input.
	5 6 7 8 9 10 11 Reserved POSSEL Select positive Value 0 1	CH5 CH6 CH7 1V25 2V5 VDD CAPSENSE To ensure co 0x0 input. Mode CH0	ompatibility wit	Channel 5 as negative input. Channel 6 as negative input. Channel 7 as negative input. 1.25 V as negative input. 2.5 V as negative input. Scaled VDD as negative input. Capacitive sense mode. th future devices, always write bits to 0. More information in Section 2.1 (p. 3) Positive Input Select Description Channel 0 as positive input. Channel 1 as positive input.
	5 6 7 8 9 10 11 Reserved POSSEL Select positive Value 0	CH5 CH6 CH7 1V25 2V5 VDD CAPSENSE To ensure co 0x0 input. Mode CH0 CH1	ompatibility wit	Channel 5 as negative input. Channel 6 as negative input. Channel 7 as negative input. 1.25 V as negative input. 2.5 V as negative input. Scaled VDD as negative input. Capacitive sense mode. th future devices, always write bits to 0. More information in Section 2.1 (p. 3) Positive Input Select Description Channel 0 as positive input. Channel 1 as positive input. Channel 2 as positive input.
	5 6 7 8 9 10 11 Reserved POSSEL Select positive Value 0 1 2	CH5 CH6 CH7 1V25 2V5 VDD CAPSENSE To ensure co 0x0 input. Mode CH0 CH1 CH2	ompatibility wit	Channel 5 as negative input. Channel 6 as negative input. Channel 7 as negative input. 1.25 V as negative input. 2.5 V as negative input. Scaled VDD as negative input. Capacitive sense mode. th future devices, always write bits to 0. More information in Section 2.1 (p. 3) Positive Input Select Description Channel 0 as positive input. Channel 1 as positive input.
	5 6 7 8 9 10 11 Reserved POSSEL Select positive Value 0 1 2 3	CH5 CH6 CH7 1V25 2V5 VDD CAPSENSE To ensure co 0x0 input. Mode CH0 CH1 CH2 CH3	ompatibility wit	Channel 5 as negative input. Channel 6 as negative input. Channel 7 as negative input. 1.25 V as negative input. 2.5 V as negative input. Scaled VDD as negative input. Capacitive sense mode. th future devices, always write bits to 0. More information in Section 2.1 (p. 3) Positive Input Select Description Channel 0 as positive input. Channel 1 as positive input. Channel 2 as positive input. Channel 3 as positive input.
	5 6 7 8 9 10 11 Reserved POSSEL Select positive Value 0 1 2 3 4	CH5 CH6 CH7 1V25 2V5 VDD CAPSENSE To ensure co 0x0 input. Mode CH0 CH1 CH2 CH3 CH4	ompatibility wit	Channel 5 as negative input. Channel 6 as negative input. Channel 7 as negative input. 1.25 V as negative input. 2.5 V as negative input. Scaled VDD as negative input. Capacitive sense mode. th future devices, always write bits to 0. More information in Section 2.1 (p. 3) Positive Input Select Description Channel 0 as positive input. Channel 1 as positive input. Channel 2 as positive input. Channel 3 as positive input. Channel 4 as positive input.



23.5.3 ACMPn_STATUS - Status Register

Offset															Bi	t Po	siti	on														
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	11	10	6	80	7	9	2	4	3	2	-	0
Reset				•																			•		•						0	0
Access																															œ	~
Name																															ACMPOUT	ACMPACT

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	ACMPOUT	0	R	Analog Comparator Output
	Analog comparator ou	tput value.		
0	ACMPACT	0	R	Analog Comparator Active
	Analog comparator ac	tive status.		

23.5.4 ACMPn_IEN - Interrupt Enable Register

Offset															Bi	t Pc	siti	on														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	11	10	6	8	7	9	2	4	8	2	-	0
Reset																			•		-										0	0
Access																															RW	RW
Name																															WARMUP	EDGE

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure comp	atibility with fut	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	WARMUP	0	RW	Warm-up Interrupt Enable
	Enable/disable interrupt or	n finished warm-up		
0	EDGE	0	RW	Edge Trigger Interrupt Enable
	Enable/disable edge trigge	ered interrupt.		

23.5.5 ACMPn_IF - Interrupt Flag Register

Offset															Bi	t Pc	siti	on														
0x010	31	30	53	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	8	7	9	2	4	3	2	-	0
Reset									•						•							•		•	•						0	0
Access																															œ	~
Name																															WARMUP	EDGE



Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	WARMUP	0	R	Warm-up Interrupt Flag
	Indicates that the a	nalog comparator warr	n-up period is fini	shed.
0	EDGE	0	R	Edge Triggered Interrupt Flag
	Indicates that there	has been a rising or fa	alling edge on the	analog comparator output.

23.5.6 ACMPn_IFS - Interrupt Flag Set Register

Offset															Bi	t Po	siti	on														
0x014	31	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	က	7	1	0
Reset																															0	0
Access																															W1	W W
Name																															WARMUP	EDGE

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	WARMUP	0	W1	Warm-up Interrupt Flag Set
	Write to 1 to set wa	arm-up finished interrup	t flag.	
0	EDGE	0	W1	Edge Triggered Interrupt Flag Set
	Write to 1 to set ed	lge triggered interrupt fl	ag.	

23.5.7 ACMPn_IFC - Interrupt Flag Clear Register

Offset															Bi	t Pc	siti	on														
0x018	31	30	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	ю	7	1	0
Reset																														,	0	0
Access																															W1	W
Name																															WARMUP	EDGE

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	WARMUP	0	W1	Warm-up Interrupt Flag Clear
	Write to 1 to clear	warm-up finished interre	upt flag.	
0	EDGE	0	W1	Edge Triggered Interrupt Flag Clear
	Write to 1 to clear	edge triggered interrupt	t flag.	



23.5.8 ACMPn_ROUTE - I/O Routing Register

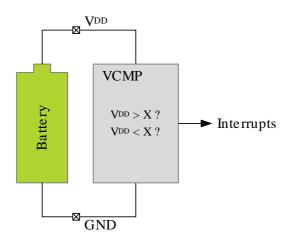
Offset				,	,			,	,			,			Bi	t Pc	siti	on						,	,							
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	9	17	16	15	4	13	12	7	10	6	œ	7	9	2	4	က	7	-	0
Reset		•		•	•				•						•							,	9	8								0
Access																							7	<u>}</u>								RW
Name																							Z C F									ACMPPEN

Bit	Name	Reset	Access	Description
31:10	Reserved	To ensure	compatibility with f	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
9:8	LOCATION	0x0	RW	I/O Location
	Decides the loc	cation of the ACMP I/O pi	n.	
	Value	Mode	De	scription
	0	LOC0	Loc	cation 0
	1	LOC1	Loc	cation 1
7:1	Reserved	To ensure	compatibility with f	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	ACMPPEN	0	RW	ACMP Output Pin Enable
	Enable/disable	analog comparator outpu	ut to pin.	



24 VCMP - Voltage Comparator





Quick Facts

What?

The Voltage Supply Comparator (VCMP) monitors the input voltage supply and generates software interrupts on events using as little as 100 nA.

Why?

The VCMP can be used for simple power supply monitoring, e.g. for a battery level indicator.

How?

The scaled power supply is compared to a programmable reference voltage, and an interrupt can be generated when the supply is higher or lower than the reference. The VCMP can also be duty-cycled by software to further reduce the energy consumption.

24.1 Introduction

The Voltage Supply Comparator is used to monitor the supply voltage from software. An interrupt can be generated when the supply falls below or rises above a programmable threshold.

Note

Note that VCMP comes in addition to the Power-on Reset and Brown-out Detector peripherals, that both generate reset signals when the voltage supply is insufficient for reliable operation. VCMP does not generate reset, only interrupt. Also note that the ADC is capable of sampling the input voltage supply.

24.2 Features

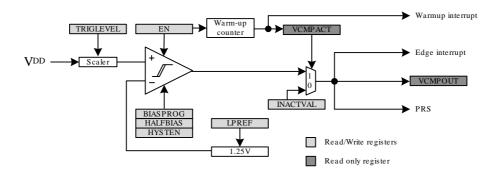
- Voltage supply monitoring
- Scalable V_{DD} in 64 steps selectable as positive comparator input
- Internal 1.25 V bandgap reference
- Low power mode for internal V_{DD} and bandgap references
- Selectable hysteresis
 - 0 or ±20 mV
- Selectable response time
- Asynchronous interrupt generation on selectable edges
 - · Rising edge
 - · Falling edge
 - Rising and Falling edges
- Operational in EM0-EM3
- Comparator output direct on PRS
- · Configurable output when inactive to avoid unwanted interrupts



24.3 Functional Description

An overview of the VCMP is shown in Figure 24.1 (p. 341).

Figure 24.1. VCMP 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 negative input voltage, the digital output is high and vice versa.

The output of the comparator can be read in the VCMPOUT bit in VCMP_STATUS. Configuration registers should only be changed while the comparator is disabled.

24.3.1 Warm-up Time

VCMP is enabled by setting the EN bit in VCMP_CTRL. When this bit is set, the comparator must stabilize before becoming active and the outputs can be used. This time period is called the warm-up time. The warm-up time is a configurable number of HFPERCLK cycles, set in WARMTIME, which should be set to at least 10 µs. When the comparator is enabled and warmed up, the VCMPACT bit in VCMP_STATUS will be set to indicate that the comparator is active.

As long as the comparator is not enabled or not warmed up, VCMPACT will be cleared and the comparator output value is set to the value in INACTVAL in VCMP_CTRL.

One should wait until the warm-up period is over before entering EM2 or EM3, otherwise no comparator interrupts will be detected. EM1 can still be entered during warm-up. After the warm-up period is completed, interrupts will be detected in EM2 and EM3.

24.3.2 Response Time

There is a delay from when the actual input voltage changes polarity, to when the output toggles. This period is called the response time and can be altered by increasing or decreasing the bias current to the comparator through the BIAS and HALFBIAS fields in VCMP_CTRL as shown in Table 24.1 (p. 341). Setting a lower bias current will result in lower power consumption, but a longer response time.

Table 24.1. Bias Configuration

BIAS	Bias Cu	rrent (µA)
	HALFBIAS=0	HALFBIAS=1
0b0000	0.1	0.05
0b0001	0.2	0.1
0b0010	0.4	0.2
0b0011	0.6	0.3

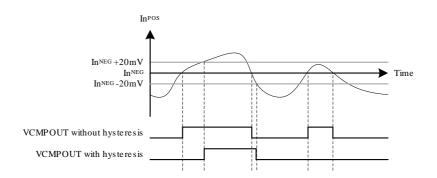


BIAS	Bias Cu	rrent (µA)							
	HALFBIAS=0	HALFBIAS=1							
0b0100	0.8	0.4							
0b0101	1.0	0.5							
0b0110	1.2	0.6							
0b0111	1.4	0.7							
0b1000	2.0	1.0							
0b1001	2.2	1.1							
0b1010	2.4	1.2							
0b1011	2.6	1.3							
0b1100	2.8	1.4							
0b1101	3.0	1.5							
0b1110	3.2	1.6							
0b1111	3.4	1.7							

24.3.3 Hysteresis

In the voltage supply comparator, hysteresis can be enabled by setting HYSTEN in VCMP_CTRL. When HYSTEN is set, the digital output will not toggle until the positive input voltage is at least 20mV above or below the negative input voltage. This feature can be used to filter out uninteresting input fluctuations around zero and only show changes that are big enough to breach the hysteresis threshold.

Figure 24.2. VCMP 20 mV Hysteresis Enabled



24.3.4 Input Selection

The positive comparator input is always connected to the scaled power supply input. The negative comparator input is connected to the internal 1.25 V bandgap reference. The V_{DD} trigger level can be configured by setting the TRIGLEVEL field in VCMP_CTRL according to the following formula:

VCMP
$$V_{DD}$$
 Trigger Level
$$V_{DD \text{ Trigger Level}} = 1.667 \text{V} + 0.034 \text{V} \times \text{TRIGLEVEL}$$
 (24.1)

A low power reference mode can be enabled by setting the LPREF bit in VCMP_INPUTSEL. In this mode, the power consumption in the reference buffer (V_{DD} and bandgap) is lowered at the cost of accuracy.



24.3.5 Interrupts and PRS Output

The VCMP includes an edge triggered interrupt flag (EDGE in VCMP_IF). If either IRISE and/or IFALL in VCMPn_CTRL is set, the EDGE interrupt flag will be set on rising and/or falling edge of the comparator output respectively. An interrupt request will be sent if the EDGE interrupt flag in VCMP_IF is set and enabled through the EDGE bit in VCMPn_IEN. The edge interrupt can also be used to wake up the device from EM3-EM1. VCMP also includes an interrupt flag, WARMUP in VCMP_IF, which is set when a warm-up sequence has finished. An interrupt request will be sent if the WARMUP interrupt flag in VCMP_IF is set and enabled through the WARMUP bit in VCMPn_IEN. The synchronized comparator output is also available as a PRS output signal.



24.4 Register Map

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

Offset	Name	Туре	Description
0x000	VCMP_CTRL	RW	Control Register
0x004	VCMP_INPUTSEL	RW	Input Selection Register
0x008	VCMP_STATUS	R	Status Register
0x00C	VCMP_IEN	RW	Interrupt Enable Register
0x010	VCMP_IF	R	Interrupt Flag Register
0x014	VCMP_IFS	W1	Interrupt Flag Set Register
0x018	VCMP_IFC	W1	Interrupt Flag Clear Register

24.5 Register Description

24.5.1 VCMP_CTRL - Control Register

Offset															Bi	t Po	siti	on														
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	11	10	6	8	7	9	2	4	က	2	-	0
Reset		-				1,	 X0								0	0		•					0x0					0		0		0
Access		W.				2	≷								RW	RW							RW					RW		ΑW		RW
Name		HALFBIAS				ç	BIASPROG								IFALL	IRISE							WARMTIME					HYSTEN		INACTVAL		N N

	HAL	BIAS		A NAME OF THE PART												
		_	_													
Bit	Name	Reset	Access	ss Description												
31	Reserved	To ens	ure compatibility with	ith future devices, always write bits to 0. More information in Section 2.1 (p. 3)												
30	HALFBIAS	1	RW	Half Bias Current												
	Set this bit to 1	to halve the bias cur	rent. Table 24.1 (p. 3	. 341) .												
29:28	Reserved	To ens	ure compatibility witl	ith future devices, always write bits to 0. More information in Section 2.1 (p. 3)												
27:24	BIASPROG	0x7	RW	VCMP Bias Programming Value												
	These bits conti	rol the bias current le	evel. Table 24.1 (p. 3	341).												
23:18	Reserved	To ens	ure compatibility witl	ith future devices, always write bits to 0. More information in Section 2.1 (p. 3)												
17	IFALL	0	RW	Falling Edge Interrupt Sense												
	Set this bit to 1	to set the EDGE inte	rrupt flag on falling e	g edges of comparator output.												
16	IRISE	0	RW	Rising Edge Interrupt Sense												
	Set this bit to 1	to set the EDGE inte	rrupt flag on rising e	edges of comparator output.												
15:11	Reserved	To ens	ure compatibility witl	ith future devices, always write bits to 0. More information in Section 2.1 (p. 3)												
10:8	WARMTIME	0x0	RW	Warm-Up Time												
	Set warm-up tin	ne														
	Value	Mode	1	Description												
	0	4CYCLES	4	4 HFPERCLK cycles												
	1	8CYCLES	8	8 HFPERCLK cycles												
	2	16CYCLES		16 HFPERCLK cycles												
	3	32CYCLES	;	32 HFPERCLK cycles												

64CYCLES

128CYCLES

256CYCLES

64 HFPERCLK cycles

128 HFPERCLK cycles

256 HFPERCLK cycles



2.		_										
Bit	Name	Reset	Access	Description								
	Value	Mode	Des	ecription								
	7	512CYCLES	512	HFPERCLK cycles								
7:5	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)								
4	HYSTEN	0	RW	Hysteresis Enable								
	Enable hystere	sis.										
	Value	Description										
	0	No hysteresis										
	1	+-20 mV hystere	sis									
3	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)								
2	INACTVAL	0	RW	Inactive Value								
	Configure the c	output value when the comp	parator is inactive									
1	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)								
0	EN	0	RW	Voltage Supply Comparator Enable								
	Enable/disable	Enable/disable voltage supply comparator.										

24.5.2 VCMP_INPUTSEL - Input Selection Register

Offset	Bit Position		
0x004	30 30 30 30 30 30 30 30 30 30 30 30 30 3	0 2 9	α
Reset	c		00×0
Access	3	2	RW
Name	ш о	<u>ц</u>	TRIGLEVEL

Bit	Name	Reset	Access	Description
31:9	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
8	LPREF	0	RW	Low Power Reference
		ower mode for VDD a	0 1	rence. When using this bit, always leave it as 0 during warm-up and then
7:6	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
5:0	TRIGLEVEL	0x00	RW	Trigger Level
	Select VDD trigger le	evel. V _{trig} = 1.667V+0	.034V×TRIGLEVE	≣L.

24.5.3 VCMP_STATUS - Status Register

Offset															Bi	t Pc	siti	on														
0x008	31	30	59	28	27	56	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	တ	8	7	9	2	4	က	7	-	0
Reset								•	•			•				•	•								•			•		,	0	0
Access																															œ	~
Name																															VCMPOUT	VCMPACT



Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	VCMPOUT	0	R	Voltage Supply Comparator Output
	Voltage supply com	parator output value		
0	VCMPACT	0	R	Voltage Supply Comparator Active
	Voltage supply com	parator active status.		

24.5.4 VCMP_IEN - Interrupt Enable Register

Offset															Bi	t Po	siti	on														
0x00C	31	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	∞	7	9	2	4	က	2	-	0
Reset																															0	0
Access																															RW	RW W
Name																															WARMUP	EDGE

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	WARMUP	0	RW	Warm-up Interrupt Enable
	Enable/disable inter	rrupt on finished warm	-up.	
0	EDGE	0	RW	Edge Trigger Interrupt Enable
	Enable/disable edge	e triggered interrupt.		

24.5.5 VCMP_IF - Interrupt Flag Register

Offset	Bit Position														
0x010	1 1 <td>-</td> <td>0</td>	-	0												
Reset		0	0												
Access		~	~												
Name		WARMUP	EDGE												

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure co.	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	WARMUP	0	R	Warm-up Interrupt Flag
	Indicates that warm	n-up has finished.		
0	EDGE	0	R	Edge Triggered Interrupt Flag
	Indicates that there	has been a rising and/	or falling edge on	the VCMP output.



24.5.6 VCMP_IFS - Interrupt Flag Set Register

Offset	Bit Position																															
0x014	31	30	59	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	80	7	9	2	4	က	2	-	0
Reset				,						•					,			•	•	•		•									0	0
Access																															W1	W1
Name																															WARMUP	EDGE

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	WARMUP	0	W1	Warm-up Interrupt Flag Set
	Write to 1 to set wa	arm-up finished interrup	t flag	
0	EDGE	0	W1	Edge Triggered Interrupt Flag Set
	Write to 1 to set ed	ge triggered interrupt fl	ag	

24.5.7 VCMP_IFC - Interrupt Flag Clear Register

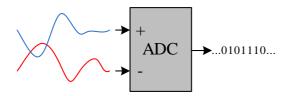
Offset	Bit Position																															
0x018	31	30	29	28	27	56	22	24	23	22	21	20	19	9	17	16	15	4	13	12	=	10	0	8	7	9	2	4	က	7	-	0
Reset																														,	0	0
Access																															W1	W M
Name																															WARMUP	EDGE

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	WARMUP	0	W1	Warm-up Interrupt Flag Clear
	Write to 1 to clear	warm-up finished interro	upt flag	
0	EDGE	0	W1	Edge Triggered Interrupt Flag Clear
	Write to 1 to clear	edge triggered interrupt	flag	



25 ADC - Analog to Digital Converter





Quick Facts

What?

The ADC is used to convert analog signals into a digital representation and features 8 external input channels

Why?

In many applications there is a need to measure analog signals and record them in a digital representation, without exhausting your energy source.

How?

A low power Successive Approximation Register ADC samples up to 8 input channels in a programmable sequence. With the help of PRS and DMA, the ADC can operate without CPU intervention, minimizing the number of powered up resources. The ADC can further be duty-cycled to reduce the energy consumption.

25.1 Introduction

The ADC is a Successive Approximation Register (SAR) architecture, with a resolution of up to 12 bits at up to one million samples per second. The integrated input mux can select inputs from 8 external pins and 6 internal signals.

25.2 Features

- Programmable resolution (6/8/12-bit)
 - 13 prescaled clock (ADC_CLK) cycles per conversion
 - Maximum 1 MSPS @ 12-bit
 - Maximum 1.86 MSPS @ 6-bit
- Configurable acquisition time
- Integrated prescaler
 - Selectable clock division factor from 1 to 128
- 13 MHz to 32 kHz allowed for ADC_CLK
- 18 input channels
 - · 8 external single ended channels
 - 6 internal single ended channels
 - · Including temperature sensor
 - · 4 external differential channels
- Integrated input filter
 - Low pass RC filter
 - Decoupling capacitor
- · Left or right adjusted results
 - Results in 2's complement representation
 - Differential results sign extended to 32-bit results



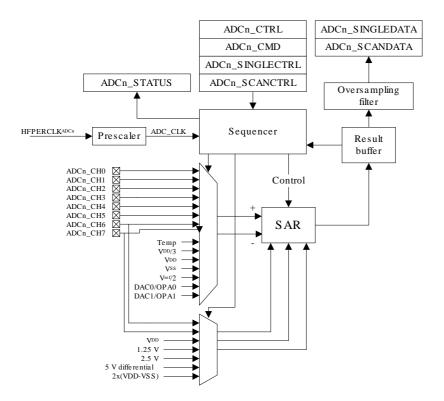
- Programmable scan sequence
 - Up to 8 configurable samples in scan sequence
 - Mask to select which pins are included in the sequence
 - · Triggered by software or PRS input
 - · One shot or repetitive mode
 - Oversampling available
 - · Overflow interrupt flag set when overwriting unread results
 - Conversion tailgating support for predictable periodic scans
- Programmable single conversion
 - Triggered by software or PRS input
 - Can be interleaved between two scan sequences
 - · One shot or repetitive mode
 - · Oversampling available
 - Overflow interrupt flag set when overwriting unread results
- · Hardware oversampling support
 - · 1st order accumulate and dump filter
 - From 2 to 4096 oversampling ratio (OSR)
 - Results in 16-bit representation
 - Enabled individually for scan sequence and single sample mode
 - Common OSR select
- · Individually selectable voltage reference for scan and single mode
 - Internal 1.25V reference
 - Internal 2.5V reference
 - V_{DD}
 - · Internal 5 V differential reference
 - Single ended external reference
 - Differential external reference
 - Unbuffered 2xV_{DD}
- Support for offset and gain calibration
- · Interrupt generation and/or DMA request
 - · Finished single conversion
 - · Finished scan conversion
 - · Single conversion results overflow
 - · Scan sequence results overflow
- Loopback configuration with DAC output measurement

25.3 Functional Description

An overview of the ADC is shown in Figure 25.1 (p. 350).



Figure 25.1. ADC Overview



25.3.1 Clock Selection

The ADC has an internal prescaler (PRESC bits in ADCn_CTRL) which can divide the peripheral clock (HFPERCLK) by any factor between 1 and 128. Note that the resulting ADC_CLK should not be set to a higher frequency than 13 MHz and not lower than 32 kHz.

The BIASPROG bitfield must be set based on the ADC_CLK frequency. See the specific device data sheet for more information on these settings.

25.3.2 Conversions

A conversion consists of two phases. The input is sampled in the acquisition phase before it is converted to digital representation during the approximation phase. The acquisition time can be configured independently for scan and single conversions (see Section 25.3.7 (p. 354)) by setting AT in ADCn_SINGLECTRL/ADCn_SCANCTRL. The acquisition times can be set to any integer power of 2 from 1 to 256 ADC_CLK cycles.

Note

For high impedance sources the acquisition time should be adjusted to allow enough time for the internal sample capacitor to fully charge. The minimum acquisition time for the internal temperature sensor and $V_{dd}/3$ is given in the electrical characteristics for the device.

www.silabs.com

The analog to digital converter core uses one clock cycle per output bit in the approximation phase.

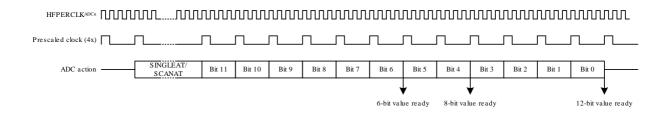
ADC Total Conversion Time (in ADC_CLK cycles) Per Output
$$T_{conv} = (T_A + N) \times OSR$$
(25.1)

T_A equals the number of acquisition cycles and N is the resolution. OSR is the oversampling ratio (see Section 25.3.7.7 (p. 356)). The minimum conversion time is 7 ADC_CYCLES with 6 bit resolution and



13 ADC_CYCLES with 12 bit resolution. The maximum conversion time is 1097728 ADC_CYCLES with the longest acquisition time, 12 bit resolution and highest oversampling rate.

Figure 25.2. ADC Conversion Timing



25.3.3 Warm-up Time

The ADC needs to be warmed up some time before a conversion can take place. This time period is called the warm-up time. When enabling the ADC or changing references between samples, the ADC is automatically warmed up for 1µs and an additional 5 µs if the bandgap is selected as reference.

Normally, the ADC will be warmed up only when samples are requested and is shut off when there are no more samples waiting. However, if lower latency is needed, configuring the WARMUPMODE field in ADCn_CTRL allows the ADC and/or reference to stay warm between samples, eliminating the need for warm-up. Figure 25.3 (p. 352) shows the analog power consumption in scenarios using the different WARMUPMODE settings.

Only the bandgap reference selected for scan mode can be kept warm. If a different bandgap reference is selected for single mode, the warm-up time still applies.

- NORMAL: ADC and references are shut off when there are no samples waiting. a) in Figure 25.3 (p. 352) shows this mode used with an internal bandgap reference. Figure d) shows this mode when using VDD or an external reference.
- FASTBG: Bandgap warm-up is eliminated, but with reduced reference accuracy. d) in Figure 25.3 (p. 352) shows this mode used with an internal bandgap reference.
- KEEPSCANREFWARM: The reference selected for scan mode is kept warm. The ADC will still need to be warmed up before conversion. b) in Figure 25.3 (p. 352) shows this mode used with an internal bandgap reference.
- KEEPADCWARM: The ADC and the reference selected for scan mode is kept warm. c) in Figure 25.3 (p. 352) shows this mode used with an internal bandgap reference.

The minimum warm-up times are given in µs. The timing is done automatically by the ADC, given that a proper time base is given in the TIMEBASE bits in ADCn_CTRL. The TIMEBASE must be set to the number of HFPERCLK which corresponds to at least 1 µs. The TIMEBASE only affects the timing of the warm-up sequence and not the ADC_CLK.

When entering Energy Modes 2 or 3, the ADC must be stopped and WARMUPMODE in ADCn_CTRL written to 0.

(w SCANREF = any)

(w SCANREF = external or VDD)

NORMAL

► Time



Bandgap reference warm-up ADC warm-up ADC conversion ADC enabled Conversion trigger Conversion trigger Power NORMAL a) 5 µs Time 1 μs $1 \mu s$ Power KEEPSCANREFWARM b) (w SCANREF = internal bandgap) 5 µs Time Power KEEPADCWARM c) 5 µs (w SCANREF = internal bandgap) FASTBG

Figure 25.3. ADC Analog Power Consumption With Different WARMUPMODE Settings

25.3.4 Input Selection

Power

d)

The ADC is connected to 8 external input pins, which can be selected as 8 different single ended inputs or 4 differential inputs. In addition, 6 single ended internal inputs can be selected. The available selections are given in the register description for ADCn_SINGLECTRL and ADCn_SCANCTRL.

For offset calibration purposes it is possible to internally short the differential ADC inputs and thereby measure a 0 V differential. Differential 0 V is selected by writing the DIFF bit to 1 and INPUTSEL to 4 in ADCn_SINGLECTRL. Calibration is described in detail in Section 25.3.10 (p. 358) .

Note

When VDD/3 is sampled, the acquisition time should be above a lower limit. The reader is referred to the datasheet for minimum VDD/3 acquisition time.

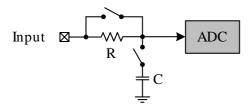
25.3.4.1 Input Filtering

The selected input signal can be filtered, either through an internal low pass RC filter or an internal decoupling capacitor. The different filter configurations can be enabled through the LPFMODE bits in ADCn_CTRL. For maximum SNR, LPFMODE is recommended set to DECAP, with a cutoff frequency of 31.5 MHz.

The RC input filter configuration is given in Figure 25.4 (p. 353). The resistance and capacitance values are given in the electrical characteristics for the device, named $R_{ADCFILT}$ and $C_{ADCFILT}$ respectively.



Figure 25.4. ADC RC Input Filter Configuration



25.3.4.2 Temperature Measurement

The ADC includes an internal temperature sensor. This sensor is characterized during production and the temperature readout from the ADC at production temperature, ADC0_TEMP_0_READ_1V25, is given in the Device Information (DI) page. The production temperature, CAL_TEMP_0, is also given in this page. The temperature gradient, TGRAD_ADCTH (mV/degree Celsius), for the sensor is found in the datasheet for the devices. By selecting 1.25 V internal reference and measuring the internal temperature sensor with 12 bit resolution, the temperature can be calculated according to the following formula:

ADC Temperature Measurement

Note

The minimum acquisition time for the temperature reference is found in the electrical characteristics for the device.

25.3.5 Reference Selection

The reference voltage can be selected from these sources:

- 1.25 V internal bandgap.
- 2.5 V internal bandgap.
- V_{DD}.
- 5 V internal differential bandgap.
- External single ended input from Ch. 6.
- Differential input, 2x(Ch. 6 Ch. 7).
- Unbuffered 2xV_{DD}.
- The 2.5 V reference needs a supply voltage higher than 2.5 V.
- The differential 5 V reference needs a supply voltage higher than 2.75 V.

Since the $2xV_{DD}$ differential reference is unbuffered, it is directly connected to the ADC supply voltage and more susceptible to supply noise. The V_{DD} reference is buffered both in single ended and differential mode.

If a differential reference with a larger range than the supply voltage is combined with single ended measurements, for instance the 5 V internal reference, the full ADC range will not be available because the maximum input voltage is limited by the maximum electrical ratings.

Note

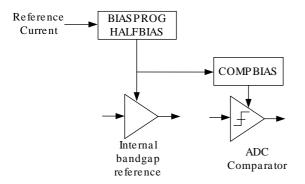
Single ended measurements with the external differential reference are not supported.



25.3.6 Programming of Bias Current

The bias current of the bandgap reference and the ADC comparator can be scaled by the BIASPROG, HALFBIAS and COMPBIAS bit fields of the ADCn_BIASPROG register. The BIASPROG and HALFBIAS bitfields scale the current of ADC bandgap reference, and the COMPBIAS bits provide an additional bias programming for the ADC comparator as illustrated in Figure 25.5 (p. 354). The electrical characteristics given in the datasheet require the bias configuration to be set to the default values, where no other bias values are given.

Figure 25.5. ADC Bias Programming



The BIASPROG bitfield must be set based on the ADC_CLK frequency. See the specific device data sheet for more information on these settings.

The minimum value of the BIASPROG and COMPBIAS bitfields of the ADCn_BIASPROG register (i.e. BIASPROG=0b0000, COMPBIAS=0b0000) represent the minimum bias currents. Similarly BIASPROG=0b1111 and COMPBIAS=0b1111 represent the maximum bias currents. Additionally, the bias current defined by the BIASPROG setting can be halved by setting the HALFBIAS bit of the ADCn_BIASPROG register.

The bias current settings should only be changed while the ADC is disabled.

25.3.7 ADC Modes

The ADC contains two separate programmable modes, one single sample mode and one scan mode. Both modes have separate configuration and result registers and can be set up to run only once per trigger or repetitively. The scan mode has priority over the single sample mode. However, if scan sequence is running, a triggered single sample will be interleaved between two scan samples.

25.3.7.1 Single Sample Mode

The single sample mode can be used to convert a single sample either once per trigger or repetitively. The configuration of the single sample mode is done in the ADCn_SINGLECTRL register and the results are found in the ADCn_SINGLEDATA register. The SINGLEDV bit in ADCn_STATUS is set high when there is valid data in the result register and is cleared when the data is read. The single mode results can also be read through ADCn_SINGLEDATAP without SINGLEDV being cleared. DIFF in ADCn_SINGLECTRL selects whether differential or single ended inputs are used and INPUTSEL selects input pin(s).

25.3.7.2 Scan mode

The scan mode is used to perform sweeps of the inputs. The configuration of the scan sequence is done in the ADCn_SCANCTRL register and the results are found in the ADCn_SCANDATA register. The SCANDV bit in ADCn_STATUS is set high when there is valid data in the result register and is cleared when the data is read. The scan mode results can also be read through ADCn_SCANDATAP without

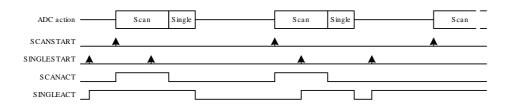


SCANDV being cleared. The inputs included in the sequence are defined by a the mask in INPUTMASK in ADCn_SCANCTRL. When the scan sequence is triggered, the sequence samples all inputs that are included in the mask, starting at the lowest pin number. DIFF in ADCn_SCANCTRL selects whether single ended or differential inputs are used.

25.3.7.3 Conversion Tailgating

The scan sequence has priority over the single sample mode. However, a scan trigger will not interrupt in the middle of a single conversion. If a scan sequence is triggered by a timer on a periodic basis, single sample just before a scan trigger can delay the start of the scan sequence, thus causing jitter in sample rate. To solve this, conversion tailgating can be chosen by setting TAILGATE in ADCn_CTRL. When this bit is set, any triggered single samples will wait for the next scan sequence to finish before activating (see Figure 25.6 (p. 355)). The single sample will then follow immediately after the scan sequence. In this way, the scan sequence will always start immediately when triggered, if the period between the scan triggers is big enough to allow any single samples that might be triggered to finish in between the scan sequences.

Figure 25.6. ADC Conversion Tailgating



25.3.7.4 Conversion Trigger

The conversion modes can be activated by writing a 1 to the SINGLESTART or SCANSTART bit in the ADCn_CMD register. The conversions can be stopped by writing a 1 to the SINGLESTOP or SCANSTOP bit in the ADCn_CMD register. A START command will have priority over a stop command. When the ADC is stopped in the middle of a conversion, the result buffer is cleared. The SINGLEACT and SCANACT bits in ADCn_STATUS are set high when the modes are actively converting or have pending conversions.

It is also possible to trigger conversions from PRS signals. The system requires one HFPERCLK cycle pulses to trigger conversions. Setting PRSEN in ADCn_SINGLECTRL/ADCn_SCANCTRL enables triggering from PRS input. Which PRS channel to listen to is defined by PRSSEL in ADCn_SINGLECTRL/ADCn_SCANCTRL. When PRS trigger is selected, it is still possible to trigger the conversion from software. The reader is referred to the PRS datasheet for more information on how to set up the PRS channels.

Note

The conversion settings should not be changed while the ADC is running as this can lead to unpredictable behavior.

The prescaled clock phase is always reset by a triggered conversion as long as a conversion is not ongoing. This gives predictable latency from the time of the trigger to the time the conversion starts, regardless of when in the prescaled clock cycle the trigger occur.

25.3.7.5 Results

The results are presented in 2's complement form and the format for differential and single ended mode is given in Table 25.1 (p. 356) and Table 25.2 (p. 356). If differential mode is selected, the results are sign extended up to 32-bit (shown in Table 25.4 (p. 357)).



Table 25.1. ADC Single Ended Conversion

Input/Reference	Res	sults
Inputivelelelice	Binary	Hex value
1	11111111111	FFF
0.5	01111111111	7FF
1/4096	00000000001	001
0	00000000000	000

Table 25.2. ADC Differential Conversion

Input/Reference	Res	sults
input/Kelerence	Binary	Hex value
0.5	011111111111	7FF
0.25	001111111111	3FF
1/2048	00000000001	001
0	00000000000	000
-1/2048	11111111111	FFF
-0.25	101111111111	BFF
-0.5	10000000000	800

25.3.7.6 Resolution

The ADC gives out 12-bit results, by default. However, if full 12-bit resolution is not needed, it is possible to speed up the conversion by selecting a lower resolution (N = 6 or 8 bits). For more information on the accuracy of the ADC, the reader is referred to the electrical characteristics section for the device.

25.3.7.7 Oversampling

To achieve higher accuracy, hardware oversampling can be enabled individually for each mode (Set RES in ADCn_SINGLECTRL/ADCn_SCANCTRL to 0x3). The oversampling rate (OVSRSEL in ADCn_CTRL) can be set to any integer power of 2 from 2 to 4096 and the configuration is shared between the scan and single sample mode (OVSRSEL field in ADCn_CTRL).

With oversampling, each selected input is sampled a number (given by the OVSR) of times, and the results are filtered by a first order accumulate and dump filter to form the end result. The data presented in the ADCn_SINGLEDATA and ADCn_SCANDATA registers are the direct contents of the accumulation register (sum of samples). However, if the oversampling ratio is set higher than 16x, the accumulated results are shifted to fit the MSB in bit 15 as shown in Table 25.3 (p. 357).



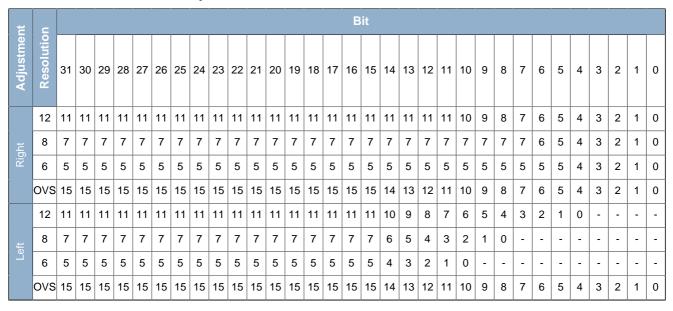
Table 25.3. Oversampling Result Shifting and Resolution

Oversampling setting	# right shifts	Result Resolution # bits
2x	0	13
4x	0	14
8x	0	15
16x	0	16
32x	1	16
64x	2	16
128x	3	16
256x	4	16
512x	5	16
1024x	6	16
2048x	7	16
4096x	8	16

25.3.7.8 Adjustment

By default, all results are right adjusted, with the LSB of the result in bit position 0 (zero). In differential mode the signed bit is extended up to bit 31, but in single ended mode the bits above the result are read as 0. By setting ADJ in ADCn_SINGLECTRL/ADCn_SCANCTRL, the results are left adjusted as shown in Table 25.4 (p. 357). When left adjusted, the MSB is always placed on bit 15 and sign extended to bit 31. All bits below the conversion result are read as 0 (zero).

Table 25.4. ADC Results Representation



25.3.8 Interrupts, PRS Output

The single and scan modes have separate interrupt flags indicating finished conversions. Setting one of these flags will result in an ADC interrupt if the corresponding interrupt enable bit is set in ADCn. IEN.

In addition to the finished conversion flags, there is a scan and single sample result overflow flag which signalizes that a result from a scan sequence or single sample has been overwritten before being read.

A finished conversion will result in a one HFPERCLK cycle pulse which is output to the Peripheral Reflex System (PRS).



25.3.9 DMA Request

The ADC has two DMA request lines, SINGLE and SCAN, which are set when a single or scan conversion has completed. The request are cleared when the corresponding single or scan result register is read.

25.3.10 Calibration

The ADC supports offset and gain calibration to correct errors due to process and temperature variations. This must be done individually for each reference used. The ADC calibration (ADCn_CAL) register contains four register fields for calibrating offset and gain for both single and scan mode. The gain and offset calibration are done in single mode, but the resulting calibration values can be used for both single and scan mode.

Gain and offset for the 1V25, 2V5 and VDD references are calibrated during production and the calibration values for these can be found in the Device Information page. During reset, the gain and offset calibration registers are loaded with the production calibration values for the 1V25 reference.

The SCANGAIN and SINGLEGAIN calibration fields are not used when the unbuffered differential 2xVDD reference is selected.

The effects of changing the calibration register values are given in Table 25.5 (p. 358). Step by step calibration procedures for offset and gain are given in Section 25.3.10.1 (p. 358) and Section 25.3.10.2 (p. 358).

Table 25.5. Calibration Register Effect

Calibration Register	ADC Result	Calibration Binary Value	Calibration Hex Value
Offset	Lowest Output	0111111	3F
Oliset	Highest Output	1000000	40
Gain	Lowest Output	0000000	00
Gain	Highest Output	1111111	7F

The offset calibration register expects a signed 2's complement value with negative effect. A high value gives a low ADC reading.

The gain calibration register expects an unsigned value with positive effect. A high value gives a high ADC reading.

25.3.10.1 Offset Calibration

Offset calibration must be performed prior to gain calibration. Follow these steps for the offset calibration in single mode:

- 1. Select wanted reference by setting the REF bitfield of the ADCn_SINGLECTRL register.
- 2. Set the AT bitfield of the ADCn_SINGLECTRL register to 16CYCLES.
- 3. Set the INPUTSEL bitfield of the ADCn_SINGLECTRL register to DIFF0, and set the DIFF bitfield to 1 for enabling differential input. Since the input voltage is 0, the expected ADC output is the half of the ADC code range as it is in differential mode.
- 4. A binary search is used to find the offset calibration value. Set the SINGLESTART bit in the ADCn_CMD register and read the ADCn_SINGLEDATA register. The result of the binary search is written to the SINGLEOFFSET field of the ADCn_CAL register.

25.3.10.2 Gain Calibration

Offset calibration must be performed prior to gain calibration. The Gain Calibration is done in the following manner:



- 1. Select an external ADC channel (a differential channel can also be used).
- 2. Apply an external voltage on the selected ADC input channel. This voltage should correspond to the top of the ADC range.
- 3. A binary search is used to find the gain calibration value. Set the SINGLESTART bit in the ADCn_CTRL register and read the ADCn_SINGLEDATA register. The target value is ideally the top of the ADC range, but it is recommended to use a value a couple of LSBs below in order to avoid overshooting. The result of the binary search is written to the SINGLEGAIN field of the ADCn_CAL register.



25.4 Register Map

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

Offset	Name	Туре	Description
0x000	ADCn_CTRL	RW	Control Register
0x004	ADCn_CMD	W1	Command Register
0x008	ADCn_STATUS	R	Status Register
0x00C	ADCn_SINGLECTRL	RW	Single Sample Control Register
0x010	ADCn_SCANCTRL	RW	Scan Control Register
0x014	ADCn_IEN	RW	Interrupt Enable Register
0x018	ADCn_IF	R	Interrupt Flag Register
0x01C	ADCn_IFS	W1	Interrupt Flag Set Register
0x020	ADCn_IFC	W1	Interrupt Flag Clear Register
0x024	ADCn_SINGLEDATA	R	Single Conversion Result Data
0x028	ADCn_SCANDATA	R	Scan Conversion Result Data
0x02C	ADCn_SINGLEDATAP	R	Single Conversion Result Data Peek Register
0x030	ADCn_SCANDATAP	R	Scan Sequence Result Data Peek Register
0x034	ADCn_CAL	RW	Calibration Register
0x03C	ADCn_BIASPROG	RW	Bias Programming Register

25.5 Register Description

25.5.1 ADCn_CTRL - Control Register

Offset															on																	
0x000	31	30	29	28	27	56	25	24	23	22	21	20	19	9	17	16	15	4	13	12	=	10	6	8	7	9	2	4	က	2	-	0
Reset						6	000							0x1F							00×0						0^0	2	0		Ç	0000
Access						2	≷ Ƴ							RW				RW									/\Q	2	RW		3	 ≩ Y
Name						i.	OVSKSEL							TIMEBASE							PRESC						PEMODE	200	TAILGATE			WAKMUPMODE

Bit	Name	Reset	Access	Description
31:28	Reserved	To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)		
27:24	OVSRSEL	0x0	RW	Oversample Rate Select

Select oversampling rate. Oversampling must be enabled for each mode for this setting to take effect.

Value	Mode	Description
0	X2	2 samples for each conversion result
1	X4	4 samples for each conversion result
2	X8	8 samples for each conversion result
3	X16	16 samples for each conversion result
4	X32	32 samples for each conversion result
5	X64	64 samples for each conversion result
6	X128	128 samples for each conversion result
7	X256	256 samples for each conversion result
8	X512	512 samples for each conversion result



Bit	Name	Reset	Access	Description										
	Value	Mode	De	escription										
	9	X1024	102	24 samples for each conversion result										
	10	X2048	204	2048 samples for each conversion result										
	11	X4096	409	4096 samples for each conversion result										
23:21	Reserved	To ensure c	compatibility with f	future devices, always write bits to 0. More information in Section 2.1 (p. 3)										
20:16	TIMEBASE	0x1F	RW	Time Base										
		used for ADC warm up se cles which should be set e		g to the HFPERCLK frequency. The time base is defined as a number of than 1 us .										
	Value			Description										
	TIMEBASE			ADC warm-up is set to TIMEBASE+1 HFPERCLK clock cycles and bandgap warm-up is set to 5x(TIMEBASE+1) HFPERCLK cycles.										
15	Reserved	To ensure c	compatibility with f	future devices, always write bits to 0. More information in Section 2.1 (p. 3)										
14:8	PRESC	0x00	RW	Prescaler Setting										
	Select clock div	vision factor.												
	Value		Desc	cription										
	PRESC		Cloc	ck division factor of PRESC+1.										
7:6	PRESC Reserved	To ensure c		ck division factor of PRESC+1. Suture devices, always write bits to 0. More information in Section 2.1 (p. 3)										
7:6 5:4		To ensure c												
	Reserved LPFMODE	0x0	compatibility with f	future devices, always write bits to 0. More information in Section 2.1 (p. 3)										
	Reserved LPFMODE	0x0	compatibility with fine RW	iuture devices, always write bits to 0. More information in Section 2.1 (p. 3) Low Pass Filter Mode										
	Reserved LPFMODE These bits cont	0x0 trol the filtering of the ADC	RW Cinput. Details on	future devices, always write bits to 0. More information in Section 2.1 (p. 3) Low Pass Filter Mode the filter characteristics can be found in the device datasheets.										
	Reserved LPFMODE These bits cont	0x0 trol the filtering of the ADC	RW C input. Details on De	future devices, always write bits to 0. More information in Section 2.1 (p. 3) Low Pass Filter Mode the filter characteristics can be found in the device datasheets.										
	Reserved LPFMODE These bits cont Value 0	0x0 trol the filtering of the ADC Mode BYPASS	RW Cinput. Details on No On	future devices, always write bits to 0. More information in Section 2.1 (p. 3) Low Pass Filter Mode the filter characteristics can be found in the device datasheets. escription of filter or decoupling capacitor										
	Reserved LPFMODE These bits cont Value 0 1	0x0 trol the filtering of the ADC Mode BYPASS DECAP	RW Cinput. Details on No On	Low Pass Filter Mode the filter characteristics can be found in the device datasheets. scription of filter or decoupling capacitor on chip decoupling capacitor selected										
5:4	Reserved LPFMODE These bits cont Value 0 1 2 TAILGATE	0x0 trol the filtering of the ADC Mode BYPASS DECAP RCFILT	RW C input. Details on No On	Low Pass Filter Mode the filter characteristics can be found in the device datasheets. scription of filter or decoupling capacitor of chip decoupling capacitor selected of chip RC filter selected										
5:4	Reserved LPFMODE These bits cont Value 0 1 2 TAILGATE	0x0 trol the filtering of the ADC Mode BYPASS DECAP RCFILT	RW C input. Details on No On	Low Pass Filter Mode the filter characteristics can be found in the device datasheets. scription of filter or decoupling capacitor of chip decoupling capacitor selected of chip RC filter selected										
5:4	Reserved LPFMODE These bits cont Value 0 1 2 TAILGATE Enable/disable	0x0 trol the filtering of the ADC Mode BYPASS DECAP RCFILT 0 conversion tailgating.	RW C input. Details on De No On RW	Low Pass Filter Mode the filter characteristics can be found in the device datasheets. scription of filter or decoupling capacitor of chip decoupling capacitor selected of chip RC filter selected										
5:4	Reserved LPFMODE These bits cont Value 0 1 2 TAILGATE Enable/disable Value	Ox0 trol the filtering of the ADC Mode BYPASS DECAP RCFILT 0 conversion tailgating. Description Scan sequence	RW C input. Details on De No On RW has priority, but can	Low Pass Filter Mode the filter characteristics can be found in the device datasheets. scription of filter or decoupling capacitor of chip decoupling capacitor selected of chip RC filter selected Conversion Tailgating										
5:4	Reserved LPFMODE These bits cont Value 0 1 2 TAILGATE Enable/disable Value	Ox0 trol the filtering of the ADC Mode BYPASS DECAP RCFILT 0 conversion tailgating. Description Scan sequence Scan sequence	RW C input. Details on De No On RW As priority, but can has priority and sing	Low Pass Filter Mode the filter characteristics can be found in the device datasheets. scription of filter or decoupling capacitor of chip decoupling capacitor selected of chip RC filter selected Conversion Tailgating the delayed by ongoing single samples. gle samples will only start immediately after scan sequence.										
5:4	Reserved LPFMODE These bits cont Value 0 1 2 TAILGATE Enable/disable Value 0 1	Ox0 trol the filtering of the ADC Mode BYPASS DECAP RCFILT 0 conversion tailgating. Description Scan sequence Scan sequence	RW C input. Details on De No On RW As priority, but can has priority and sing	Low Pass Filter Mode the filter characteristics can be found in the device datasheets. scription of filter or decoupling capacitor of chip decoupling capacitor selected of chip RC filter selected Conversion Tailgating of the delayed by ongoing single samples.										
5:4	Reserved LPFMODE These bits cont Value 0 1 2 TAILGATE Enable/disable Value 0 1 Reserved WARMUPMOD	Ox0 trol the filtering of the ADC Mode BYPASS DECAP RCFILT 0 conversion tailgating. Description Scan sequence Scan sequence	RW C input. Details on De No On RW RW Compatibility with first	Low Pass Filter Mode the filter characteristics can be found in the device datasheets. scription filter or decoupling capacitor a chip decoupling capacitor selected Conversion Tailgating the delayed by ongoing single samples. gle samples will only start immediately after scan sequence. future devices, always write bits to 0. More information in Section 2.1 (p. 3)										
5:4	Reserved LPFMODE These bits cont Value 0 1 2 TAILGATE Enable/disable Value 0 1 Reserved WARMUPMOD	Ox0 trol the filtering of the ADC Mode BYPASS DECAP RCFILT 0 conversion tailgating. Description Scan sequence Scan sequence To ensure content of the ADC OE OX0	RW C input. Details on De No On RW has priority, but can has priority and sing compatibility with fixed the RW	Low Pass Filter Mode the filter characteristics can be found in the device datasheets. scription filter or decoupling capacitor a chip decoupling capacitor selected Conversion Tailgating the delayed by ongoing single samples. gle samples will only start immediately after scan sequence. future devices, always write bits to 0. More information in Section 2.1 (p. 3)										
5:4	Reserved LPFMODE These bits cont Value 0 1 2 TAILGATE Enable/disable Value 0 1 Reserved WARMUPMOD Select Warm-up	Ox0 trol the filtering of the ADC Mode BYPASS DECAP RCFILT 0 conversion tailgating. Description Scan sequence Scan sequence To ensure companies of the ADC	RW C input. Details on De No On On RW has priority, but can has priority and sing	Low Pass Filter Mode the filter characteristics can be found in the device datasheets. escription of filter or decoupling capacitor of chip decoupling capacitor selected chip RC filter selected Conversion Tailgating of be delayed by ongoing single samples. gle samples will only start immediately after scan sequence. future devices, always write bits to 0. More information in Section 2.1 (p. 3) Warm-up Mode										
5:4	Reserved LPFMODE These bits cont Value 0 1 2 TAILGATE Enable/disable Value 0 1 Reserved WARMUPMOD Select Warm-up Value	Ox0 trol the filtering of the ADC Mode BYPASS DECAP RCFILT 0 conversion tailgating. Description Scan sequence Scan sequence To ensure concept of the ADC Mode	RW C input. Details on De No On On RW has priority, but can has priority and sing compatibility with fix RW	Low Pass Filter Mode the filter characteristics can be found in the device datasheets. scription offilter or decoupling capacitor chip decoupling capacitor selected chip RC filter selected Conversion Tailgating the delayed by ongoing single samples. gle samples will only start immediately after scan sequence. future devices, always write bits to 0. More information in Section 2.1 (p. 3) Warm-up Mode										
5:4	Reserved LPFMODE These bits cont Value 0 1 2 TAILGATE Enable/disable Value 0 1 Reserved WARMUPMOD Select Warm-up Value 0	Ox0 trol the filtering of the ADC Mode BYPASS DECAP RCFILT 0 conversion tailgating. Description Scan sequence Scan sequence To ensure companies of the ADC Mode NORMAL	RW C input. Details on De No On RW has priority, but can has priority and sing compatibility with fixed the priority of the priority and sing compatibility with fixed the priority and sing the priority and sing compatibility with fixed the priority and sing t	Low Pass Filter Mode the filter characteristics can be found in the device datasheets. scription offilter or decoupling capacitor of chip decoupling capacitor selected of chip RC filter selected Conversion Tailgating of the delayed by ongoing single samples. gle samples will only start immediately after scan sequence. future devices, always write bits to 0. More information in Section 2.1 (p. 3) Warm-up Mode										

25.5.2 ADCn_CMD - Command Register

Offset	Bit Position								
0x004	3 3 4 4 5 5 6 7 7 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9 9 9 9	2	- 0						
Reset	0	0	0 0						
Access	×	W1	× ×						
Name	SCANSTOP	SCANSTART	SINGLESTOP						



Bit	Name	Reset	Access	Description						
31:4	Reserved	To ensure c	ompatibility with fu	ure devices, always write bits to 0. More information in Section 2.1 (p.						
3	SCANSTOP	0	W1	Scan Sequence Stop						
	Write a 1 to stop scar	sequence.								
2	SCANSTART	0	W1	Scan Sequence Start						
	Write a 1 to start scar	n sequence.								
1	SINGLESTOP	0	W1	Single Conversion Stop						
	Write a 1 to stop sing	le conversion.								
0	SINGLESTART	0	W1	Single Conversion Start						
	Write to 1 to start sing	gle conversion.								

25.5.3 ADCn_STATUS - Status Register

0x0

Offset		Bit Position																														
0x008	33	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	က	2	-	0
Reset							0x0								0	0				0			0	0							0	0
Access							~							-	~	~				~			~	~							~	œ
Name							SCANDATASRC								SCANDV	SINGLEDV				WARM			SCANREFWARM	SINGLEREFWARM							SCANACT	SINGLEACT

Bit	Name	Reset	Access	Description
31:27	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)

This value indicates from which input channel the results in the ADCn_SCANDATA register originates.

R

Value	Mode	Description
0	CH0	Single ended mode: SCANDATA result originates from ADCn_CH0. Differential mode: SCANDATA result originates from ADCn_CH0-ADCn_CH1
1	CH1	Single ended mode: SCANDATA result originates from ADCn_CH1. Differential mode: SCANDATA result originates from ADCn_CH2_ADCn_CH3
2	CH2	Single ended mode: SCANDATA result originates from ADCn_CH2. Differential mode: SCANDATA result originates from ADCn_CH4-ADCn_CH5
3	CH3	Single ended mode: SCANDATA result originates from ADCn_CH3. Differential mode: SCANDATA result originates from ADCn_CH6-ADCn_CH7
4	CH4	SCANDATA result originates from ADCn_CH4
5	CH5	SCANDATA result originates from ADCn_CH5
6	CH6	SCANDATA result originates from ADCn_CH6
7	CH7	SCANDATA result originates from ADCn_CH7

Scan Data Source

9	SCANREFWARM	0	R	Scan Reference Warmed Un
11:10	Reserved	To ensure	e compatibility with	h future devices, always write bits to 0. More information in Section 2.1 (p. 3)
	ADC is warmed up.			
12	WARM	0	R	ADC Warmed Up
15:13	Reserved	To ensure	e compatibility witl	h future devices, always write bits to 0. More information in Section 2.1 (p. 3)
	Single conversion data	is valid.		
16	SINGLEDV	0	R	Single Sample Data Valid
	Scan conversion data is	s valid.		
17	SCANDV	0	R	Scan Data Valid
23:18	Reserved	To ensure	e compatibility with	h future devices, always write bits to 0. More information in Section 2.1 (p. 3)

SCANDATASRC

26:24



Bit	Name	Reset	Access	Description									
	Reference selected for s	can mode is warr	med up.										
8	SINGLEREFWARM	0	R	Single Reference Warmed Up									
	Reference selected for s	single mode is wa	rmed up.										
7:2	Reserved	Reserved To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)											
1	SCANACT	0	R	Scan Conversion Active									
	Scan sequence is active	or has pending o	conversions.										
0	SINGLEACT	0	R	Single Conversion Active									
	Single conversion is acti	ve or has pending	g conversions.										

25.5.4 ADCn_SINGLECTRL - Single Sample Control Register

Offset		Bit Position																														
0x00C	31	31 30 30 30 30 31 22 24 25 26 27 27 28 28 29 20 20 20 20 20 20 20 20 20 20 20 20 20									20	19	18	17	16	15	4	13	12	=	10	0	∞	7	9	2	4	ю	2	-	0	
Reset			0x0					0	0 00						0x0					•		OX O				5	3		0	0	0	
Access														RW																		
Name			PRSSEL					PRSEN		ŀ	Ā				REF							<u></u>	INFO I SEL				0	2		ADJ	DIFF	REP

Bit	Name	Reset	Access	Description
31	Reserved	To ensure co	mpatibility with i	future devices, always write bits to 0. More information in Section 2.1 (p. 3)
30:28	PRSSEL	0x0	RW	Single Sample PRS Trigger Select
	Select PRS trig	ger for single sample.		
	Value	Mode	De	escription
	0	PRSCH0	PF	RS ch 0 triggers single sample
	1	PRSCH1	PF	RS ch 1 triggers single sample
	2	PRSCH2	PF	RS ch 2 triggers single sample
	3	PRSCH3	PF	RS ch 3 triggers single sample

7	PRSCH7	PRS ch 7 triggers single sample
6	PRSCH6	PRS ch 6 triggers single sample
5	PRSCH5	PRS ch 5 triggers single sample
4	PRSCH4	PRS ch 4 triggers single sample
3	PRSCH3	PRS ch 3 triggers single sample
-	TROOFIE	The sit 2 angers sample

27:25 Reserved To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

24 PRSEN 0 RW Single Sample PRS Trigger Enable

Enabled/disable PRS trigger of single sample.

Value	Description
0	Single sample is not triggered by PRS input
1	Single sample is triggered by PRS input selected by PRSSEL

23:20 AT 0x0 RW Single Sample Acquisition Time

Select the acquisition time for single sample.

Value	Mode	Description
0	1CYCLE	1 ADC_CLK cycle acquisition time for single sample
1	2CYCLES	2 ADC_CLK cycles acquisition time for single sample
2	4CYCLES	4 ADC_CLK cycles acquisition time for single sample
3	8CYCLES	8 ADC_CLK cycles acquisition time for single sample
4	16CYCLES	16 ADC_CLK cycles acquisition time for single sample
5	32CYCLES	32 ADC_CLK cycles acquisition time for single sample
6	64CYCLES	64 ADC_CLK cycles acquisition time for single sample



15:12

Reserved

Bit	Name	Reset Acc	ss Description
	Value	Mode	Description
	7	128CYCLES	128 ADC_CLK cycles acquisition time for single sample
	8	256CYCLES	256 ADC_CLK cycles acquisition time for single sample

19 Reserved To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

18:16 REF 0x0 RW Single Sample Reference Selection

Select reference to ADC single sample mode.

Value	Mode	Description
0	1V25	Internal 1.25 V reference
1	2V5	Internal 2.5 V reference
2	VDD	Buffered VDD
3	5VDIFF	Internal differential 5 V reference
4	EXTSINGLE	Single ended external reference from ADCn_CH6
5	2XEXTDIFF	Differential external reference, 2x(ADCn_CH6 - ADCn_CH7)
6	2XVDD	Unbuffered 2xVDD

To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

11:8 INPUTSEL 0x0 RW Single Sample Input Selection

Select input to ADC single sample mode in either single ended mode or differential mode.

DIFF = 0		
Mode	Value	Description
CH0	0	ADCn_CH0
CH1	1	ADCn_CH1
CH2	2	ADCn_CH2
СНЗ	3	ADCn_CH3
CH4	4	ADCn_CH4
CH5	5	ADCn_CH5
CH6	6	ADCn_CH6
CH7	7	ADCn_CH7
TEMP	8	Temperature reference
VDDDIV3	9	VDD/3
VDD	10	VDD
VSS	11	VSS
VREFDIV2	12	VREF/2
DAC0OUT0	13	DAC0 output 0
DAC0OUT1	14	DAC0 output 1
DIFF = 1		
Mode	Value	Description
CH0CH1	0	Positive input: ADCn_CH0 Negative input: ADCn_CH1
CH2CH3	1	Positive input: ADCn_CH2 Negative input: ADCn_CH3
CH4CH5	2	Positive input: ADCn_CH4 Negative input: ADCn_CH5
CH6CH7	3	Positive input: ADCn_CH6 Negative input: ADCn_CH7
DIFF0	4	Differential 0 (Short between positive and negative inputs)

7:6 Reserved To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

5:4 RES 0x0 RW Single Sample Resolution Select

Select single sample conversion resolution.

Value	Mode	Description
0	12BIT	12-bit resolution
1	8BIT	8-bit resolution
2	6BIT	6-bit resolution
3	ovs	Oversampling enabled. Oversampling rate is set in OVSRSEL

Reserved To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)



Bit	Name	Reset	Access	Description				
2	ADJ	0	RW	Single Sample Result Adjustment				
	Select single s	sample result adjustment.						
	Value	Mode	De	scription				
	0	RIGHT	Re	sults are right adjusted				
	1	LEFT	Re	sults are left adjusted				
1	DIFF	0	RW	Single Sample Differential Mode				
	Select single	ended or differential input.						
	Value	Description						
	0	Single ended inpu	ut					
	1	Differential input						
0	REP	0	RW	Single Sample Repetitive Mode				
	Enable/disable	e repetitive single samples.						
	Value	Description						
	0	Single conversion	n mode is deactiva	ted after one conversion				
	1 Single conversion mode is converting continuously until SINGLESTOP is written							

25.5.5	Α	DC	DCn_SCANCTRL - Scan Control Register																													
Offset		Bit Position																														
0x010	31	30	59	28	27	56	52	24	23	72	21	20	19	18	17	16	15	4	13	12	=	10	6	80	7	9	2	4	က	7	-	0
Reset			0x0					0		2	OXO				0x0						0x0						0	040		0	0	0
Access			RW					RW		///	<u>}</u>				RW						RW						//\0	2		RW	RW	RW
Name			PRSSEL					PRSEN		Ļ	Ī				REF						INPUTMASK						020	2		ADJ	DIFF	REP
Bit	Na	me						Re	set			A	CC	ess		De	scr	ipti	on													
31	Re	serv	ed			To ensure compa				atib	tibility with future devices, always write bits to 0. More information in Section 2.1 (p.										. 3)											
30:28	PR	PRSSEL 0x0			R	W	Scan Sequence PRS Trigger Select																									

Select PRS trigger for scan sequence.

Value	Mode	Description
0	PRSCH0	PRS ch 0 triggers scan sequence
1	PRSCH1	PRS ch 1 triggers scan sequence
2	PRSCH2	PRS ch 2 triggers scan sequence
3	PRSCH3	PRS ch 3 triggers scan sequence
4	PRSCH4	PRS ch 4 triggers scan sequence
5	PRSCH5	PRS ch 5 triggers scan sequence
6	PRSCH6	PRS ch 6 triggers scan sequence
7	PRSCH7	PRS ch 7 triggers scan sequence

27:25 Reserved To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

24 **PRSEN** RW Scan Sequence PRS Trigger Enable

Enabled/disable PRS trigger of scan sequence.

Value	Description				
0	Scan sequence is not triggered by PRS input				
1	Scan sequence is triggered by PRS input selected by PRSSEL				

23:20 0x0 RW **Scan Sample Acquisition Time**

Select the acquisition time for scan samples.



15:8

Bit	Name	Reset	Acce	ess Description
	Value	Mode		Description
	0	1CYCLE		1 ADC_CLK cycle acquisition time for scan samples
	1	2CYCLES		2 ADC_CLK cycles acquisition time for scan samples
	2	4CYCLES		4 ADC_CLK cycles acquisition time for scan samples
	3	8CYCLES		8 ADC_CLK cycles acquisition time for scan samples
	4	16CYCLES		16 ADC_CLK cycles acquisition time for scan samples
	5	32CYCLES		32 ADC_CLK cycles acquisition time for scan samples
	6	64CYCLES		64 ADC_CLK cycles acquisition time for scan samples
	7	128CYCLES		128 ADC_CLK cycles acquisition time for scan samples
	8	256CYCLES		256 ADC_CLK cycles acquisition time for scan samples
19	Reserved	To ensure co	ompatibility v	with future devices, always write bits to 0. More information in Section 2.1 (p. 3)
18:16	REF	0x0	RW	Scan Sequence Reference Selection

Select reference to ADC scan sequence.

Value	Mode	Description				
0	1V25	Internal 1.25 V reference				
1	2V5	Internal 2.5 V reference				
2	VDD	VDD				
3	5VDIFF	Internal differential 5 V reference				
4	EXTSINGLE	Single ended external reference from ADCn_CH6				
5	2XEXTDIFF	Differential external reference, 2x(ADCn_CH6 - ADCn_CH7)				
6	2XVDD	Unbuffered 2xVDD				

INPUTMASK 0x00 RW Scan Sequence Input Mask

Set one or more bits in this mask to select which inputs are included the scan sequence in either single ended or differential mode.

DIFF = 0		
Mode	Value	Description
CH0	0000001	ADCn_CH0 included in mask
CH1	0000010	ADCn_CH1 included in mask
CH2	00000100	ADCn_CH2 included in mask
СНЗ	00001000	ADCn_CH3 included in mask
CH4	00010000	ADCn_CH4 included in mask
CH5	00100000	ADCn_CH5 included in mask
CH6	01000000	ADCn_CH6 included in mask
CH7	10000000	ADCn_CH7 included in mask
DIFF = 1		
Mode	Value	Description
CH0CH1	00000001	(Positive input: ADCn_CH0 Negative input: ADCn_CH1) included in mask
CH2CH3	00000010	(Positive input: ADCn_CH2 Negative input: ADCn_CH3) included in mask
CH4CH5	00000100	(Positive input: ADCn_CH4 Negative input: ADCn_CH5) included in mask
CH6CH7	00001000	(Positive input: ADCn_CH6 Negative input: ADCn_CH7) included in mask
	0001xxxx-1111xxxx	Reserved

7:6 Reserved To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

5:4 RES 0x0 RW Scan Sequence Resolution Select

Select scan sequence conversion resolution.

Value	Mode	Description
0	12BIT	12-bit resolution
1	8BIT	8-bit resolution
2	6BIT	6-bit resolution
3	ovs	Oversampling enabled. Oversampling rate is set in OVSRSEL

3 Reserved

To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)



Bit	Name	Reset	Access	Description	
2	ADJ	0	RW	Scan Sequence Result Adjustment	
	Select scan s	equence result adjustment.			
	Value	Mode	De	escription	
	0	RIGHT	Re	sults are right adjusted	
	1	LEFT	Re	sults are left adjusted	
1	DIFF	0	RW	Scan Sequence Differential Mode	
	Select single	ended or differential input.			
	Value	Description			
	0	Single ended inpu	ut		
	1	Differential input			
0	REP	0	RW	Scan Sequence Repetitive Mode	
	Enable/disab	le repetitive scan sequence.			
	Value	Description			
	0	Scan conversion	mode is deactivate	ed after one sequence	
	1	Scan conversion	mode is converting	g continuously until SCANSTOP is written	

25.5.6 ADCn_IEN - Interrupt Enable Register

Offset															Bi	t Po	siti	on														
0x014	31	30	59	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	∞	7	9	2	4	က	2	-	0
Reset																							0	0							0	0
Access	-																						8 ⊗	RW							RW	R W
Name																							SCANOF	SINGLEOF							SCAN	SINGLE

Bit	Name	Reset	Access	Description
31:10	Reserved	To ensure com	patibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
9	SCANOF	0	RW	Scan Result Overflow Interrupt Enable
	Enable/disable scan resu	It overflow interrupt	t.	
8	SINGLEOF	0	RW	Single Result Overflow Interrupt Enable
	Enable/disable single res	ult overflow interrup	pt.	
7:2	Reserved	To ensure com	patibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	SCAN	0	RW	Scan Conversion Complete Interrupt Enable
	Enable/disable scan conv	version complete in	terrupt.	
0	SINGLE	0	RW	Single Conversion Complete Interrupt Enable
	Enable/disable single cor	nversion complete i	nterrupt.	



25.5.7 ADCn_IF - Interrupt Flag Register

Offset															Bi	t Po	siti	on														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	0	8	7	9	2	4	ю	2	-	0
Reset		,			•				•													,	0	0						,	0	0
Access																							~	~							~	~
Name																							SCANOF	SINGLEOF							SCAN	SINGLE

Bit	Name	Reset	Access	Description
31:10	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
9	SCANOF	0	R	Scan Result Overflow Interrupt Flag
	Indicates scan resu	It overflow when this b	it is set.	
8	SINGLEOF	0	R	Single Result Overflow Interrupt Flag
	Indicates single res	ult overflow when this	bit is set.	
7:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	SCAN	0	R	Scan Conversion Complete Interrupt Flag
	Indicates scan conv	version complete when	this bit is set.	
0	SINGLE	0	R	Single Conversion Complete Interrupt Flag
	Indicates single cor	nversion complete whe	n this bit is set.	

25.5.8 ADCn_IFS - Interrupt Flag Set Register

Offset															Bi	t Pc	siti	on														
0x01C	31	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	∞	7	9	2	4	ю	7	-	0
Reset																						,	0	0						,	0	0
Access																						-	×	W							N V	W1
Name																							SCANOF	SINGLEOF							SCAN	SINGLE

Bit	Name	Reset	Access	Description
31:10	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
9	SCANOF	0	W1	Scan Result Overflow Interrupt Flag Set
	Write to 1 to set scan	result overflow inter	rupt flag	
8	SINGLEOF	0	W1	Single Result Overflow Interrupt Flag Set
	Write to 1 to set single	e result overflow inte	errupt flag.	
7:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	SCAN	0	W1	Scan Conversion Complete Interrupt Flag Set
	Write to 1 to set scan	conversion complete	e interrupt flag.	
0	SINGLE	0	W1	Single Conversion Complete Interrupt Flag Set
	Write to 1 to set single	e conversion comple	te interrupt flag.	



25.5.9 ADCn_IFC - Interrupt Flag Clear Register

Offset															Bi	t Pc	siti	on														
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	∞	7	9	2	4	က	2	-	0
Reset																							0	0							0	0
Access																							W	W							W	W
Name																							SCANOF	SINGLEOF							SCAN	SINGLE

Bit	Name	Reset	Access	Description
31:10	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
9	SCANOF	0	W1	Scan Result Overflow Interrupt Flag Clear
	Write to 1 to clear s	can result overflow inte	errupt flag.	
8	SINGLEOF	0	W1	Single Result Overflow Interrupt Flag Clear
	Write to 1 to clear s	ingle result overflow in	terrupt flag.	
7:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	SCAN	0	W1	Scan Conversion Complete Interrupt Flag Clear
	Write to 1 to clear s	can conversion comple	ete interrupt flag.	
0	SINGLE	0	W1	Single Conversion Complete Interrupt Flag Clear
	Write to 1 to clear s	ingle conversion comp	lete interrupt flag.	

25.5.10 ADCn_SINGLEDATA - Single Conversion Result Data

Offset															Bi	t Po	siti	on						,								
0x024	31	8	53	78	27	26	25	24	23	22	21	20	19	8	17	16	15	41	13	12	=	10	6	80	7	9	2	4	က	2	-	0
Reset																	00000000															
Access																	צ															
Name																\ \ \ \	K K															

Bit	Name	Reset	Access	Description
31:0	DATA	0x00000000	R	Single Conversion Result Data
	The register holds the register.	ne results from the last s	single convers	ion. Reading this field clears the SINGLEDV bit in the ADCn_STATUS



25.5.11 ADCn_SCANDATA - Scan Conversion Result Data

Offset															Bi	t Po	siti	on														
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	8	7	9	2	4	3	2	-	0
Reset																000000	000000000000000000000000000000000000000															
Access																۵	۷															
Name																\ \ \	<u> </u>															

Bit	Name	Reset	Access	Description
31:0	DATA	0x00000000	R	Scan Conversion Result Data
	The register holds the resu	Its from the last sca	n conversion.	Reading this field clears the SCANDV bit in the ADCn_STATUS register.

25.5.12 ADCn_SINGLEDATAP - Single Conversion Result Data Peek Register

Offset								·				·		E	it Po	ositi	on													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	17	16	15	14	13	11	10	6	8	7	9	2	4	က	2	-	0
Reset																000000000														
Access															ſ	¥														
Name															(H	DATAP														

Bit	Name	Reset	Access	Description
31:0	DATAP	0x0000000	R	Single Conversion Result Data Peek
	The register holds the resistance SINGLE DMA request.	ults from the last s	single conversi	on. Reading this field will not clear SINGLEDV in ADCn_STATUS or



25.5.13 ADCn_SCANDATAP - Scan Sequence Result Data Peek Register

Offset	Bit Position
0x030	33 4 5 6 6 7 7 8 8 8 9 9 10
Reset	00000000000000000000000000000000000000
Access	α
Name	DATAP

	3it	Name	Reset	Access	Description
3	31:0	DATAP	0x00000000	R	Scan Conversion Result Data Peek
		The register holds the resul DMA request.	ts from the last sca	n conversion.	Reading this field will not clear SCANDV in ADCn_STATUS or single

25.5.14 ADCn_CAL - Calibration Register

Offset															Bi	t Pc	siti	on														
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	8	7	9	2	4	ю	2	-	0
Reset					0x3F								00×0								0x3F								0x0			
Access					RW								₽								RW								RW			
Name					SCANGAIN								SCANOFFSET								SINGLEGAIN								SINGLEOFFSET			

Bit	Name	Reset	Access	Description
DIL	Ivaille	Reset	Access	Description
31	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3
30:24	SCANGAIN	0x3F	RW	Scan Mode Gain Calibration Value
	•	reference during res		can conversions. This field is set to the production gain calibration value to value might differ from device to device. The field is unsigned. Higher
23	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3
22:16	SCANOFFSET	0x00	RW	Scan Mode Offset Calibration Value
22:16	This register contains	the offset calibration	on value used with	h scan conversions. This field is set to the production offset calibration reset value might differ from device to device. The field is encoded a
	This register contains value for the 1V25 int	the offset calibration ernal reference duri ent number. Higher	on value used with ng reset, hence the values lead to low	h scan conversions. This field is set to the production offset calibration reset value might differ from device to device. The field is encoded a
22:16 15 14:8	This register contains value for the 1V25 int a signed 2's complement	the offset calibration ernal reference duri ent number. Higher	on value used with ng reset, hence the values lead to low	h scan conversions. This field is set to the production offset calibration reset value might differ from device to device. The field is encoded a ver ADC results.
15	This register contains value for the 1V25 int a signed 2's complem Reserved SINGLEGAIN This register contains	the offset calibration ernal reference during the transmission of the control of the calibration reference during residuals.	on value used with ng reset, hence the values lead to low ompatibility with furnithm. RW value used with si	h scan conversions. This field is set to the production offset calibration are reset value might differ from device to device. The field is encoded a ver ADC results. Iture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15	This register contains value for the 1V25 int a signed 2's complem Reserved SINGLEGAIN This register contains for the 1V25 internal	the offset calibrativernal reference during the transmission of the control of the calibration of the calibration reference during research and calibration research calibration research calibration reference during research calibration research calibration reference during reference during reference during reference during reference during reference during reference during reference during reference during reference during reference during reference during reference during reference during reference during reference during reference during reference during reference during reference during refer	on value used with ng reset, hence the values lead to low ompatibility with further RW value used with siest, hence the research	h scan conversions. This field is set to the production offset calibration reset value might differ from device to device. The field is encoded a wer ADC results. Iture devices, always write bits to 0. More information in Section 2.1 (p. 3 Single Mode Gain Calibration Value ingle conversions. This field is set to the production gain calibration value



Bit	Name	Reset	Access	Description
DIL	Ivaille	I LESEL	AUUUUS	Description

This register contains the offset calibration value used with single conversions. This field is set to the production offset calibration value for the 1V25 internal reference during reset, hence the reset value might differ from device to device. The field is encoded as a signed 2's complement number. Higher values lead to lower ADC results.

25.5.15 ADCn_BIASPROG - Bias Programming Register

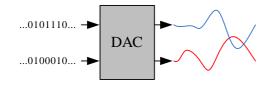
Offset															Bi	t Pc	siti	on														
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	7	10	6	8	7	9	2	4	3	2	1	0
Reset																							3			-				0x7	;	
Access																				-			<u> </u>			₩.				Z.		
Name																						OVIDANO				HALFBIAS				BIASPROG		

Bit	Name	Reset	Access	Description
31:12	Reserved	To ensure c	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
11:8	COMPBIAS	0x7	RW	Comparator Bias Value
	These bits are used	to adjust the bias cur	rent to the ADC C	omparator.
7	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
6	HALFBIAS	1	RW	Half Bias Current
	Set this bit to halve	the bias current.		
5:4	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
3:0	BIASPROG	0x7	RW	Bias Programming Value
		to adjust the bias cur		



26 DAC - Digital to Analog Converter





Quick Facts

What?

The DAC 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 kilo samples/ second and with 12-bit accuracy.

Why?

The DAC is able to generate accurate analog signals using only a limited amount of energy.

How?

The DAC can generate high-resolution analog signals while the MCU is operating at low frequencies and with low total power consumption. Using DMA and a timer, the DAC can be used to generate waveforms without any CPU intervention.

26.1 Introduction

The Digital to Analog Converter (DAC) can convert a digital value to an analog output voltage. The DAC is fully differential rail-to-rail, with 12-bit resolution. It has two single ended output buffers which can be combined into one differential output. The DAC may be used for a number of different applications such as sensor interfaces or sound output.

26.2 Features

- 500 ksamples/s operation
- Two single ended output channels
 - · Can be combined into one differential output
- Integrated prescaler with division factor selectable between 1-128
- Selectable voltage reference
 - Internal 2.5V
 - Internal 1.25V
 - V_{DD}
- · Conversion triggers
 - · Data write
 - PRS input
- · Automatic refresh timer
 - Selection from 16-64 prescaled HFPERCLK cycles
 - · Individual refresh enable for each channel
- Interrupt generation on finished conversion
 - Separate interrupt flag for each channel
- PRS output pulse on finished conversion
 - · Separate line for each channel
- DMA request on finished conversion
 - Separate request for each channel
- Support for offset and gain calibration

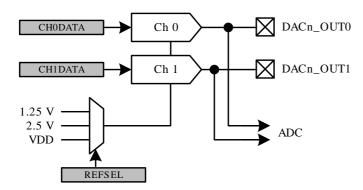


- · Output to ADC
- · Sine generation mode
- · Optional high strength line driver

26.3 Functional Description

An overview of the DAC module is shown in Figure 26.1 (p. 374).

Figure 26.1. DAC Overview



26.3.1 Conversions

The DAC consists of two channels (Channel 0 and 1) with separate 12-bit data registers (DACn_CH0DATA and DACn_CH1DATA). These can be used to produce two independent single ended outputs or the channel 0 register can be used to drive both outputs in differential mode. The DAC supports three conversion modes, continuous, sample/hold, sample/off.

26.3.1.1 Continuous Mode

In continuous mode the DAC channels will drive their outputs continuously with the data in the DACn_CHxDATA registers. This mode will maintain the output voltage and refresh is therefore not needed.

26.3.1.2 Sample/Hold Mode

In sample/hold mode, the DAC core converts data on a triggered conversion and then holds the output in a sample/hold element. When not converting, the DAC core is turned off between samples, which reduces the power consumption. Because of output voltage drift the sample/hold element will only hold the output for a certain period without a refresh conversion. The reader is referred to the electrical characteristics for the details on the voltage drift. The sampling period in this mode is set to the length of one prescaled clock cycle.

26.3.1.3 Sample/Off Mode

In sample/off mode the DAC and the sample/hold element is turned completely off between samples, tri-stating the DAC output. This requires the DAC output voltage to be held externally. The references are also turned off between samples, which means that a new warm-up period is needed before each conversion. The sampling period in this mode is set to the length of one prescaled clock cycle.

26.3.1.4 Conversion Start

The DAC channel must be enabled before it can be used. When the channel is enabled, a conversion can be started by writing to the DACn_CHxDATA register. These data registers are also mapped into



a combined data register, DACn_COMBDATA, where the data values for both channels can be written simultaneously. Writing to this register will start all enabled channels.

If the PRSEN bit in DACn_CHxCTRL is set, a DAC conversion on channel x will not be started by data write, but when a positive one HFPERCLK cycle pulse is received on the PRS input selected by PRSSEL in DACn_CHxCTRL.

The CH0DV and CH1DV bits in DACn_STATUS indicate that the corresponding channel contains data that has not yet been converted.

When entering Energy Modes 2,3 or 4, both DAC channels must be stopped. If the DAC is enabled for the first time after entering Energy Mode 2,3 or 4 the output of the DAC will be undefined. This can be worked around by enabling the DAC before entering a lower energy mode. The DAC channel can be enabled and the data registers written to even though the output is disabled.

26.3.1.5 Clock Prescaling

The DAC has an internal clock prescaler, which can divide the HFPERCLK by any factor between 1 and 128, by setting the PRESC bits in DACnCTRL. The resulting DAC_CLK is used by the converter core and the frequency is given by Equation 26.1 (p. 375):

DAC Clock Prescaling
$$f_{DAC_CLK} = f_{HFPERCLK} / 2^{PRESC}$$
 (26.1)

where f_{HFPERCLK} is the HFPERCLK frequency. One conversion takes 2 DAC_CLK cycles and the DAC_CLK should not be set higher than 1 MHz.

Normally the PRESCALER runs continuously when either of the channels are enabled. When running with a prescaler setting higher than 0, there will be an unpredictable delay from the time the conversion was triggered to the time the actual conversion takes place. This is because the conversions is controlled by the prescaled clock and the conversion can arrive at any time during a prescaled clock (DAC_CLK) period. However, if the CH0PRESCRST bit in DACn_CTRL 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 trigger and conversion.

26.3.2 Reference Selection

Three internal voltage references are available and are selected by setting the REFSEL bits in DACn_CTRL:

- Internal 2.5V
- Internal 1.25V
- V_{DD}

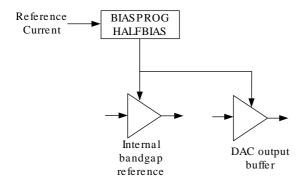
The reference selection can only be changed while both channels are disabled. The references for the DAC need to be enabled for some time before they can be used. This is called the warm-up period, and starts when one of the channels is enabled. For a bandgap reference, this period is 5 DAC_CLK cycles while the V_{DD} reference needs 1 DAC_CLK cycle. The DAC will time this period automatically(given that the prescaler is set correctly) and delay any conversion triggers received during the warm-up until the references have stabilized.

26.3.3 Programming of Bias Current

The bias current of the bandgap reference and the DAC output buffer can be scaled by the BIASPROG and HALFBIAS bit fields of the DACn_BIASPROG register as illustrated in Figure 26.2 (p. 376).



Figure 26.2. DAC Bias Programming



The minimum value of the BIASPROG bit-field of the DACn_BIASPROG register (i.e. BIASPROG=0b0000) represents the minimum bias current. Similarly BIASPROG=0b1111 represents the maximum bias current. The bias current defined by the BIASPROG setting can be halved by setting the HALFBIAS bit of the DACn_BIASPROG register.

The bias current settings should only be changed while both DAC channels are disabled. The electrical characteristics given in the datasheet require the bias configuration to be set to the default values, where no other bias values are given.

26.3.4 Mode

The two DAC channels can act as two separate single ended channels or be combined into one differential channel. This is selected through the DIFF bit in DACn_CTRL.

26.3.4.1 Single Ended Output

When operating in single ended mode, the channel 0 output is on DACn_OUT0 and the channel 1 output is on DACn_OUT1. The output voltage can be calculated using Equation 26.2 (p. 376)

DAC Single Ended Output Voltage

$$V_{OUT} = V_{DACn_OUTx} - V_{SS} = V_{ref} x CHxDATA/4095$$
 (26.2)

where CHxDATA is a 12-bit unsigned integer.

26.3.4.2 Differential Output

When operating in differential mode, both DAC outputs are used as output for the bipolar voltage. The differential conversion uses DACn_CH0DATA as source. The positive output is on DACn_OUT1 and the negative output is on DACn_OUT0. Since the output can be negative, it is expected that the data is written in 2's complement form with the MSB of the 12-bit value being the signed bit. The output voltage can be calculated using Equation 26.3 (p. 376):

DAC Differential Output Voltage

$$V_{OUT} = V_{DACn_OUT1} - V_{DACn_OUT0} = V_{ref} \times CH0DATA/2047$$
 (26.3)

where CH0DATA is a 12-bit signed integer. The common mode voltage is $V_{DD}/2$.

26.3.5 Sine Generation Mode

The DAC contains an automatic sine-generation mode, which is enabled by setting the SINEMODE bit in DACn_CTRL. In this mode, the DAC data is overridden with a conversion data taken from a sine lookup

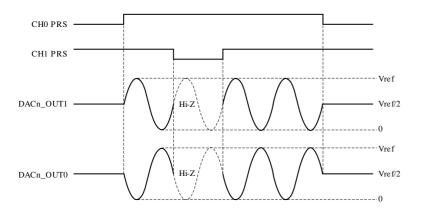


table. The sine signal is controlled by the PRS line selected by CH0PRSSEL in DACn_CH0CTRL. When the PRS line is low, a voltage of Vref/2 will be produced. When the line is high, a sine wave will be produced. Each period, starting at 0 degrees, is made up of 16 samples and the frequency is given by Equation 26.4 (p. 377):

DAC Sine Generation
$$f_{sine} = f_{HFPERCLK} / 32 \times 2^{PRESC}$$
 (26.4)

The SINE wave will be output on channel 0. If DIFF is set in DACn_CTRL, the sine wave will be output on both channels (if enabled), but inverted (see Figure 26.1 (p. 374)). Note that when OUTENPRS in DACn_CTRL is set, the sine output will be reset to 0 degrees when the PRS line selected by CH1PRSSEL is low.

Figure 26.3. DAC Sine Mode



26.3.6 Interrupts and PRS Output

Both DAC channels have separate interrupt flags (in DACn_IF) indicating that a conversion has finished on the channel and that new data can be written to the data registers. Setting one of these flags will result in a DAC interrupt if the corresponding interrupt enable bit is set in DACn_IEN. All generated interrupts from the DAC will activate the same interrupt vector when enabled.

The DAC has two PRS outputs which will carry a one cycle (HFPERCLK) high pulse when the corresponding channel has finished a conversion.

26.3.7 DMA Request

The DAC sends out a DMA request when a conversion on a channel is complete. This request is cleared when the corresponding channel's data register is written.

26.3.8 Analog Output

Each DAC channel has its own output pin (DACn_OUT0 and DACn_OUT1) in addition to an internal loopback to the ADC. These outputs can be enabled and disabled individually in the EN field in DACn_CHxCTRL registers in combination with OUTPUTSEL in DACn_CTRL. The DAC outputs can also be directed to the ADC, which is also configurable in the OUTPUTSEL field in DACn_CTRL.

The DAC outputs are tri-stated when the channels are not enabled. By setting the OUTENPRS bit in DACn_CTRL, the outputs are also tri-stated when the PRS line selected by CH1PRSSEL in DACn_CH1CTRL is low. When the PRS signal is high, the outputs are enabled as normal.



26.3.9 Calibration

The DAC contains a calibration register, DACn_CAL, where calibration values for both offset and gain correction can be written. Offset calibration is done separately for each channel through the CHxOFFSET bit-fields. Gain is calibrated in one common register field, GAIN. The gain calibration is linked to the reference and when the reference is changed, the gain must be re-calibrated. Gain and offset for the 1V25, 2V5 and VDD references are calibrated during production and the calibration values for these can be found in the Device Information page. During reset, the gain and offset calibration registers are loaded with the production calibration values for the 1V25 reference.



26.4 Register Map

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

Offset	Name	Туре	Description
0x000	DACn_CTRL	RW	Control Register
0x004	DACn_STATUS	R	Status Register
0x008	DACn_CH0CTRL	RW	Channel 0 Control Register
0x00C	DACn_CH1CTRL	RW	Channel 1 Control Register
0x010	DACn_IEN	RW	Interrupt Enable Register
0x014	DACn_IF	R	Interrupt Flag Register
0x018	DACn_IFS	W1	Interrupt Flag Set Register
0x01C	DACn_IFC	W1	Interrupt Flag Clear Register
0x020	DACn_CH0DATA	RW	Channel 0 Data Register
0x024	DACn_CH1DATA	RW	Channel 1 Data Register
0x028	DACn_COMBDATA	W	Combined Data Register
0x02C	DACn_CAL	RW	Calibration Register
0x030	DACn_BIASPROG	RW	Bias Programming Register

26.5 Register Description

26.5.1 DACn_CTRL - Control Register

Offset															Bi	t Po	ositi	on														
0x000	31	30	29	28	27	26	22	24	23	22	21	20	19	92	17	16	15	4	13	12	7	9	6	∞	7	9	2	4	ო	2	-	0
Reset											2) X			0×0								2	S S	0	0	5	5	2	Š	0	0
Access							3	≥ Y			-W								2	} Ł	RW	8 ≷	<u> </u>	<u> </u>	×	} {	RW	R ≷				
Name											0	И			PRESC								L C L	KETOEL	CHOPRESCRST	OUTENPRS	E C	500	a CONVINCO		SINEMODE	DIFF

Bit	Name	Reset	Access	Description
31:22	Reserved	To ensure con	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
21:20	REFRSEL	0x0	RW	Refresh Interval Select
	Select refresh counter	timeout value. A d	channel x will be	e refreshed with the interval set in this register if the REFREN bit in

DACn_CHxCTRL is set.

Value	Mode	Description
0	8CYCLES	All channels with enabled refresh are refreshed every 8 prescaled cycles
1	16CYCLES	All channels with enabled refresh are refreshed every 16 prescaled cycles
2	32CYCLES	All channels with enabled refresh are refreshed every 32 prescaled cycles
3	64CYCLES	All channels with enabled refresh are refreshed every 64 prescaled cycles

18:16	DRESC	0v0	D\M	Prescalar Satting
19	Reserved	To ensure compa	atibility with futu	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)

Select clock division factor.

Value	Description
PRESC	Clock division factor of 2^PRESC.



Bit	Name	Reset	Acce	ss Description
15:10	Reserved	To ensure o	compatibility v	vith future devices, always write bits to 0. More information in Section 2.1 (p. 3)
9:8	REFSEL	0x0	RW	Reference Selection
	Select reference	e.		
	Value	Mode		Description
	0	1V25		Internal 1.25 V bandgap reference
	1	2V5		Internal 2.5 V bandgap reference
	2	VDD		VDD reference
7	CH0PRESCRS	ST 0	RW	Channel 0 Start Reset Prescaler
	Select if presca	aler is reset on channel 0	start.	
	Value	Description		
	0	Prescaler not re	eset on channel	0 start
	1	Prescaler reset	on channel 0 s	tart
6	OUTENPRS	0	RW	PRS Controlled Output Enable
		ontrol of DAC output enab		
		· · · · · · · · · · · · · · · · · · ·	,io.	
	Value	Description DAC output ena	ahla alwaya aa	
	1			by PRS signal selected for CH1.
				•
5:4	OUTMODE	0x1	RW	Output Mode
	Select output n	node.		
	Value	Mode		Description
	0	DISABLE		DAC output to pin and ADC disabled
	1	PIN		DAC output to pin enabled. DAC output to ADC disabled
	2	ADC		DAC output to pin disabled. DAC output to ADC enabled
	3	PINADC		DAC output to pin and ADC enabled
3:2	CONVMODE	0x0	RW	Conversion Mode
	Configure conv	version mode.		
	Value	Mode		Description
	0	CONTINUOUS		DAC is set in continuous mode
	1	SAMPLEHOLD		DAC is set in sample/hold mode
	2	SAMPLEOFF		DAC is set in sample/shut off mode
1	SINEMODE	0	RW	Sine Mode
	Enable/disable	sine mode.		
	Value	Description		
	0	Sine mode disa	bled. Sine rese	t to 0 degrees
	1	Sine mode ena	bled	
0	DIFF	0	RW	Differential Mode
	Select single e	nded or differential mode.		
	Value	Description		
	0	Single ended o	utput	
	1	Differential outp	out	



26.5.2 DACn_STATUS - Status Register

Offset				,	,				,						Bi	t Pc	siti	on						,								
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	11	10	တ	œ	7	9	2	4	က	2	_	0
Reset				•	•				•							•	•							•							0	0
Access																															œ	<u>م</u>
Name																															CH1DV	СНОДЛ

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure comp	atibility with fut	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	CH1DV	0	R	Channel 1 Data Valid
	This bit is set high when C	H1DATA is written	and is set low	when CH1DATA is used in conversion.
0	CH0DV	0	R	Channel 0 Data Valid
	This bit is set high when C	H0DATA is written	and is set low	when CH0DATA is used in conversion.

26.5.3 DACn_CH0CTRL - Channel 0 Control Register

Offset															Bi	t Pc	siti	on														
0x008	31	30	29	28	27	56	25	24	23	22	21	20	19	18	17	16	15	4	13	12	11	10	6	80	7	9	2	4	8	2	-	0
Reset			•	,							•	•			,			•			•		•				0x0			0	0	0
Access																											R W			RW	W.	RW
Name																											PRSSEL			PRSEN	REFREN	Z

Bit	Name		Reset	Acces	ss Description
31:7	Reserved		To ensure co	ompatibility wi	ith future devices, always write bits to 0. More information in Section 2.1 (p. 3)
6:4	PRSSEL		0x0	RW	Channel 0 PRS Trigger Select
	Select Channe	el 0 PRS input	channel.		
	Value	Mode			Description
	0	PRSCH0			PRS ch 0 triggers channel 0 conversion.
	1	PRSCH1			PRS ch 1 triggers channel 0 conversion.
	2	PRSCH2			PRS ch 2 triggers channel 0 conversion.
	3	PRSCH3			PRS ch 3 triggers channel 0 conversion.
	4	PRSCH4			PRS ch 4 triggers channel 0 conversion.
	5	PRSCH5			PRS ch 5 triggers channel 0 conversion.
	6	PRSCH6			PRS ch 6 triggers channel 0 conversion.
	7	PRSCH7			PRS ch 7 triggers channel 0 conversion.
3	Reserved		To ensure co	ompatibility wi	ith future devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	PRSEN		0	RW	Channel 0 PRS Trigger Enable
	Select Channe	el 0 conversior	n trigger.		
	Value	D	escription		
	0	C	hannel 0 is trigo	nered by CH0D	ATA or COMBDATA write

Channel 0 is triggered by PRS input



Bit	Name	Reset	Access	Description
1	REFREN	0	RW	Channel 0 Automatic Refresh Enable
	Set to enable automatic	refresh of channel 0	. Refresh perio	d is set by REFRSEL in DACn_CTRL.
	Value	Description		
	0	Channel 0 is not refu	ally	
	1	Channel 0 is refresh	ed automatically	
0	EN	0	RW	Channel 0 Enable
	Enable/disable channel	0.		

26.5.4 DACn_CH1CTRL - Channel 1 Control Register

Offset															Bi	t Pc	siti	on														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	-	10	6	8	7	9	2	4	က	2	-	0
Reset			•		•										•	•				•	•		•				0x0			0	0	0
Access																											RW			R W	RW	RW
Name																											PRSSEL			PRSEN	REFREN	EN

Bit	Name	Reset	Access	Description
31:7	Reserved	To ensu	ire compatibility with f	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
6:4	PRSSEL	0x0	RW	Channel 1 PRS Trigger Select
	Select Channe	el 1 PRS input channel.		
	Value	Mode	Des	scription
	0	PRSCH0	PR	S ch 0 triggers channel 1 conversion.
	1	PRSCH1	PR	S ch 1 triggers channel 1 conversion.
	2	PRSCH2	PR	S ch 2 triggers channel 1 conversion.
	3	PRSCH3	PR	S ch 3 triggers channel 1 conversion.
	4	PRSCH4	PR	S ch 4 triggers channel 1 conversion.
	5	PRSCH5	PR	S ch 5 triggers channel 1 conversion.
	6	PRSCH6	PR	S ch 6 triggers channel 1 conversion.
	7	PRSCH7	PR	S ch 7 triggers channel 1 conversion.
3	Reserved	To ensu	re compatibility with f	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	PRSEN	0	RW	Channel 1 PRS Trigger Enable
	Select Channe	el 1 conversion trigger.		
	Value	Description		
	0	Channel 1 is	s triggered by CH1DATA	or COMBDATA write
	1	Channel 1 is	s triggered by PRS input	
1	REFREN	0	RW	Channel 1 Automatic Refresh Enable
	Set to enable	automatic refresh of ch	annel 1. Refresh perio	od is set by REFRSEL in DACn_CTRL.
	Value	Description		
	0	Channel 1 i	s not refreshed automatic	cally
	1	Channel 1 is	s refreshed automatically	1
	!			
0	EN	0	RW	Channel 1 Enable



26.5.5 DACn_IEN - Interrupt Enable Register

Offset															Bi	t Po	siti	on														
0x010	31	30	29	28	27	56	25	24	23	22	21	20	19	18	17	16	15	41	13	12	=	10	တ	8	7	9	2	4	က	7	-	0
Reset												•			,								,				0	0			0	0
Access																											RW	RW			RW	RW
Name																											CH10F	CHOUF			CH1	СНО

Bit	Name	Reset	Access	Description
31:6	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
5	CH1UF	0	RW	Channel 1 Conversion Data Underflow Interrupt Enable
	Enable/disable cha	annel 1 data underflow i	interrupt.	
4	CH0UF	0	RW	Channel 0 Conversion Data Underflow Interrupt Enable
	Enable/disable cha	annel 0 data underflow i	interrupt.	
3:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	CH1	0	RW	Channel 1 Conversion Complete Interrupt Enable
	Enable/disable cha	annel 1 conversion com	plete interrupt.	
0	CH0	0	RW	Channel 0 Conversion Complete Interrupt Enable
	Enable/disable cha	annel 0 conversion com	plete interrupt.	

26.5.6 DACn_IF - Interrupt Flag Register

Offset															Bi	t Pc	siti	on														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	11	10	6	8	7	9	2	4	က	2	_	0
Reset			•	,	,						•	•	•								•		,				0	0			0	0
Access																											œ	œ			œ	~
Name																											CH10F	CHOUF			CH1	СНО

Bit	Name	Reset	Access	Description
31:6	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
5	CH1UF	0	R	Channel 1 Data Underflow Interrupt Flag
	Indicates channel	1 data underflow.		
4	CH0UF	0	R	Channel 0 Data Underflow Interrupt Flag
	Indicates channel	0 data underflow.		
3:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	CH1	0	R	Channel 1 Conversion Complete Interrupt Flag
	Indicates channel	1 conversion complete	and that new data	can be written to the data register.
0	CH0	0	R	Channel 0 Conversion Complete Interrupt Flag
	Indicates channel	O conversion complete	and that new data	can be written to the data register.



26.5.7 DACn_IFS - Interrupt Flag Set Register

Offset							,								Bi	t Pc	siti	on														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	11	10	6	8	7	9	2	4	က	2	_	0
Reset																											0	0			0	0
Access																											W1	W1			M	W1
Name																											CH1UF	CHOUF			CH1	СНО

Bit	Name	Reset	Access	Description
31:6	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
5	CH1UF	0	W1	Channel 1 Data Underflow Interrupt Flag Set
	Write to 1 to set ch	nannel 1 Data Underflov	v interrupt flag.	
4	CH0UF	0	W1	Channel 0 Data Underflow Interrupt Flag Set
	Write to 1 to set ch	nannel 0 Data Underflov	v interrupt flag.	
3:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	CH1	0	W1	Channel 1 Conversion Complete Interrupt Flag Set
	Write to 1 to set ch	nannel 1 conversion con	nplete interrupt fla	ag.
0	CH0	0	W1	Channel 0 Conversion Complete Interrupt Flag Set
	Write to 1 to set ch	nannel 0 conversion con	mplete interrupt fla	ag.

26.5.8 DACn_IFC - Interrupt Flag Clear Register

Offset															Bi	t Pc	siti	on														
0x01C	31	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	က	2	-	0
Reset									•			•				•					•						0	0			0	0
Access																											W	W			M	M
Name																											CH1UF	CHOUF			CH1	СНО

Bit	Name	Reset	Access	Description
31:6	Reserved	To ensure c	ompatibility with f	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
5	CH1UF	0	W1	Channel 1 Data Underflow Interrupt Flag Clear
	Write to 1 to clear of	channel 1 data underflo	ow interrupt flag.	
4	CH0UF	0	W1	Channel 0 Data Underflow Interrupt Flag Clear
	Write to 1 to clear of	channel 0 data underflo	ow interrupt flag.	
3:2	Reserved	To ensure c	ompatibility with f	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	CH1	0	W1	Channel 1 Conversion Complete Interrupt Flag Clear
	Write to 1 to clear of	channel 1 conversion of	omplete interrupt	flag.
0	CH0	0	W1	Channel 0 Conversion Complete Interrupt Flag Clear
	Write to 1 to clear of	channel 0 conversion o	omplete interrupt	flag.



26.5.9 DACn_CH0DATA - Channel 0 Data Register

Offset															Bi	t Po	siti	on														
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	3	2	-	0
Reset																										0	000x0					
Access																										i	S ⊗					
Name																										i i	DATA					

Bit	Name	Reset	Access	Description
31:12	Reserved	To ensure co	mpatibility with	future devices, always write bits to 0. More information in Section 2.1 (p. 3)
11:0	DATA	0x000	RW	Channel 0 Data
	This register conta	ins the value which will	be converted by	y channel 0.

26.5.10 DACn_CH1DATA - Channel 1 Data Register

Offset															Bi	t Po	siti	on														
0x024	31	30	29	28	27	26	25	24	23	22	21	70	19	18	17	16	15	4	13	12	=	10	6	8	7	9	2	4	က	2	-	0
Reset																										0	00000					
Access																										2	≩ Ƴ					
Name																										i i	DATA					

Bit	Name	Reset	Access	Description								
31:12	Reserved	To ensure comp	atibility with fut	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)								
11:0	DATA	0x000	RW	Channel 1 Data								
	This register contains the value which will be converted by channel 1.											

26.5.11 DACn_COMBDATA - Combined Data Register

Offset															Bi	it Po	siti	on												Ė		Ţ	
0x028	31	30	28 28 28 28 29 26 27 23 24 25 27 21 22 23 24 25 26 27 27 28 29 40 41 41 42 43 44 45 46 47 48 41 41 42 43 44 45 46 47 <th>16</th> <th>15</th> <th>14</th> <th>13</th> <th>12</th> <th>1</th> <th>10</th> <th>6</th> <th>8</th> <th>7</th> <th>9</th> <th>2</th> <th>4</th> <th>က</th> <th>) (</th> <th>2 4</th> <th>- </th> <th>0</th>								16	15	14	13	12	1	10	6	8	7	9	2	4	က) (2 4	-	0					
Reset		00000																				000x0											
Access					}										3																		
Name		CH1DATA																			СНОДАТА												

Bit	Name	Reset	Access	Description
31:28	Reserved	To ensure cor	mpatibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)



Bit	Name	Reset	Access	Description							
27:16	CH1DATA	0x000	W	Channel 1 Data							
	Data written to this	register will be written t	to DATA in DACn	_CH1DATA.							
15:12	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)							
11:0	CH0DATA	0x000	W	Channel 0 Data							
	Data written to this register will be written to DATA in DACn_CH0DATA.										

26.5.12 DACn_CAL - Calibration Register

Offset															Bi	t Po	osition															
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	1	10	စ	œ	7	9	2	4	က	2	-	0
Reset											0x40					0000										0000						
Access											W.						RW								RW							
Name											GAIN										CH10FFSET							THOUSE				

Bit	Name	Reset	Access	Description
31:23	Reserved	To ensure co	ompatibility with fu	iture devices, always write bits to 0. More information in Section 2.1 (p. 3)
22:16	GAIN	0x40	RW	Gain Calibration Value
				set to the production gain calibration value for the 1V25 internal reference to device. The field is unsigned. Higher values lead to lower DAC results.
15:14	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
13:8	CH1OFFSET	0x00	RW	Channel 1 Offset Calibration Value
	calibration value for		erence during res	channel 1 conversions. This field is set to the production channel 1 offset et, hence the reset value might differ from device to device. The field is results.
7:6	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
5:0	CH0OFFSET	0x00	RW	Channel 0 Offset Calibration Value
	calibration value for		erence during res	channel 0 conversions. This field is set to the production channel 0 offset et, hence the reset value might differ from device to device. The field is results.

26.5.13 DACn_BIASPROG - Bias Programming Register

Offset		Bit Position																														
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	8	7	9	2	4	3	2	-	0
Reset					•				•							•		•			•		•			-				0x7		
Access																										RW				8		
Name																										HALFBIAS				RIASPROG		

Bit	Name	Reset Access Description	
31:7	Reserved	To ensure compatibility with future devices, always write bits to 0. More information in Sect	ion 2.1 (p. 3)
6	HALFBIAS	1 RW Half Bias Current	

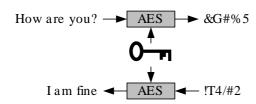


Bit	Name	Reset	Access	Description								
	Set this bit to halve th	ne bias current.										
5:4	Reserved	To ensure c	ompatibility with fo	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)								
3:0	BIASPROG	0x7	RW	Bias Programming Value								
	These bits control the bias current level.											



27 AES - Advanced Encryption Standard Accelerator





Quick Facts

What?

A fast and energy efficient hardware accelerator for AES-128 and AES-256 encryption and decryption.

Why?

Efficient encryption/decryption with little or no CPU intervention helps to meet the speed and energy demands of the application.

How?

High AES throughput allows the EFM32G to spend more time in lower energy modes. In addition, specialized data access functions allow autonomous DMA/AES operation in both EM0 and EM1.

27.1 Introduction

The Advanced Encryption Standard (FIPS-197) is a symmetric block cipher operating on 128-bit blocks of data and 128-, 192- or 256-bit keys.

The AES accelerator performs AES encryption and decryption with 128-bit or 256-bit keys. Encrypting or decrypting one 128-bit data block takes 54 HFCORECLK cycles with 128-bit keys and 75 HFCORECLK cycles with 256-bit keys. The AES module is an AHB slave which enables efficient access to the data and key registers. All write accesses to the AES module must be 32-bit operations, i.e. 8- or 16-bit operations are not supported.

27.2 Features

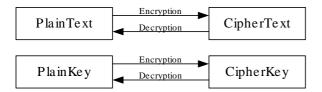
- AES hardware encryption/decryption
 - 128-bit key (54 HFCORECLK cycles)
 - 256-bit key (75 HFCORECLK cycles)
- Efficient CPU/DMA support
- Interrupt on finished encryption/decryption
- DMA request on finished encryption/decryption
- Key buffer in AES128 mode
- Optional XOR on Data write

27.3 Functional Description

Some data and a key must be loaded into the KEY and DATA registers before an encryption or decryption can take place. The input data before encryption is called the PlainText and output from the encryption is called CipherText. For encryption, the key is called PlainKey. After one encryption, the resulting key in the KEY registers is the CipherKey. This key must be loaded into the KEY registers before every decryption. After one decryption, the resulting key will be the PlainKey. The resulting PlainKey/CipherKey is only dependent on the value in the KEY registers before encryption/decryption. The resulting keys and data are shown in Figure 27.1 (p. 389) .



Figure 27.1. AES Key and Data Definitions



27.3.1 Encryption/Decryption

The AES module can be set to encrypt or decrypt by clearing/setting the DECRYPT bit in AES_CTRL. The AES256 bit in AES_CTRL configures the size of the key used for encryption/decryption. The AES_CTRL register should not be altered while AES is running, as this may lead to unpredictable behaviour.

An AES encryption/decryption can be started in the following ways:

- Writing a 1 to the START bit in AES_CMD
- Writing 4 times 32 bits to AES_DATA when the DATASTART control bit is set
- Writing 4 times 32 bits to AES_XORDATA when the XORSTART control bit is set

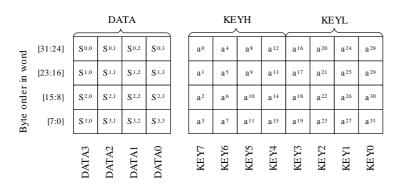
An AES encryption/decryption can be stopped by writing a 1 to the STOP bit in AES_CMD. The RUNNING bit in AES_STATUS indicates that an AES encryption/decryption is ongoing.

27.3.2 Data and Key Access

The AES module contains a 128-bit DATA (State) register and two 128-bit KEY registers defined as DATA3-DATA0, KEY3-KEY0 (KEYL) and KEY7-KEY4 (KEYH). In AES128 mode, the 128-bit key is read from KEYL, while both KEYH and KEYL are used in AES256 mode. See Figure 27.2 (p. 389) . The figure presents the key byte order for 256-bit keys. In 128-bit mode a_{16} represents the first byte of the 128-bit key.

It is important to note the order of the individual bytes in the key and state in relation to how they are defined in the Advanced Encryption Standard (FIPS-197).

Figure 27.2. AES Data and Key Orientation as Defined in the Advanced Encryption Standard

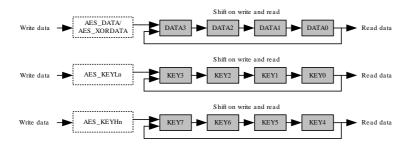


The registers DATA3-DATA0, are not memory mapped directly, but can be written/read by accessing AES_DATA or AES_XORDATA. The same applies for the key registers, KEY3-KEY0 which are accessed through AES_KEYLn (n=A, B, C or D), while KEY7-KEY4 are accessed through KEYHn



(n=A, B, C or D). Writing DATA3-DATA0 is then done through 4 consecutive writes to AES_DATA (or AES_XORDATA), starting with the word which is to be written to DATA0. For each write, the words will be word wise barrel shifted towards the least significant word. Accessing the KEY registers are done in the same fashion through KEYLn and KEYHn. See Figure 27.3 (p. 390). Note that KEYHA, KEYHB, KEYHC and KEYHD are really the same register, just mapped to four different addresses. You can then choose freely which of these addresses you want to use to update the KEY7-KEY4 registers. The same principle applies to the KEYLn registers. Mapping the same registers to multiple addresses like this, allows the DMA controller to write a full 256-bit key in one sweep, when incrementing the address between each word write.

Figure 27.3. AES Data and Key Register Operation



27.3.2.1 Key Buffer

When encrypting multiple blocks of data in a row, the PlainKey must be written to the key register between each encryption, since the contents of the key registers will be turned into the CipherKey during the encryption. The opposite applies when decrypting, where you have to re-supply the CipherKey between each block. However, in AES128 mode, KEY4-KEY7 can be used as a buffer register, to hold an extra copy of the KEY3-KEY0 registers. When KEYBUFEN is set in AES_CTRL, the contents of KEY7-KEY4 are copied to KEY3-KEY0, when an encryption/decryption is started. This eliminates the need for re-loading the KEY for every encrypted/decrypted block when running in AES128 mode.

27.3.2.2 Data Write XOR

The AES module contains an array of XOR gates connected to the DATA registers, which can be used during a data write to XOR the existing contents of the registers with the new data written. To use the XOR function, the data must be written to AES_XORDATA location.

Reading data from AES XORDATA is equivalent to reading data from AES DATA.

27.3.2.3 Start on Data Write

The AES module can be configured to start an encryption/decryption when the new data has been written to AES_DATA and/or AES_XORDATA. A 2-bit counter is incremented each time the AES_DATA or AES_XORDATA registers are written. This counter indicates which data word is written. If DATASTART/XORSTART in AES_CTRL is set, an encryption will start each time the counter overflows (DATA3 is written). Writing to the AES_CTRL register will reset the counter to 0.

27.3.3 Interrupt Request

The DONE interrupt flag is set when an encryption/ decryption has finished.

27.3.4 DMA Request

The AES module has 4 DMA requests which are all set on a finished encryption/decryption and cleared on the following conditions:



- DATAWR: Cleared on a AES_DATA write or AES_CTRL write
- XORDATAWR: Cleared on a AES_XORDATA write or AES_CTRL write
- DATARD: Cleared on a AES_DATA read or AES_CTRL write
- KEYWR: Cleared on a AES_KEYHn write or AES_CTRL write

27.3.5 Block Chaining Example

Example 27.1 (p. 391) below illustrates how the AES module could be configured to perform Cipher Block Chaining with 128-bit keys.

Example 27.1. AES Cipher Block Chaining

- 1. Configure module to encryption, key buffer enabled and XORSTART in AES_CTRL.
- 2. Write 128-bit initialization vector to AES_DATA, starting with least significant word.
- 3. Write PlainKey to AES_KEYHn, starting with least significant word.
- 4. Write PlainText to AES_XORDATA, starting with least significant word. Encryption will be started when the DATA3 is written. KEYH (PlainKey) will be copied to KEYL before encryption starts.
- 5. When encryption is finished, read CipherText from AES_DATA, starting with least significant word.
- 6. Loop to step 4, if new PlainText is available.



27.4 Register Map

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

Offset	Name	Туре	Description
0x000	AES_CTRL	RW	Control Register
0x004	AES_CMD	W1	Command Register
0x008	AES_STATUS	R	Status Register
0x00C	AES_IEN	RW	Interrupt Enable Register
0x010	AES_IF	R	Interrupt Flag Register
0x014	AES_IFS	W1	Interrupt Flag Set Register
0x018	AES_IFC	W1	Interrupt Flag Clear Register
0x01C	AES_DATA	RW	DATA Register
0x020	AES_XORDATA	RW	XORDATA Register
0x030	AES_KEYLA	RW	KEY Low Register
0x034	AES_KEYLB	RW	KEY Low Register
0x038	AES_KEYLC	RW	KEY Low Register
0x03C	AES_KEYLD	RW	KEY Low Register
0x040	AES_KEYHA	RW	KEY High Register
0x044	AES_KEYHB	RW	KEY High Register
0x048	AES_KEYHC	RW	KEY High Register
0x04C	AES_KEYHD	RW	KEY High Register

27.5 Register Description

27.5.1 AES_CTRL - Control Register

Offset	Bit Position						
0x000	E 0						
Reset	0 0 0 0						
Access	N						
Name	XORSTART DATASTART KEYBUFEN AES256 DECRYPT						

				
Bit	Name	Reset	Access	Description
31:6	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
5	XORSTART	0	RW	AES_XORDATA Write Start
	Set this bit to start er	ncryption/decryption v	vhen DATA3 is wr	ritten through AES_XORDATA.
4	DATASTART	0	RW	AES_DATA Write Start
	Set this bit to start er	ncryption/decryption v	vhen DATA3 is wr	ritten through AES_DATA.
3	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
2	KEYBUFEN	0	RW	Key Buffer Enable
	Enable/disable key b	uffer in AES-128 mod	de.	
1	AES256	0	RW	AES-256 Mode
	Select AES-128 or A	ES-256 mode.		



Bit	Name	Reset	Access	Description	
	Value	Description			
	0	AES-128 mode			
	1	AES-256 mode			
0	DECRYPT	0	RW	Decryption/Encryption Mode	
	Select encryption	or decryption.			
	Value	Description			
	0	AES Encryption			
	1	AES Decryption			

27.5.2 AES_CMD - Command Register

Offset	Bit Position				·	
0x004	88 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	2 9	·Ω 4	8 2	-	0
Reset					0	0
Access					W1	W1
Name					STOP	START

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1	STOP	0	W1	Encryption/Decryption Stop
	Set to stop encryption	on/decryption.		
0	START	0	W1	Encryption/Decryption Start
	Set to start encryption	on/decryption.		

27.5.3 AES_STATUS - Status Register

Offset															Bi	t Pc	siti	on														
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	∞	7	9	2	4	ю	2	-	0
Reset																													•			0
Access																																~
Name																																RUNNING

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	RUNNING	0	R	AES Running
	This bit indicates th	at the AES module is r	unning an encryp	tion/decryption.



27.5.4 AES_IEN - Interrupt Enable Register

Offset															Bi	t Po	siti	on														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	11	10	6	8	7	9	2	4	က	2	-	0
Reset																																0
Access																																R
Name																																DONE

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure compa	tibility with futu	re devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	DONE	0	RW	Encryption/Decryption Done Interrupt Enable
	Enable/disable interrupt on	encryption/decrypti	on done.	

27.5.5 AES_IF - Interrupt Flag Register

Offset															Bi	t Pc	siti	on														
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	∞	7	9	2	4	ю	2	-	0
Reset																					-											0
Access																																~
Name																																DONE

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	DONE	0	R	Encryption/Decryption Done Interrupt Flag
	Set when an encryp	otion/decryption has fin	ished.	

27.5.6 AES_IFS - Interrupt Flag Set Register

Offset				,	,				,						Bi	it Po	siti	on														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	-	10	6	8	7	9	2	4	က	2	-	0
Reset				•	•				•								•															0
Access																																N N
Name																																DONE

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	DONE	0	W1	Encryption/Decryption Done Interrupt Flag Set
	Write to 1 to set en	cryption/decryption don	ne interrupt flag	



27.5.7 AES_IFC - Interrupt Flag Clear Register

Offset															Bi	t Po	siti	on														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	6	ω	7	9	2	4	က	2	-	0
Reset																													-			0
Access																																W N
Name																																DONE

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure compa	tibility with futu	re devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	DONE	0	W1	Encryption/Decryption Done Interrupt Flag Clear
	Write to 1 to clear encryptio	n/decryption done i	nterrupt flag	

27.5.8 AES_DATA - DATA Register

Offset		Bit Position																												
0x01C	33	30	28	27	56	22	24	23	22	24	20	19	1	16	15	4	13	12	1	9	6	80	7	9	2	4	က	2	-	0
Reset															0000000000															
Access															Ŗ ≷															
Name															DATA															
Rit	Nan	ne					Res	set			Δ	cces	35	D	escr	intid	on													

Bit	Name	Reset	Access	Description
31:0	DATA	0x00000000	RW	Data Access
	Access data through this re			

27.5.9 AES_XORDATA - XORDATA Register

Offset		Bit Position																														
0x020	31	31 2 2 <																														
Reset		000000000000000000000000000000000000000																														
Access																<u> </u>	^															
Name		XORDATA																														



Bit	Name	Reset	Access	Description								
31:0	XORDATA	0x00000000	RW	XOR Data Access								
	Access data with XOR function through this register.											

27.5.10 AES_KEYLA - KEY Low Register

Offset	Bit Position
0x030	30 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
Reset	00000000000000000000000000000000000000
Access	R W
Name	KEYLA

Bit	Name	Reset	Access	Description								
31:0	KEYLA	0x00000000	RW	Key Low Access A								
	Access the low key words through this register.											

27.5.11 AES_KEYLB - KEY Low Register

Offset		Bit Position																														
0x034	31	30	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	8	7	9	2	4	က	2	1	0
Reset																00000000	000000000															
Access																2	2															
Name																2 2 2	9 - - - -															

Bit	Name	Reset	Access	Description								
31:0	KEYLB	0x00000000	RW	Key Low Access B								
	Access the low key words through this register.											



27.5.12 AES_KEYLC - KEY Low Register

Offset				,	,			,							Bi	t Po	siti	on					,									
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	∞	7	9	2	4	က	2	-	0
Reset																00000000	000000000000000000000000000000000000000															
Access																<u> </u>	2															
Name																<u> </u>	2															
Bit	Na	me						Re	set			A	CCE	ess		De	scri	iptic	on													
31:0	KE'	YLC						0x0	0000	0000)	R	W			Key	/ Lo	w A	CC	ess (С											
	Acc	ess	the I	low k	кеу ч	word	ds th	nrou	gh th	is re	giste	er.																				

27.5.13 AES_KEYLD - KEY Low Register

Offset										,			E	Bit F	Posit	ion														
0x03C	33	29	28	27	56	22	24	23	22	21	20	19	2 7	. 4	15	4	13	12	7	10	တ	∞	7	9	2	4	က	7	-	0
Reset															00000000x0															
Access															RW															
Name															KEYLD															
Bit	Name	9					Re	set			A	cce	ss	0)escr	iptic	on													
31:0	KEYLI)					0x0	0000	0000		R'	W		K	ey Lo	w A	CC	ess I	D											
	Acces	s the	low k	кеу ч	word	ds th	nroug	gh thi	is re	giste	r.																			

27.5.14 AES_KEYHA - KEY High Register

Offset															Bi	t Pc	siti	on														
0x040	31	30	29	28	27	26	22	24	23	22	21	70	19	18	17	16	15	14	13	12	7	10	တ	∞	7	9	2	4	က	2	-	0
Reset																000000000000000000000000000000000000000	000000000															
Access																2	<u>}</u>															
Name																\ \ \ \	K L L															



Bit	Name	Reset	Access	Description
31:0	KEYHA	0x00000000	RW	Key High Access A
	Access the high ke	ey words through this regis	ter.	

27.5.15 AES_KEYHB - KEY High Register

Offset	Bit Position
0x044	3 3 4 5 5 6 7 7 8 8 8 8 8 7 7 9 8 8 8 8 8 8 8 8 8 8
Reset	00000000000000000000000000000000000000
Access	R&
Name	KEY HB

Bit	Name	Reset	Access	Description
31:0	KEYHB	0x0000000	RW	Key High Access B
	Access the high key words	through this registe	er.	

27.5.16 AES_KEYHC - KEY High Register

Offset	Bit Position
0x048	1 1
Reset	00000000000000000000000000000000000000
Access	RW.
Name	KEYHC

Bit	Name	Reset	Access	Description
31:0	KEYHC	0x00000000	RW	Key High Access C
	Access the high key words	through this regist	er.	



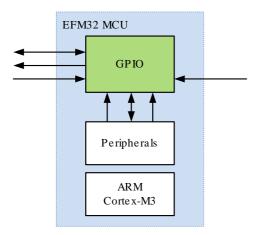
27.5.17 AES_KEYHD - KEY High Register

Offset													·		Bi	t Po	ositi	on														
0x04C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	1	10	6	8	7	9	2	4	က	2	-	0
Reset																0000000	000000000															
Access																Ž	Š Y															
Name																2	KEYHU															
Bit	Na	me						Re	set			А	CC	ess		De	escr	iptio	on													
31:0	KE'	YHD						0x0	000	0000	0	R	W			Ke	y Hiç	gh A	СС	ess	D											
	Acc	ess	the	hig	h key	wo	rds	throu	ıgh t	his r	egist	er.																				



28 GPIO - General Purpose Input/Output





Quick Facts

What?

The GPIO (General Purpose Input/Output) is used for pin configuration and direct pin manipulation and sensing 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 several different locations, thus solving congestion issues that may arise with multiple functions on the same pin. Fully asynchronous interrupts can also be generated from any pin.

28.1 Introduction

In the EFM32G devices the General Purpose Input/Output (GPIO) pins are organized into ports with up to 16 pins each. These pins can individually be configured as either an output or input. More advanced configurations like open-drain, filtering and drive strength can also be configured individually for the pins. The GPIO pins can also be overridden by peripheral pin connections, like Timer PWM outputs or USART communication, which can be routed to several locations on the device. The GPIO supports up to 16 asynchronous external pin interrupts, which enables interrupts from any pin on the device. Also, the input value of a pin can be routed through the Peripheral Reflex System to other peripherals.

28.2 Features

- · Individual configuration for each pin
 - Tristate (reset state)
 - Push-pull
 - Open-drain
 - · Pull-up resistor
 - · Pull-down resistor
 - Four drive strength modes
 - HIGH
 - STANDARD
 - LOW
 - LOWEST



- · Glitch suppression input filter.
- Analog connection to e.g. ADC.
- Alternate functions (e.g. peripheral outputs and inputs)
 - Routed to several locations on the device
 - Pin connections can be enabled individually
 - Output data can be overridden by peripheral
 - Output enable can be overridden by peripheral
- · Toggle, set and clear registers for output data
- Dedicated data input register (read-only)
- Interrupts
 - 2 interrupt lines from up to 16 pending sources
 - All GPIO pins are selectable
 - · Separate enable, status, set and clear registers
 - Asynchronous sensing
 - · Rising, falling or both edges
 - Wake up from EM0-EM3
- Peripheral Reflex System producer
 - · All GPIO pins are selectable
- · Configuration lock functionality to avoid accidental changes

28.3 Functional Description

An overview of the GPIO module is shown in Figure 28.1 (p. 402). 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 bits 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 is disabled for all pins on the device, except for debug pins. To use a pin, the port GPIO_Px_MODEL/GPIO_Px_MODEH registers 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 Section 28.3.1 (p. 402). When the port is either 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.

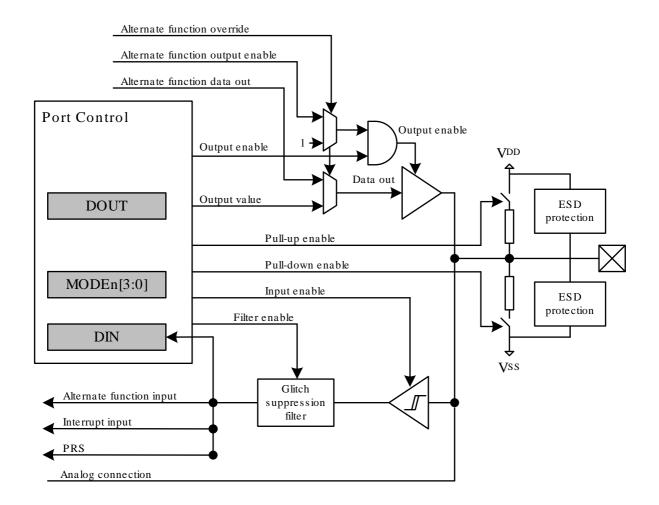
The DOUT value can be changed in 4 different ways

- Writing to the GPIO_Px_DOUT register.
- Writing a 1 to a bit in the GPIO_Px_DOUTSET register sets the corresponding DOUT bit
- Writing a 1 to a bit in the GPIO_Px_DOUTCLR register clears the corresponding DOUT bit
- Writing a 1 to a bit in the GPIO_Px_DOUTTGL register toggles the corresponding DOUT bit

Reading the GPIO_Px_DOUT register will return its contents. Reading the GPIO_Px_DOUTSET, GPIO_Px_CLR or GPIO_Px_TGL will return 0.



Figure 28.1. Pin Configuration



28.3.1 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 28.1 (p. 402) shows the available configurations.

Table 28.1. Pin Configuration

MODEn	Input	Output	DOUT	Pull- down	Pull- up	Alt. strength	Input Filter	Description
0b0000	Disabled	Disabled	0					Input disabled
			1		On			Input disabled with pull-up
0b0001	Enabled		0					Input enabled
			1				On	Input enabled with filter
0b0010			0	On				Input enabled with pull-down
			1		On			Input enabled with pull-up

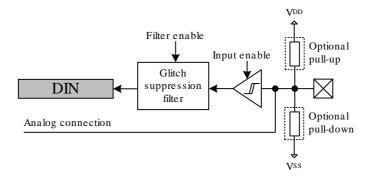


MODEn	Input	Output	DOUT	Pull- down	Pull- up	Alt. strength	Input Filter	Description
0b0011			0	On			On	Input enabled with pull-down and filter
			1		On		On	Input enabled with pull-up and filter
0b0100		Push-pull	х					Push-pull
0b0101			х			On		Push-pull with alt. drive strength
0b0110		Open	х					Open-source
0b0111		Source (Wired-OR)	х	On				Open-source with pull-down
0b1000	-	Open Drain	х					Open-drain
0b1001	-	(Wired- AND)	х				On	Open-drain with filter
0b1010			х		On			Open-drain with pull-up
0b1011			х		On		On	Open-drain with pull-up and filter
0b1100			х			On		Open-drain with alt. drive strength
0b1101			х			On	On	Open-drain with alt. drive strength and filter
0b1110			х		On	On		Open-drain with alt. drive strength and pull-up
0b1111			х		On	On	On	Open-drain with alt. drive strength, pull-up and filter

MODEn determines which mode the pin is in at a given time. Setting MODEn to 0b0000 disables the pin, reducing power consumption to a minimum. When the output driver is disabled, the pin can be used as a connection for an analog module (e.g. ADC). Input is enabled by setting MODEn to any value other than 0b0000. The pull-up, pull-down and filter function can optionally be applied to the input, see Figure 28.2 (p. 403) .

The internal pull-up resistance, R_{PU} , and pull-down resistance, R_{PD} , are defined in the device datasheet. When the filter is enabled it suppresses glitches with pulse widths as defined by the parameter $t_{IOGLITCH}$ in the device datasheet.

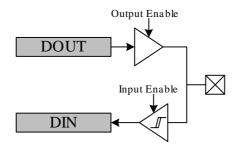
Figure 28.2. Tristated Output with Optional Pull-up or Pull-down



When MODEn=0b0100 or MODEn=0b0101, the pin operates in push-pull mode. In this mode, the pin is driven either high or low, dependent on the value of GPIO_Px_DOUT. The push-pull configuration is shown in Figure 28.3 (p. 404).



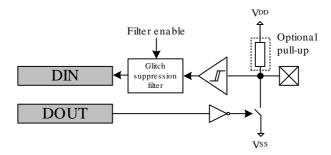
Figure 28.3. Push-Pull Configuration



When MODEn is 0110 or 0111, the pin operates in open-source mode, the latter with a pull-down resistor. When driving a high value in open-source mode, the pull-down is disconnected to save power.

For the remaining MODEn values, i.e. MODEn >= 1000, the pin operates in open-drain mode as shown in Figure 28.4 (p. 404). In open-drain mode, the pin can have an input filter, a pull-up, different driver strengths or any combination of these. When driving a low value in open-drain mode, the pull-up is disconnected to save power.

Figure 28.4. Open-drain



When MODEn=0b0101 or 0b11xx, the output driver uses the drive strength specified in DRIVEMODE in GPIO_Px_CTRL. In all other output modes, the drive strength is set to STANDARD.

28.3.1.1 Configuration Lock

GPIO_Px_MODEL, GPIO_Px_MODEH, GPIO_Px_CTRL, GPIO_Px_PINLOCKN, GPIO_EXTIPSELL, GPIO_EXTIPSELH, GPIO_INSENSE and GPIO_ROUTE can be locked by writing any other value than 0xA534 to GPIO_LOCK. Writing the value 0xA534 to the GPIOx_LOCK register unlocks the configuration registers.

In addition to configuration lock, GPIO_Px_MODEL, GPIO_Px_MODEH, GPIO_Px_DOUT, GPIO_Px_DOUTSET, GPIO_Px_DOUTCLR, and GPIO_Px_DOUTTGL can be locked individually for each pin by clearing the corresponding bit in GPIO_Px_PINLOCKN. Bits in the GPIO_Px_PINLOCKN register can only be cleared, they are set high again after reset.

28.3.2 Alternate Functions

Alternate functions are connections to pins from Timers, USARTs etc. These modules contain route registers, where the pin connections are enabled. In addition, these registers contain a location bit field, which configures which pins the outputs of that module will be connected to if they are enabled. If an alternate signal output is enabled for a pin and output is enabled for the pin, the alternate



function's output data and output enable signals override the data output and output enable signals from the GPIO. However, the pin configuration stays as set in GPIO_Px_MODEL, GPIO_Px_MODEH and GPIO_Px_DOUT registers. I.e. the pin configuration must be set to output enable in GPIO for a peripheral to be able to use the pin as an output.

It is possible, but not recommended to select two or more peripherals as output on the same pin. These signals will then be OR'ed together. However, TIMER CCx and CDTIx outputs, which are routed as alternate functions, have priority, and will never be OR'ed with other alternate functions. The reader is referred to the pin map section of the device datasheet for more information on the possible locations of each alternate function and any priority settings.

28.3.2.1 Serial Wire Debug Port Connection

The SW Debug Port is routed as an alternate function and 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 SWDIOPEN and SWCLKPEN bits in GPIO_ROUTE to 0.

WARNING: 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 default state as enabled. 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 halt the device after a reset before the pins are disabled.

The Serial Wire Viewer Output pin (SWO) can be enabled by setting the SWOPEN bit in GPIO_ROUTE. This bit can also be routed to alternate locations by configuring the LOCATION bitfield in GPIO_ROUTE.

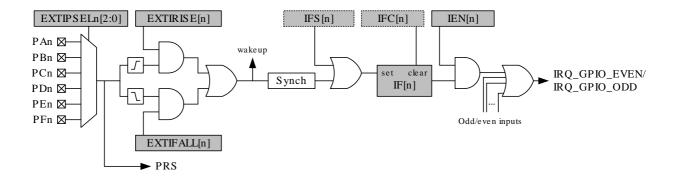
28.3.2.2 Analog Connections

When using the GPIO pin for analog functionality, it is recommended to disable the digital output and set the MODEn in GPIO_Px_MODEL/GPIO_Px_MODEH equal to 0b0000 to disable the input sense and pull resistors.

28.3.3 Interrupt Generation

The GPIO can generate an interrupt from the input of any GPIO pin on a device. The interrupts have asynchronous sense capability, enabling wake-up from energy modes as low as EM3, see Figure 28.5 (p. 405).

Figure 28.5. Pin n Interrupt Generation



All pins with the same pin number (n) are grouped together to trigger one interrupt flag (EXT[n] in GPIO_IF). The EXTIPSELn[2:0] bits in GPIO_EXTIPSELL or GPIO_EXTIPSELH select which port will trigger the interrupt flag. The GPIO_EXTIRISE[n] and GPIO_EXTIFALL[n] registers enables 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,



while the odd is triggered by odd flags. The interrupt flags can be set and cleared by software by writing the GPIO_IFS and GPIO_IFC registers, see Example 28.1 (p. 406). Since the external interrupts are asynchronous, they are sensitive to noise. To increase noise tolerance, the MODEL and MODEH fields in the GPIO_Px_MODEL and GPIO_Px_MODEH registers, respectively, should be set to include filtering for pins that have external interrupts enabled.

Example 28.1. GPIO Interrupt Example

Setting EXTIPSEL3 in GPIO_EXTIPSELL to 2 (Port C) and setting the GPIO_EXTIRISE[3] bit, the interrupt flag EXT[3] in GPIO_IF will be triggered by a rising edge on pin 3 on PORT C. If EXT[3] in GPIO_IEN is set as well, a interrupt request will be sent on IRQ_GPIO_ODD.

28.3.4 Output to PRS

All pins with the same pin number (n) are grouped together to form one PRS producer output, giving a total of 16 outputs to the PRS. The port on which the output n should be taken is selected by the EXTIPSELn[3:0] bits in the GPIO_EXTIPSELL or the GPIO_EXTIPSELH registers.

28.3.5 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 HFCORECLK. Consequently, when a pin changes state, the change will have propagated to GPIO_Px_DIN after 2 positive HFCORECLK edges, or maximum 2 HFCORECLK cycles.

Synchronization (also running on the HFCORECLK) is also added for interrupt input. The input to the PRS generation is also synchronized, but these flip-flops run on the HFPERCLK. To save power when the external interrupts or PRS generation is not used, the synchronization flip-flops for these can be turned off by clearing the INTSENSE or PRSSENSE, respectively, in GPIO_INSENSE register.

Note

To use the GPIO, the GPIO clock must first be enabled in CMU_HFPERCLKEN0. Setting this bit enables the HFCORECLK and the HFPERCLK for the GPIO. HFCORECLK is used for updating registers, while HFPERCLK is only used to synchronize PRS and interrupts. The PRS and interrupt synchronization can also be disabled through GPIO_INSENSE, if these are not used.



28.4 Register Map

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

Offset	Name	Туре	Description
0x000	GPIO_PA_CTRL	RW	Port Control Register
0x004	GPIO_PA_MODEL	RW	Port Pin Mode Low Register
0x008	GPIO_PA_MODEH	RW	Port Pin Mode High Register
0x00C	GPIO_PA_DOUT	RW	Port Data Out Register
0x010	GPIO_PA_DOUTSET	W1	Port Data Out Set Register
0x014	GPIO_PA_DOUTCLR	W1	Port Data Out Clear Register
0x018	GPIO_PA_DOUTTGL	W1	Port Data Out Toggle Register
0x01C	GPIO_PA_DIN	R	Port Data In Register
0x020	GPIO_PA_PINLOCKN	RW	Port Unlocked Pins Register
0x024	GPIO_PB_CTRL	RW	Port Control Register
0x028	GPIO_PB_MODEL	RW	Port Pin Mode Low Register
0x02C	GPIO_PB_MODEH	RW	Port Pin Mode High Register
0x030	GPIO_PB_DOUT	RW	Port Data Out Register
0x034	GPIO_PB_DOUTSET	W1	Port Data Out Set Register
0x038	GPIO_PB_DOUTCLR	W1	Port Data Out Clear Register
0x03C	GPIO_PB_DOUTTGL	W1	Port Data Out Toggle Register
0x040	GPIO_PB_DIN	R	Port Data In Register
0x044	GPIO_PB_PINLOCKN	RW	Port Unlocked Pins Register
0x048	GPIO_PC_CTRL	RW	Port Control Register
0x04C	GPIO_PC_MODEL	RW	Port Pin Mode Low Register
0x050	GPIO_PC_MODEH	RW	Port Pin Mode High Register
0x054	GPIO_PC_DOUT	RW	Port Data Out Register
0x058	GPIO_PC_DOUTSET	W1	Port Data Out Set Register
0x05C	GPIO_PC_DOUTCLR	W1	Port Data Out Clear Register
0x060	GPIO_PC_DOUTTGL	W1	Port Data Out Toggle Register
0x064	GPIO_PC_DIN	R	Port Data In Register
0x068	GPIO_PC_PINLOCKN	RW	Port Unlocked Pins Register
0x06C	GPIO_PD_CTRL	RW	Port Control Register
0x070	GPIO_PD_MODEL	RW	Port Pin Mode Low Register
0x074	GPIO_PD_MODEH	RW	Port Pin Mode High Register
0x078	GPIO_PD_DOUT	RW	Port Data Out Register
0x07C	GPIO_PD_DOUTSET	W1	Port Data Out Set Register
0x080	GPIO_PD_DOUTCLR	W1	Port Data Out Clear Register
0x084	GPIO_PD_DOUTTGL	W1	Port Data Out Toggle Register
0x088	GPIO_PD_DIN	R	Port Data In Register
0x08C	GPIO_PD_PINLOCKN	RW	Port Unlocked Pins Register
0x090	GPIO_PE_CTRL	RW	Port Control Register
0x094	GPIO_PE_MODEL	RW	Port Pin Mode Low Register
0x098	GPIO_PE_MODEH	RW	Port Pin Mode High Register
0x09C	GPIO_PE_DOUT	RW	Port Data Out Register



Offset	Name	Туре	Description								
0x0A0	GPIO_PE_DOUTSET	W1	Port Data Out Set Register								
0x0A4	GPIO_PE_DOUTCLR	W1	Port Data Out Clear Register								
0x0A8	GPIO_PE_DOUTTGL	W1	Port Data Out Toggle Register								
0x0AC	GPIO_PE_DIN	R	Port Data In Register								
0x0B0	GPIO_PE_PINLOCKN	RW	Port Unlocked Pins Register								
0x0B4	GPIO_PF_CTRL	RW	Port Control Register								
0x0B8	GPIO_PF_MODEL	RW	Port Pin Mode Low Register								
0x0BC	GPIO_PF_MODEH	RW	Port Pin Mode High Register								
0x0C0	GPIO_PF_DOUT	RW	Port Data Out Register								
0x0C4	GPIO_PF_DOUTSET	W1	Port Data Out Set Register								
0x0C8	GPIO_PF_DOUTCLR	W1	Port Data Out Clear Register								
0x0CC	GPIO_PF_DOUTTGL	W1	Port Data Out Toggle Register								
0x0D0	GPIO_PF_DIN	R	Port Data In Register								
0x0D4	GPIO_PF_PINLOCKN	RW	Port Unlocked Pins Register								
0x100	GPIO_EXTIPSELL	RW	External Interrupt Port Select Low Register								
0x104	GPIO_EXTIPSELH	RW	External Interrupt Port Select High Register								
0x108	GPIO_EXTIRISE	RW	External Interrupt Rising Edge Trigger Register								
0x10C	GPIO_EXTIFALL	RW	External Interrupt Falling Edge Trigger Register								
0x110	GPIO_IEN	RW	Interrupt Enable Register								
0x114	GPIO_IF	R	Interrupt Flag Register								
0x118	GPIO_IFS	W1	Interrupt Flag Set Register								
0x11C	GPIO_IFC	W1	Interrupt Flag Clear Register								
0x120	GPIO_ROUTE	RW	I/O Routing Register								
0x124	GPIO_INSENSE	RW	Input Sense Register								
0x128	GPIO_LOCK	RW	Configuration Lock Register								

28.5 Register Description

28.5.1 GPIO_Px_CTRL - Port Control Register

Offset															Bi	t Pc	siti	on														
0x000	31	30	29	28	27	56	25	24	23	22	21	20	19	8	17	16	15	4	13	12	7	10	စ	80	7	9	2	4	က	2	-	0
Reset																															2	S S
Access																															2	<u> </u>
Name																																ם סואום אואס

Bit	Name	Reset	Access	Description
31:2	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
1:0	DRIVEMODE	0x0	RW	Drive Mode Select
	Select drive mode for	all pins on port conf	igured with altern	ate drive strength.

Value	Mode	Description
0	STANDARD	6 mA drive current



Bit	Name	Reset Acces	s Description
	Value	Mode	Description
	1	LOWEST	0.1 mA drive current
	2	HIGH	20 mA drive current
	3	LOW	1 mA drive current

28.5.2 GPIO_Px_MODEL - Port Pin Mode Low Register

Offset				Bit Po	sition			
0x004	30 29 28	27 26 25 24	23 22 21 20 20	18 17 17 16	51 4 51 21	11 10 8	r 8 2 4	e 2 t 0
Reset	0×0	0×0	0×0	0×0	0×0	0×0	0×0	0×0
Access	RW	RW	RW	RW	RW	RW	RW	RW
Name	MODE7	MODE6	MODE5	MODE4	MODE3	MODE2	MODE1	MODE

Bit	Name	Reset	Access	Description
31:28	MODE7	0x0	RW	Pin 7 Mode
	Configure mode for pin 7.	Enumeration is equ	al to MODE0.	
27:24	MODE6	0x0	RW	Pin 6 Mode
	Configure mode for pin 6.	Enumeration is equ	al to MODE0.	
23:20	MODE5	0x0	RW	Pin 5 Mode
	Configure mode for pin 5.	Enumeration is equ	al to MODE0.	
19:16	MODE4	0x0	RW	Pin 4 Mode
	Configure mode for pin 4.	Enumeration is equ	al to MODE0.	
15:12	MODE3	0x0	RW	Pin 3 Mode
	Configure mode for pin 3.	Enumeration is equ	al to MODE0.	
11:8	MODE2	0x0	RW	Pin 2 Mode
	Configure mode for pin 2.	Enumeration is equ	al to MODE0.	
7:4	MODE1	0x0	RW	Pin 1 Mode
	Configure mode for pin 1.	Enumeration is equ	al to MODE0.	
3:0	MODE0	0x0	RW	Pin 0 Mode
	Configure mode for pin 0.			

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	PUSHPULLDRIVE	Push-pull output with drive-strength set by DRIVEMODE
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	WIREDANDDRIVE	Open-drain output with drive-strength set by DRIVEMODE
13	WIREDANDDRIVEFILTER	Open-drain output with filter and drive-strength set by DRIVEMODE
14	WIREDANDDRIVEPULLUP	Open-drain output with pullup and drive-strength set by DRIVEMODE
15	WIREDANDDRIVEPULLUPFILTER	Open-drain output with filter, pullup and drive-strength set by DRIVEMODE



28.5.3 GPIO_Px_MODEH - Port Pin Mode High Register

Offset															Bi	t Po	siti	on														
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	1	10	6	8	7	9	2	4	3	2	-	0
Reset	000									2	SX S			0x0				0x0				000					3		000			
Access	RW W					RW							ΚW		R.					Z Š				RW								
Name	MODE15							27	-			MODE12				MODE11 F				MODE10 RV					NO DE		MODE8 R					

				,					
Bit	Name	Reset	Access	s Description					
31:28	MODE15	0x0	RW	Pin 15 Mode					
	Configure mode t	for pin 15. Enumeration is	equal to MOD	DE8.					
27:24	MODE14	0x0	RW	Pin 14 Mode					
	Configure mode f	for pin 14. Enumeration is	equal to MOD	DE8.					
23:20	MODE13	0x0	RW	Pin 13 Mode					
	Configure mode f	for pin 13. Enumeration is	equal to MOD	DE8.					
19:16	MODE12	0x0	RW	Pin 12 Mode					
	Configure mode f	for pin 12. Enumeration is	equal to MOD	DE8.					
15:12	MODE11	0x0	RW	Pin 11 Mode					
	Configure mode to	for pin 11. Enumeration is	equal to MOD	DE8.					
11:8	MODE10	0x0	RW	Pin 10 Mode					
	Configure mode to	for pin 10. Enumeration is	equal to MOD	DE8.					
7:4	MODE9	0x0	RW	Pin 9 Mode					
	Configure mode to	for pin 9. Enumeration is e	qual to MODI	DE8.					
3:0	MODE8	0x0	RW	Pin 8 Mode					
	Configure mode f	for pin 8.							
	Value	Mode	1	Description					
	0	DISABLED	I	Input disabled. Pullup if DOUT is set.					
	1	INPUT	I	Input enabled. Filter if DOUT is set					

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	PUSHPULLDRIVE	Push-pull output with drive-strength set by DRIVEMODE
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	WIREDANDDRIVE	Open-drain output with drive-strength set by DRIVEMODE
13	WIREDANDDRIVEFILTER	Open-drain output with filter and drive-strength set by DRIVEMODE
14	WIREDANDDRIVEPULLUP	Open-drain output with pullup and drive-strength set by DRIVEMODE
15	WIREDANDDRIVEPULLUPFILTER	Open-drain output with filter, pullup and drive-strength set by DRIVEMODE



28.5.4 GPIO_Px_DOUT - Port Data Out Register

Offset															t Po	siti	on															
0x00C	34	30	59	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	=	10	6	80	7	9	2	4	က	2	-	0
Reset																								000	000000							
Access																								7	<u>}</u>							
Name																								Ē	500							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	DOUT	0x0000	RW	Data Out
	Data output on port.			

28.5.5 GPIO_Px_DOUTSET - Port Data Out Set Register

Offset															Bi	t Po	siti	on														
0x010	31	30	53	28	27	26	22	24	23	22	21	20	19	18	17	16	15	4	13	12	7	9	6	∞	7	9	2	4	ო	7	-	0
Reset																								0000	OXOOOXO							
Access																								787	>							
Name																								E G E	DO01 3E1							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	mpatibility with	future devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	DOUTSET	0x0000	W1	Data Out Set
	Write bits to 1 to set	corresponding bits in	GPIO_Px_DOU	T. Bits written to 0 will have no effect.

28.5.6 GPIO_Px_DOUTCLR - Port Data Out Clear Register

Offset															Bi	t Po	siti	on														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	1	10	6	8	7	9	2	4	က	2	-	0
Reset	0000x0																															
Access																								7	>							
Name																								2 E	DOOLCER							



Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	mpatibility with fo	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	DOUTCLR	0x0000	W1	Data Out Clear
	Write bits to 1 to clear	r corresponding bits in	n GPIO_Px_DO	JT. Bits written to 0 will have no effect.

28.5.7 GPIO_Px_DOUTTGL - Port Data Out Toggle Register

Offset															Bi	t Po	siti	on														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	ω	7	9	2	4	ю	2	-	0
Reset	0000x0																															
Access																																
Name																								<u> </u>	DOULLGE							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure compa	ntibility with futu	re devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	DOUTTGL	0x0000	W1	Data Out Toggle
	Write bits to 1 to toggle corr	esponding bits in C	GPIO_Px_DOU	T. Bits written to 0 will have no effect.

28.5.8 GPIO_Px_DIN - Port Data In Register

Offset									,						Bi	t Pc	siti	on									,					
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	1	10	6	8	7	9	2	4	3	2	1	0
Reset	00000000																															
Access																								۵	Ľ							
Name																								Ž								

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure com	patibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	DIN	0x0000	R	Data In
	Port data input.			



28.5.9 GPIO_Px_PINLOCKN - Port Unlocked Pins Register

Offset															Bi	t Po	siti	on														
0x020	31	30	53	28	27	26	52	24	23	22	21	20	19	18	17	16	15	41	13	12	1	10	6	80	7	9	2	4	က	7	-	0
Reset																								L	OXFFFF							
Access																								i	Š Y							
Name																									PINLOCKN							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	mpatibility with	future devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	PINLOCKN	0xFFFF	RW	Unlocked Pins
	Shows unlocked pin	s in the port. To lock p	in n, clear bit n	. The pin is then locked until reset.

28.5.10 GPIO_EXTIPSELL - External Interrupt Port Select Low Register

Offset					,				,			,			Bi	t Pc	siti	on	·													
0x100	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	80	7	9	2	4	9	2	-	0
Reset			0x0				0x0				0x0				0x0				0x0				0x0				0x0	,			0x0	
Access			RW				RW				RW				RW.				RW				RW				RW				RW W	
Name			EXTIPSEL7				EXTIPSEL6				EXTIPSEL5				EXTIPSEL4				EXTIPSEL3				EXTIPSEL2				EXTIPSEL1				EXTIPSEL0	

Bit	Name	Reset	Access	S Description
31	Reserved	To ensure co	ompatibility with	n future devices, always write bits to 0. More information in Section 2.1 (p. 3)
30:28	EXTIPSEL7	0x0	RW	External Interrupt 7 Port Select
	Select input por	t for external interrupt 7.		
	Value	Mode	Г	Description
	0	PORTA	F	Port A pin 7 selected for external interrupt 7
	1	PORTB	F	Port B pin 7 selected for external interrupt 7
	2	PORTC	F	Port C pin 7 selected for external interrupt 7
	3	PORTD	F	Port D pin 7 selected for external interrupt 7
	4	PORTE	F	Port E pin 7 selected for external interrupt 7
	5	PORTF	F	Port F pin 7 selected for external interrupt 7
27	Reserved	To ensure co	ompatibility with	n future devices, always write bits to 0. More information in Section 2.1 (p. 3)

EXTIPSEL6 0x0 RW External Interrupt 6 Port Select

Select input port for external interrupt 6.

26:24

Value	Mode	Description
0	PORTA	Port A pin 6 selected for external interrupt 6
1	PORTB	Port B pin 6 selected for external interrupt 6
2	PORTC	Port C pin 6 selected for external interrupt 6
3	PORTD	Port D pin 6 selected for external interrupt 6
4	PORTE	Port E pin 6 selected for external interrupt 6
5	PORTF	Port F pin 6 selected for external interrupt 6



Bit	Name	Reset	Access	Description
23	Reserved	To ensure co	ompatibility with	future devices, always write bits to 0. More information in Section 2.1 (p. 3
22:20	EXTIPSEL5	0x0	RW	External Interrupt 5 Port Select
	Select input po	ort for external interrupt 5.		
	Value	Mode		escription
	0	PORTA		ort A pin 5 selected for external interrupt 5
	1	PORTB		ort B pin 5 selected for external interrupt 5
	2	PORTC		ort C pin 5 selected for external interrupt 5
	3	PORTD		ort D pin 5 selected for external interrupt 5
	4	PORTE		ort E pin 5 selected for external interrupt 5
	5	PORTF	P	ort F pin 5 selected for external interrupt 5
19	Reserved	To ensure co	ompatibility with	future devices, always write bits to 0. More information in Section 2.1 (p.
 18:16	EXTIPSEL4	0x0	RW	External Interrupt 4 Port Select
10:10		ort for external interrupt 4.	KVV	External interrupt 4 Port Select
	Value	Mode	D	escription
	0	PORTA	P	ort A pin 4 selected for external interrupt 4
	1	PORTB	Pi	ort B pin 4 selected for external interrupt 4
	2	PORTC	Pr	ort C pin 4 selected for external interrupt 4
	3	PORTD	P	ort D pin 4 selected for external interrupt 4
	4	PORTE	P	ort E pin 4 selected for external interrupt 4
	5	PORTF	P	ort F pin 4 selected for external interrupt 4
	EXTIPSEL3	0x0	RW	External Interrupt 3 Port Select
	EXTIPSEL3 Select input po	0x0 ort for external interrupt 3.	RW	External Interrupt 3 Port Select
	EXTIPSEL3 Select input po	0x0 ort for external interrupt 3.	RW D	escription
	EXTIPSEL3 Select input po Value 0	0x0 ort for external interrupt 3. Mode PORTA	RW D	External Interrupt 3 Port Select escription ort A pin 3 selected for external interrupt 3
	EXTIPSEL3 Select input po Value 0 1	Ox0 ort for external interrupt 3. Mode PORTA PORTB	RW D	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3
	EXTIPSEL3 Select input po Value 0	Ox0 ort for external interrupt 3. Mode PORTA PORTB PORTC	RW D	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3
	EXTIPSEL3 Select input po Value 0 1	Ox0 ort for external interrupt 3. Mode PORTA PORTB	RW Di Pi	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3
	EXTIPSEL3 Select input po Value 0 1 2 3	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD	RW Di Pri Pri Pri Pri Pri Pri	External Interrupt 3 Port Select escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort D pin 3 selected for external interrupt 3
15 14:12	EXTIPSEL3 Select input po Value 0 1 2 3 4	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF	RW Di Pri Pri Pri Pri Pri Pri Pri Pri	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort D pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3
14:12	EXTIPSEL3 Select input po Value 0 1 2 3 4 5	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF	RW Di Pri Pri Pri Pri Pri Pri Pri Pri	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort D pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3
14:12	EXTIPSEL3 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co	RW Do Pri Pri Pri Pri Pri Pri Pri Pri Pri Pri	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort D pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3
14:12	EXTIPSEL3 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2 Select input po	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co	RW Do Po Po Po Po Po Po RW	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort D pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 future devices, always write bits to 0. More information in Section 2.1 (p. 3 External Interrupt 2 Port Select
14:12	EXTIPSEL3 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2 Select input po Value	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co	RW D P P P P P P P P P P P P P P P P P P	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort D pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 future devices, always write bits to 0. More information in Section 2.1 (p. secription
14:12	EXTIPSEL3 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2 Select input po	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co	RW DO PO PO PO PO PO PO PO PO PO PO PO PO PO	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort D pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 future devices, always write bits to 0. More information in Section 2.1 (p. 3 External Interrupt 2 Port Select escription ort A pin 2 selected for external interrupt 2
14:12	EXTIPSEL3 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2 Select input po Value 0	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co	RW Do Per Per Per Per Per Per Per Per Per Per	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort D pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 future devices, always write bits to 0. More information in Section 2.1 (p. 3 External Interrupt 2 Port Select escription ort A pin 2 selected for external interrupt 2 ort B pin 2 selected for external interrupt 2
14:12	EXTIPSEL3 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2 Select input po Value 0 1	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co Ox0 Ort for external interrupt 2. Mode PORTA PORTB	RW Do Pri Pri Pri Pri Pri Pri Pri Pri Pri Pri	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort D pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 future devices, always write bits to 0. More information in Section 2.1 (p. Section 2.1) External Interrupt 2 Port Select
14:12	EXTIPSEL3 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2 Select input po Value 0 1 2	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co Ox0 Ort for external interrupt 2. Mode PORTA PORTB PORTB PORTC	RW Di Pri Pri Pri Pri Pri Pri Pri Pri Pri Pr	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort D pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 future devices, always write bits to 0. More information in Section 2.1 (p. x External Interrupt 2 Port Select escription ort A pin 2 selected for external interrupt 2 ort B pin 2 selected for external interrupt 2 ort C pin 2 selected for external interrupt 2
14:12	EXTIPSEL3 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2 Select input po Value 0 1 2 3	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co Ox0 Ort for external interrupt 2. Mode PORTA PORTB PORTC PORTB PORTC PORTD PORTB PORTC PORTD	RW Di Pi Pi Pi Pi Pi Pi Pi Pi Pi Pi Pi Pi Pi	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort D pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 future devices, always write bits to 0. More information in Section 2.1 (p. External Interrupt 2 Port Select escription ort A pin 2 selected for external interrupt 2 ort B pin 2 selected for external interrupt 2 ort C pin 2 selected for external interrupt 2 ort D pin 2 selected for external interrupt 2
14:12	EXTIPSEL3 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2 Select input po Value 0 1 2 3 4 4 4 4	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co Ox0 Ort for external interrupt 2. Mode PORTA PORTB PORTC PORTB PORTC PORTB PORTC PORTB PORTC PORTB PORTC PORTD PORTC PORTD PORTE PORTF	RW Di Pri Pri Pri Pri Pri Pri Pri Pri Pri Pr	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort D pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 future devices, always write bits to 0. More information in Section 2.1 (p. 3 External Interrupt 2 Port Select escription ort A pin 2 selected for external interrupt 2 ort B pin 2 selected for external interrupt 2 ort C pin 2 selected for external interrupt 2 ort D pin 2 selected for external interrupt 2 ort E pin 2 selected for external interrupt 2 ort E pin 2 selected for external interrupt 2
14:12 11 10:8	EXTIPSEL3 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2 Select input po	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co Ox0 Ort for external interrupt 2. Mode PORTA PORTB PORTC PORTD PORTE PORTC PORTD PORTC PORTC PORTD PORTC PORTD PORTC PORTD PORTC PORTD PORTE PORTF	RW Do Proportibility with RW Do Proportibility with Proportion P	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 future devices, always write bits to 0. More information in Section 2.1 (p. 3 External Interrupt 2 Port Select escription ort A pin 2 selected for external interrupt 2 ort B pin 2 selected for external interrupt 2 ort C pin 2 selected for external interrupt 2 ort D pin 2 selected for external interrupt 2 ort E pin 2 selected for external interrupt 2 ort E pin 2 selected for external interrupt 2 ort F pin 2 selected for external interrupt 2 future devices, always write bits to 0. More information in Section 2.1 (p. 3 future devices, always write bits to 0. More information in Section 2.1 (p. 3)
14:12 11 10:8	EXTIPSEL3 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL1	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co Ox0 Ort for external interrupt 2. Mode PORTA PORTB PORTC PORTB PORTC PORTB PORTC PORTB PORTC PORTB PORTC PORTD PORTC PORTD PORTE PORTF	RW Di Pri Pri Pri Pri Pri Pri Pri Pri Pri Pr	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort D pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 future devices, always write bits to 0. More information in Section 2.1 (p. secreption ort A pin 2 selected for external interrupt 2 ort B pin 2 selected for external interrupt 2 ort C pin 2 selected for external interrupt 2 ort D pin 2 selected for external interrupt 2 ort E pin 2 selected for external interrupt 2 ort E pin 2 selected for external interrupt 2 ort F pin 2 selected for external interrupt 2 ort F pin 2 selected for external interrupt 2
14:12 11 10:8	EXTIPSEL3 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL1	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co Ox0 Ort for external interrupt 2. Mode PORTA PORTB PORTC PORTB PORTC PORTB PORTC PORTB PORTC PORTB PORTC PORTD PORTC PORTD PORTE PORTC Ox0	RW Do Proportion of the propo	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 future devices, always write bits to 0. More information in Section 2.1 (p. 3 External Interrupt 2 Port Select escription ort A pin 2 selected for external interrupt 2 ort B pin 2 selected for external interrupt 2 ort C pin 2 selected for external interrupt 2 ort D pin 2 selected for external interrupt 2 ort E pin 2 selected for external interrupt 2 ort E pin 2 selected for external interrupt 2 ort F pin 2 selected for external interrupt 2 future devices, always write bits to 0. More information in Section 2.1 (p. 3 future devices, always write bits to 0. More information in Section 2.1 (p. 3)
14:12 11 10:8	EXTIPSEL3 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL1 Select input po	Ox0 ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co Ox0 ort for external interrupt 2. Mode PORTA PORTB PORTC PORTB PORTC PORTB PORTC PORTB PORTC PORTD PORTC PORTD PORTC Ox0 ort for external interrupt 1.	RW Properties and the second	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 future devices, always write bits to 0. More information in Section 2.1 (p. External Interrupt 2 Port Select escription ort A pin 2 selected for external interrupt 2 ort B pin 2 selected for external interrupt 2 ort C pin 2 selected for external interrupt 2 ort E pin 2 selected for external interrupt 2 ort E pin 2 selected for external interrupt 2 ort F pin 2 selected for external interrupt 2 future devices, always write bits to 0. More information in Section 2.1 (p. External Interrupt 1 Port Select
14:12 11 10:8	EXTIPSEL3 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL1 Select input po Value O 1 2 3 4 5	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co Ox0 Ort for external interrupt 2. Mode PORTA PORTB PORTC PORTD PORTB PORTC PORTB PORTC PORTB PORTC PORTD PORTC PORTD PORTE PORTC NOTE PORTE PORTF To ensure co Ox0 Ox0 Ort for external interrupt 1. Mode	RW Description of the property of the propert	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort D pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 future devices, always write bits to 0. More information in Section 2.1 (p. External Interrupt 2 Port Select escription ort A pin 2 selected for external interrupt 2 ort B pin 2 selected for external interrupt 2 ort C pin 2 selected for external interrupt 2 ort E pin 2 selected for external interrupt 2 ort E pin 2 selected for external interrupt 2 ort F pin 2 selected for external interrupt 2 future devices, always write bits to 0. More information in Section 2.1 (p. External Interrupt 1 Port Select escription
14:12 11 10:8	EXTIPSEL3 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL1 Select input po Value 0 1 2 3 4 5	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co Ox0 Ort for external interrupt 2. Mode PORTA PORTB PORTC PORTB PORTC PORTB PORTC PORTB PORTC PORTC PORTD PORTC PORTD PORTC PORTD PORTE PORTT Mode PORTA PORTA PORTB PORTC PORTD PORTE PORTT To ensure co Ox0 Ox1 Ox1 Ox1 Mode PORTA	RW Do Proportion Prop	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort D pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 future devices, always write bits to 0. More information in Section 2.1 (p. External Interrupt 2 Port Select escription ort A pin 2 selected for external interrupt 2 ort B pin 2 selected for external interrupt 2 ort C pin 2 selected for external interrupt 2 ort E pin 2 selected for external interrupt 2 ort E pin 2 selected for external interrupt 2 ort F pin 2 selected for external interrupt 2 future devices, always write bits to 0. More information in Section 2.1 (p. External Interrupt 1 Port Select escription ort A pin 1 selected for external interrupt 1
14:12 11 10:8	EXTIPSEL3 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL2 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL1 Select input po Value 0 1 2 3 4 5	Ox0 Ort for external interrupt 3. Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co Ox0 Ort for external interrupt 2. Mode PORTA PORTB PORTC PORTB PORTC PORTD PORTB PORTC PORTD PORTC PORTD PORTC PORTD PORTC PORTD PORTE PORTF To ensure co Ox0 Ox1 Ox1 Ox2 Ox3 Ox4 Ox4 Ox4 Ox4 Ox4 Ox4 Ox4 Ox4 Ox4 Ox4	RW Properties a second	escription ort A pin 3 selected for external interrupt 3 ort B pin 3 selected for external interrupt 3 ort C pin 3 selected for external interrupt 3 ort D pin 3 selected for external interrupt 3 ort E pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 ort F pin 3 selected for external interrupt 3 future devices, always write bits to 0. More information in Section 2.1 (p. External Interrupt 2 Port Select escription ort A pin 2 selected for external interrupt 2 ort B pin 2 selected for external interrupt 2 ort C pin 2 selected for external interrupt 2 ort E pin 2 selected for external interrupt 2 ort F pin 2 selected for external interrupt 2 future devices, always write bits to 0. More information in Section 2.1 (p. External Interrupt 1 Port Select escription ort A pin 1 selected for external interrupt 1 ort B pin 1 selected for external interrupt 1



Bit	Name	Reset	Acces	ss Description
	Value	Mode		Description
	5	PORTF		Port F pin 1 selected for external interrupt 1
3	Reserved	To ensure co	ompatibility w	ith future devices, always write bits to 0. More information in Section 2.1 (p. 3)
2:0	EXTIPSEL0	0x0	RW	External Interrupt 0 Port Select
	Select input por	rt for external interrupt 0.		
	Value	Mode		Description
	0	PORTA		Port A pin 0 selected for external interrupt 0
	1	PORTB		Port B pin 0 selected for external interrupt 0
	2	PORTC		Port C pin 0 selected for external interrupt 0
	3	PORTD		Port D pin 0 selected for external interrupt 0
	4	PORTE		Port E pin 0 selected for external interrupt 0
	5	PORTF		Port F pin 0 selected for external interrupt 0

28.5.11 GPIO_EXTIPSELH - External Interrupt Port Select High Register

Offset															Bi	t Po	siti	on														
0x104	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	œ	7	9	2	4	က	7	-	0
Reset			0x0				0x0				0x0				0x0				0x0				0x0				0x0				0x0	
Access			-W				RW				RW				-W				M				-W				RW				-W	
Name			EXTIPSEL15				EXTIPSEL14				EXTIPSEL13				EXTIPSEL12				EXTIPSEL11				EXTIPSEL10				EXTIPSEL9				EXTIPSEL8	

	Ш	Ш		Ш		Ш		Ш		Ш				
D'4	N		D											
Bit	Name		Reset		Acce	ess	Descr	iption						
31	Reserved		To ensu	ire comp	atibility (with futur	e device	es, always	write	bits to 0. Mo	ore info	ormation in	Secti	ion 2.1 (p.
30:28	EXTIPSEL15		0x0		RW		Externa	al Interrup	t 15 F	ort Select				
	Select input port	t for external	interrup	t 15.										
	Value	Mode				Descrip	otion							
	0	PORTA				Port A	pin 15 se	lected for ex	ternal	interrupt 15				
	1	PORTB				Port B	pin 15 se	lected for ex	ternal	interrupt 15				
	2	PORTC				Port C	pin 15 se	lected for ex	cternal	interrupt 15				
	3	PORTD				Port D	pin 15 se	lected for ex	cternal	interrupt 15				
	4	PORTE				Port E	pin 15 se	lected for ex	ternal	interrupt 15				
	5	PORTF				Port F	oin 15 se	lected for ex	ternal	interrupt 15				
27	Reserved		To ensu	ıre comp	atibility (with futur	e device	es, always	write	bits to 0. Mo	ore info	ormation in	n Secti	ion 2.1 (p.
26:24	EXTIPSEL14		0x0		RW		Externa	al Interrup	t 14 F	Port Select				
	Select input port	for external	interrup	t 14.										
	Value	Mode				Descrip	otion							
	0	PORTA				Port A	pin 14 se	lected for ex	ternal	interrupt 14				
	1	PORTB				Port B	pin 14 se	lected for ex	ternal	interrupt 14				
	2	PORTC				Port C	pin 14 se	lected for ex	kternal	interrupt 14				
	3	PORTD				Port D	pin 14 se	lected for ex	cternal	interrupt 14				
	4	PORTE				Port E	pin 14 se	lected for ex	ternal	interrupt 14				
	5	PORTF				Port F	oin 14 se	lected for ex	ternal	interrupt 14				
23	Reserved		To ensu	ire comp	atibility (with futur	e device	es, always	write	bits to 0. Mo	ore info	ormation in	n Secti	ion 2.1 (p.

Select input port for external interrupt 13.

0x0

EXTIPSEL13

22:20

External Interrupt 13 Port Select

RW



Bit	Name	Reset	Acces	s Description
	Value	Mode		Description
	0	PORTA		Port A pin 13 selected for external interrupt 13
	1	PORTB		Port B pin 13 selected for external interrupt 13
	2	PORTC		Port C pin 13 selected for external interrupt 13
	3	PORTD		Port D pin 13 selected for external interrupt 13
	4	PORTE		Port E pin 13 selected for external interrupt 13
	5	PORTF		Port F pin 13 selected for external interrupt 13
19	Reserved	To ensure co	mpatibility wit	th future devices, always write bits to 0. More information in Section 2.1 (p. 3
18:16	EXTIPSEL12	0x0	RW	External Interrupt 12 Port Select
		ort for external interrupt 12.	1	
	Value	Mode		Description
	0	PORTA		Port A pin 12 selected for external interrupt 12
	1	PORTB		Port B pin 12 selected for external interrupt 12
	2	PORTC		Port C pin 12 selected for external interrupt 12
	3	PORTD		Port D pin 12 selected for external interrupt 12
	4	PORTE		Port E pin 12 selected for external interrupt 12
	5	PORTF		Port F pin 12 selected for external interrupt 12
15	Reserved	To ensure co	mpatibility wit	th future devices, always write bits to 0. More information in Section 2.1 (p.
4:12	EXTIPSEL11	0x0	RW	External Interrupt 11 Port Select
		ort for external interrupt 11.		
	Value	Mode		Description
	0	PORTA		Port A pin 11 selected for external interrupt 11
	1	PORTB		Port B pin 11 selected for external interrupt 11
	2	PORTC		Port C pin 11 selected for external interrupt 11
	3	PORTD		Port D pin 11 selected for external interrupt 11
	4	PORTE		Port E pin 11 selected for external interrupt 11
	5	PORTE		Port F pin 11 selected for external interrupt 11
11	Reserved		omnatihility wit	th future devices, always write bits to 0. More information in Section 2.1 (p.
1 1		10 6113016 60	irripatibility wit	infuture devices, always write bits to 0. More information in Section 2.1 (p.
10.0		00	DW	Fortennal Internant 40 Part Calast
10:8	EXTIPSEL10	0x0	RW	External Interrupt 10 Port Select
0:8	EXTIPSEL10 Select input po	ort for external interrupt 10.	RW	
0:8	EXTIPSEL10 Select input po	ort for external interrupt 10.	RW	Description
0:8	EXTIPSEL10 Select input po	Mode PORTA		Description Port A pin 10 selected for external interrupt 10
0:8	EXTIPSEL10 Select input po Value 0 1	Mode PORTA PORTB		Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10
0:8	EXTIPSEL10 Select input po Value 0 1	Mode PORTA PORTB PORTC		Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10 Port C pin 10 selected for external interrupt 10
0:8	EXTIPSEL10 Select input po Value 0 1	Mode PORTA PORTB		Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10 Port C pin 10 selected for external interrupt 10 Port D pin 10 selected for external interrupt 10
0:8	EXTIPSEL10 Select input po Value 0 1	Mode PORTA PORTB PORTC		Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10 Port C pin 10 selected for external interrupt 10
0:8	EXTIPSEL10 Select input po Value 0 1 2 3	Mode PORTA PORTB PORTC PORTD		Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10 Port C pin 10 selected for external interrupt 10 Port D pin 10 selected for external interrupt 10
7	EXTIPSEL10 Select input po Value 0 1 2 3 4 5	Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co	mpatibility wit	Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10 Port C pin 10 selected for external interrupt 10 Port D pin 10 selected for external interrupt 10 Port E pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 th future devices, always write bits to 0. More information in Section 2.1 (p.
7	EXTIPSEL10 Select input po Value 0 1 2 3 4 5	Mode PORTA PORTB PORTC PORTD PORTE PORTF		Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10 Port C pin 10 selected for external interrupt 10 Port D pin 10 selected for external interrupt 10 Port E pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10
7	EXTIPSEL10 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL9 Select input po	Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co	mpatibility wit	Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10 Port C pin 10 selected for external interrupt 10 Port D pin 10 selected for external interrupt 10 Port E pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 th future devices, always write bits to 0. More information in Section 2.1 (p. External Interrupt 9 Port Select
7	EXTIPSEL10 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL9	Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co	mpatibility wit	Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10 Port C pin 10 selected for external interrupt 10 Port D pin 10 selected for external interrupt 10 Port E pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 th future devices, always write bits to 0. More information in Section 2.1 (p.
7	EXTIPSEL10 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL9 Select input po	Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co	mpatibility wit	Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10 Port C pin 10 selected for external interrupt 10 Port D pin 10 selected for external interrupt 10 Port E pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 th future devices, always write bits to 0. More information in Section 2.1 (p. External Interrupt 9 Port Select
7	EXTIPSEL10 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL9 Select input po Value	Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co 0x0 art for external interrupt 9.	mpatibility wit	Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10 Port C pin 10 selected for external interrupt 10 Port D pin 10 selected for external interrupt 10 Port E pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 th future devices, always write bits to 0. More information in Section 2.1 (p. External Interrupt 9 Port Select
7	EXTIPSEL10 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL9 Select input po Value	Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co	mpatibility wit	Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10 Port C pin 10 selected for external interrupt 10 Port D pin 10 selected for external interrupt 10 Port E pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 th future devices, always write bits to 0. More information in Section 2.1 (p. External Interrupt 9 Port Select Description Port A pin 9 selected for external interrupt 9
7	EXTIPSEL10 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL9 Select input po Value 0 1	Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co Ox0 out for external interrupt 9. Mode PORTA PORTB	empatibility wit	Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10 Port C pin 10 selected for external interrupt 10 Port D pin 10 selected for external interrupt 10 Port E pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 th future devices, always write bits to 0. More information in Section 2.1 (p. External Interrupt 9 Port Select Description Port A pin 9 selected for external interrupt 9 Port B pin 9 selected for external interrupt 9
7	EXTIPSEL10 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL9 Select input po Value 0 1 2	Mode PORTA PORTB PORTC PORTD PORTE PORTF To ensure co 0x0 out for external interrupt 9. Mode PORTA PORTB PORTB PORTF	mpatibility wit	Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10 Port C pin 10 selected for external interrupt 10 Port D pin 10 selected for external interrupt 10 Port E pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 th future devices, always write bits to 0. More information in Section 2.1 (p. External Interrupt 9 Port Select Description Port A pin 9 selected for external interrupt 9 Port B pin 9 selected for external interrupt 9 Port C pin 9 selected for external interrupt 9
7	EXTIPSEL10 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL9 Select input po Value 0 1 2 3	Mode PORTA PORTB PORTC PORTB PORTC PORTF To ensure co Ox0 out for external interrupt 9. Mode PORTA PORTB PORTC Ox0 port for external interrupt 9.	mpatibility wit	Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10 Port C pin 10 selected for external interrupt 10 Port D pin 10 selected for external interrupt 10 Port E pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 External Interrupt 9 Port Select Description Port A pin 9 selected for external interrupt 9 Port B pin 9 selected for external interrupt 9 Port C pin 9 selected for external interrupt 9 Port D pin 9 selected for external interrupt 9
5:4	EXTIPSEL10 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL9 Select input po Value 0 1 2 3 4 5 5	Mode PORTA PORTB PORTC PORTB PORTF To ensure co Ox0 out for external interrupt 9. Mode PORTA PORTB PORTF PORTF To ensure co Ox0 out for external interrupt 9.	nmpatibility wit	Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10 Port C pin 10 selected for external interrupt 10 Port D pin 10 selected for external interrupt 10 Port E pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 External Interrupt 9 Port Select Description Port A pin 9 selected for external interrupt 9 Port B pin 9 selected for external interrupt 9 Port C pin 9 selected for external interrupt 9 Port D pin 9 selected for external interrupt 9 Port E pin 9 selected for external interrupt 9 Port E pin 9 selected for external interrupt 9 Port F pin 9 selected for external interrupt 9 Port F pin 9 selected for external interrupt 9
7 6:4	EXTIPSEL10 Select input po Value 0 1 2 3 4 5 Reserved EXTIPSEL9 Select input po Value 0 1 2 3 4 4 4	Mode PORTA PORTB PORTC PORTB PORTF To ensure co Ox0 out for external interrupt 9. Mode PORTA PORTB PORTF PORTF To ensure co Ox0 out for external interrupt 9.	nmpatibility wit	Description Port A pin 10 selected for external interrupt 10 Port B pin 10 selected for external interrupt 10 Port C pin 10 selected for external interrupt 10 Port D pin 10 selected for external interrupt 10 Port E pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 Port F pin 10 selected for external interrupt 10 External Interrupt 9 Port Select Description Port A pin 9 selected for external interrupt 9 Port B pin 9 selected for external interrupt 9 Port C pin 9 selected for external interrupt 9 Port D pin 9 selected for external interrupt 9 Port E pin 9 selected for external interrupt 9



Bit	Name	Reset	Acces	s Description
	Select input port for	or external interrupt 8.		
	Value	Mode		Description
	0	PORTA		Port A pin 8 selected for external interrupt 8
	1	PORTB		Port B pin 8 selected for external interrupt 8
	2	PORTC		Port C pin 8 selected for external interrupt 8
	3	PORTD		Port D pin 8 selected for external interrupt 8
	4	PORTE		Port E pin 8 selected for external interrupt 8
	5	PORTF		Port F pin 8 selected for external interrupt 8

28.5.12 GPIO_EXTIRISE - External Interrupt Rising Edge Trigger Register

Offset															Bi	t Pc	siti	on														
0x108	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	8	7	9	2	4	3	2	1	0
Reset																								0000	000000							
Access																								7	<u>}</u>							
Name																									EX INISE							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	EXTIRISE	0x0000	RW	External Interrupt n Rising Edge Trigger Enable
	Set bit n to enable t	riggering of external in	terrupt n on rising	edge.
	Value		Des	cription
	EXTIRISE[n] = 0		Risi	ng edge trigger disabled
	EXTIRISE[n] = 1		Risi	ng edge trigger enabled

28.5.13 GPIO_EXTIFALL - External Interrupt Falling Edge Trigger Register

Offset															Bi	t Po	siti	on														
0x10C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	8	7	9	2	4	3	2	-	0
Reset																								000	000000							
Access																								2	≥ Y							
Name																								- L	EXIIFALL							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co.	mpatibility with f	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	EXTIFALL	0x0000	RW	External Interrupt n Falling Edge Trigger Enable



Bit	Name	Reset	Access	Description
	Set bit n to enable triggering	g of external interrup	t n on falling e	dge.
	Value		Descri	ption
	EXTIFALL[n] = 0		Falling	edge trigger disabled
	EXTIFALL[n] = 1		Falling	edge trigger enabled

28.5.14 GPIO_IEN - Interrupt Enable Register

Offset															Bi	t Pc	siti	on														
0x110	33	99	29	28	27	26	52	24	23	22	21	70	19	92	17	16	15	4	13	12	=	10	6	ω	7	9	2	4	က	2	-	0
Reset																								0000	00000							
Access																								2	2							
Name																								}	- - -							

Bit	Name	Reset	Acces	s Description
31:16	Reserved	To ensure co	mpatibility wi	th future devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	EXT	0x0000	RW	External Interrupt n Enable
	Set bit n to enable externa	al interrupt from p	oin n.	
	Value		1	Description
	EXT[n] = 0		F	Pin n external interrupt disabled
	EXT[n] = 1		F	Pin n external interrupt enabled

28.5.15 GPIO_IF - Interrupt Flag Register

Offset															Bit	t Po	siti	on														
0x114	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	7	10	စ	8	7	9	2	4	က	2	-	0
Reset																									000000							
Access																								C	Y							
Name																								ŀ	EX							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	EXT	0x0000	R	External Interrupt Flag n
	Pin n external interrupt flag	j .		
	Value		Desci	ription
	EXT[n] = 0		Pin n	external interrupt flag cleared



Bit	Name	Reset	Acce	ess	Description
	Value			Descript	ion
	EXT[n] = 1			Pin n ext	ternal interrupt flag set

28.5.16 GPIO_IFS - Interrupt Flag Set Register

Offset															Bi	t Po	siti	on														
0x118	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	8	7	9	2	4	ю	2	-	0
Reset																								0000	000000							
Access																								787	>							
Name																								} !	- K J							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	EXT	0x0000	W1	External Interrupt Flag n Set
	Write bit n to 1 to set inte	rrupt flag n.		
	Value		Descrip	otion
	EXT[n] = 0		Pin n e	xternal interrupt flag unchanged
	EXT[n] = 1		Pin n e	xternal interrupt flag set

28.5.17 GPIO_IFC - Interrupt Flag Clear Register

Offset															Bi	t Pc	siti	on														
0x11C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	71	7	10	တ	8	7	9	2	4	က	2	-	0
Reset																								000	000000							
Access																								747	<u>-</u>							
Name																								ļ.	EXI							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	EXT	0x0000	W1	External Interrupt Flag Clear
	Write bit n to 1 to c	lear external interrupt f	lag n.	
	Value		Descri	ption
	EXT[n] = 0		Pin n e	external interrupt flag unchanged
	EXT[n] = 1		Pin n e	external interrupt flag cleared



28.5.18 GPIO_ROUTE - I/O Routing Register

Offset															Bi	t Pc	siti	on														
0x120	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	=	10	6	ω	7	9	2	4	က	2	-	0
Reset																					-		2) X						0	-	-
Access																							2	<u>}</u>						RW	W.	RW
Name																							T C T WAY	3						SWOPEN	SWDIOPEN	SWCLKPEN

				80
Bit	Name	Reset	Access	s Description
31:10	Reserved	To ensure	compatibility witl	h future devices, always write bits to 0. More information in Section 2.1 (p. 3
9:8	SWLOCATION	0x0	RW	I/O Location
	Decides the loca	ation of the SW pins.		
	Value	Mode	1	Description
	0	LOC0	I	Location 0
	1	LOC1	I	Location 1
7:3	Reserved	To ensure	compatibility witl	h future devices, always write bits to 0. More information in Section 2.1 (p. 3
2	SWOPEN	0	RW	Serial Wire Viewer Output Pin Enable
	Enable Serial W	rire Viewer Output conne	ection to pin.	
1	SWDIOPEN	1	RW	Serial Wire Data Pin Enable
	A reset will set	the pin back to a default ou program code before	state as enable	hen this pin is disabled, the device can no longer be accessed by a debugge ed. If you disable this pin, make sure you have at least a 3 second timeoupin. This way, the debugger will have time to halt the device after a rese
0	SWCLKPEN	1	RW	Serial Wire Clock Pin Enable
	debugger. A res timeout at the s	et will set the pin back to	a default state	6: When this pin is disabled, the device can no longer be accessed by as enabled. If you disable this pin, make sure you have at least a 3 secon able the pin. This way, the debugger will have time to halt the device after

28.5.19 GPIO_INSENSE - Input Sense Register

a reset before the pin is disabled.

Offset		Bit Position																														
0x124	31	98	29	28	27	56	25	24	23	22	21	20	19	18	17	16	15	4	13	12	11	10	6	80	7	9	2	4	က	2	_	0
Reset				,							•		•								•	•	,								-	-
Access																															RW	X W
Name																															PRS	L _N

Bit	Name	Reset	Access	Description						
31:2	Reserved	To ensure co	To ensure compatibility with future devices, always write bits to 0. More information in Se							
1	PRS	1	RW	PRS Sense Enable						
	Set this bit to enab	is bit to enable input sensing for PRS.								
0	INT	1	RW	Interrupt Sense Enable						
	Set this bit to enable input sensing for interrupts.									



28.5.20 GPIO_LOCK - Configuration Lock Register

Offset		Bit							t Po	osition																						
0x128	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	11	10	6	8	7	9	2	4	က	2	-	0
Reset																								0000	000000							
Access																								9	<u>}</u>							
Name																								\ \ \ \ \	LOCANE							

Bit	Name	Reset	Access	Description
31:16	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
15:0	LOCKKEY	0x0000	RW	Configuration Lock Key

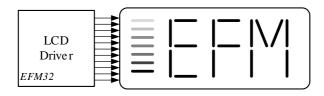
Write any other value than the unlock code to lock MODEL, MODEH, CTRL, PINLOCKN, EPISELL, EIPSELH, INSENSE and SWDPROUTE from editing. Write the unlock code to unlock. When reading the register, bit 0 is set when the lock is enabled.

Mode	Value	Description
Read Operation		
UNLOCKED	0	GPIO registers are unlocked
LOCKED	1	GPIO registers are locked
Write Operation		
LOCK	0	Lock GPIO registers
UNLOCK	0xA534	Unlock GPIO registers



29 LCD - Liquid Crystal Display Driver





Quick Facts

What?

The LCD driver can drive up to 4x40 segmented LCD directly. The LCD driver consumes less than 900 nA in EM2. The animation feature makes it possible to have active animations without CPU intervention.

Why?

Segmented LCD displays are 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 EFM32G the optimal choice for battery-driven systems with LCD panels.

29.1 Introduction

The LCD driver is capable of driving a segmented LCD display with up to 4x40 segments. A voltage boost 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 that can wake-up the device on a regular basis for updating data.

29.2 Features

- Up to 4x40 segments.
- Configurable multiplexing (1, 2, 3, 4)
- 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
- · Voltage boost capabilities
- · Possible to run on external power
- · Programmable contrast
- · Frame Counter
- LCD frame interrupt
- · Direct segment control

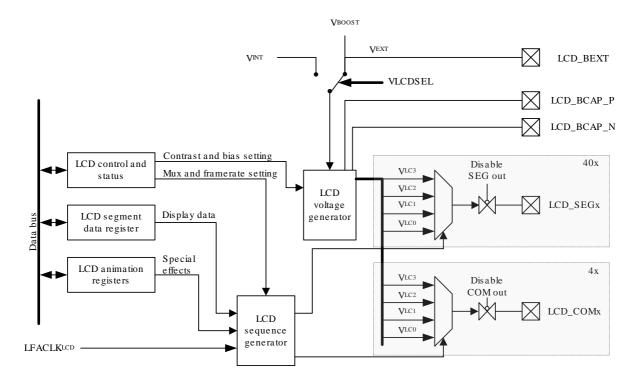


29.3 Functional Description

An overview of the LCD module is shown in Figure 29.1 (p. 423). 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 waveforms are multiplexed between four different common lines and 40 segment lines to support up to 160 different LCD segments. 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 29.1. LCD Block Diagram



For simplicity, only one segment pin and one common terminal is shown in the figure.

29.3.1 LCD Driver Enable

Setting the EN bit in LCD_CTRL 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.

The LCD_SEGEN register determines which segment lines are enabled. Segment lines can be enabled or disabled in groups of 4.

Each LCD segment pin can also be individually disabled by setting the pin to any other state than DISABLED in the GPIO pin configuration. Note that this feature is not available on EFM32G revisions A and B.



29.3.2 Multiplexing, Bias, and Wave Settings

The LCD driver supports different multiplexing and bias settings, and these can be set individually in the MUX and BIAS bits in LCD_DISPCTRL respectively, see Table 29.1 (p. 424) and Table 29.2 (p. 424).

Note

If the MUX and BIAS settings in LCD_DISPCTRL are changed while the LCD driver is enabled, the output waveform is unpredictable and may lead to a DC-component for one LCD frame.

The MUX setting determines the number of LCD COM lines that are enabled. 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 finally when quadruplex multiplexing is selected, all COM lines are driven by the LCD driver.

See Section 29.3.3 (p. 425) for waveforms for the different bias and multiplexing settings.

The waveforms generated by the LCD controller can be generated in two different versions, regular and low-power. The low power mode waveforms have a lower switching frequency than the regular waveforms, and thus consume less power. The WAVE bit in LCD_DISPCTRL decides which waveforms to generate. An example of a low-power waveform is shown in Figure 29.2 (p. 425), and an example of a regular waveform is shown in Figure 29.3 (p. 425).

Table 29.1. LCD Mux Settings

MUX	Mode	Multiplexing	
0	00	Static	Static (segments can be multiplexed with LCD_COM[0])
0	01	Duplex	Duplex (segments can be multiplexed with LCD_COM[1:0])
0	10	Triplex	Triplex (segments can be multiplexed with LCD_COM[2:0])
0	11	Quadruplex	Quadruplex (segments can be multiplexed with LCD_COM[3:0])

Table 29.2. 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 29.3. LCD Wave Settings

WAVE	Mode	Wave mode
0	LowPower	Low power optimized waveform output
1	Normal	Regular waveform output



Figure 29.2. LCD Low-power Waveform for LCD_COM0 in Quadruples Multiplex Mode, 1/3 Bias

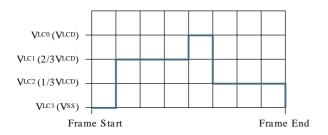
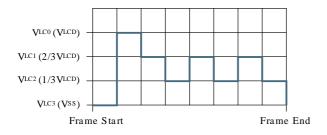


Figure 29.3. LCD Normal Waveform for LCD_COM0 in Quadruples Multiplex Mode, 1/3 Bias



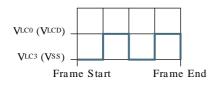
29.3.3 Waveform Examples

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.

29.3.3.1 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} OUT
- V_{RMS} (off) = 0 (V_{SS})

Figure 29.4. LCD Static Bias and Multiplexing - LCD_COM0



29.3.3.2 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. Figures show 1/2 bias and duplex multiplexing (waveforms show two frames)



Figure 29.5. LCD 1/2 Bias and Duplex Multiplexing - LCD_COM0

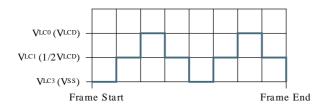
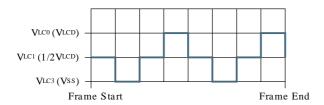


Figure 29.6. LCD 1/2 Bias and Duplex Multiplexing - LCD_COM1



1/2 bias and duplex multiplexing - LCD_SEG0

The LCD_SEG0 waveform on the left is just 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 the figures below, this waveform will turn ON pixels connected to LCD_COM0, while pixels connected to LCD_COM1 will be turned OFF.

Figure 29.7. LCD 1/2 Bias and Duplex Multiplexing - LCD_SEG0

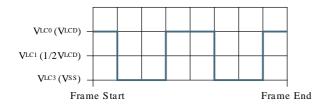
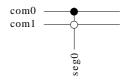


Figure 29.8. LCD 1/2 Bias and Duplex Multiplexing - LCD_SEG0 Connection

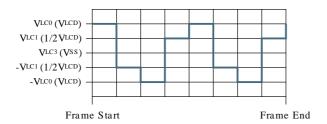


1/2 bias and duplex multiplexing - LCD_SEG0-LCD_COM0

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.79 \times V_{LCD OUT}$
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM0 will be ON with this waveform.



Figure 29.9. 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)
- $V_{RMS} = 0.35 \times V_{LCD_OUT}$
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM0 will be OFF with this waveform

Figure 29.10. LCD 1/2 Bias and Duplex Multiplexing - LCD_SEG0-LCD_COM1



29.3.3.3 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. Figures show 1/3 bias and duplex multiplexing (waveforms show two frames).

Figure 29.11. LCD 1/3 Bias and Duplex Multiplexing - LCD_COM0

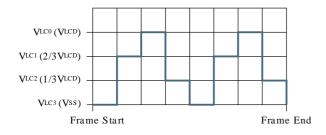
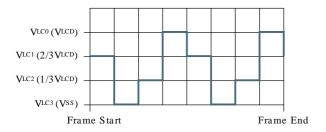




Figure 29.12. LCD 1/3 Bias and Duplex Multiplexing - LCD_COM1



1/3 bias and duplex multiplexing - LCD_SEG0

The LCD_SEG0 waveform on the left 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 the figures below, this waveform will turn ON pixels connected to LCD_COM0, while pixels connected to LCD_COM1 will be turned OFF.

Figure 29.13. LCD 1/3 Bias and Duplex Multiplexing - LCD_SEG0

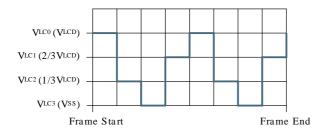
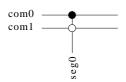


Figure 29.14. LCD 1/3 Bias and Duplex Multiplexing - LCD_SEG0 Connection

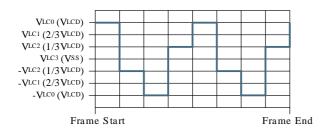


1/3 bias and duplex multiplexing - LCD_SEG0-LCD_COM0

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.75 \times V_{LCD_OUT}$
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM0 will be ON with this waveform



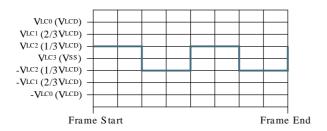
Figure 29.15. LCD 1/3 Bias and Duplex Multiplexing - LCD_SEG0-LCD_COM0



1/3 bias and duplex multiplexing - LCD_SEG0-LCD_COM0

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.33 \times V_{LCD OUT}$
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM1 will be OFF with this waveform

Figure 29.16. LCD 1/3 Bias and Duplex Multiplexing - LCD_SEG0-LCD_COM1



29.3.3.4 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. Figures show 1/2 bias and triplex multiplexing (waveforms show two frames).

Figure 29.17. LCD 1/2 Bias and Triplex Multiplexing - LCD_COM0

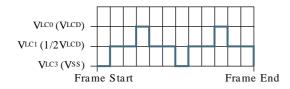


Figure 29.18. LCD 1/2 Bias and Triplex Multiplexing - LCD_COM1

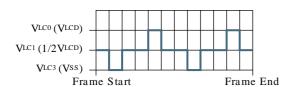
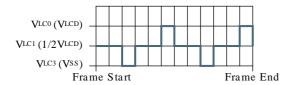




Figure 29.19. LCD 1/2 Bias and Triplex Multiplexing - LCD_COM2



1/2 bias and triplex multiplexing - LCD_SEG0

The LCD_SEG0 waveform on the left 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 the figures below, this waveform will turn ON pixels connected to LCD_COM1, while pixels connected to LCD_COM0 and LCD_COM2 will be turned OFF.

Figure 29.20. LCD 1/2 Bias and Triplex Multiplexing - LCD_SEG0

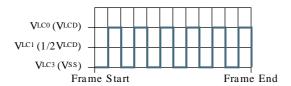
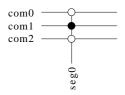


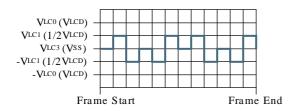
Figure 29.21. LCD 1/2 Bias and Triplex Multiplexing - LCD_SEG0 Connection



1/2 bias and triplex multiplexing - LCD_SEG0-LCD_COM0

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.4 \times V_{LCD_OUT}$
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM0 will be OFF with this waveform

Figure 29.22. LCD 1/2 Bias and Triplex Multiplexing - LCD_SEG0-LCD_COM0

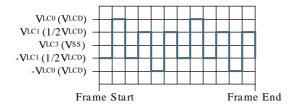




1/2 bias and triplex multiplexing - LCD_SEG0-LCD_COM1

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.7 V_{LCD_OUT}$
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM1 will be ON with this waveform

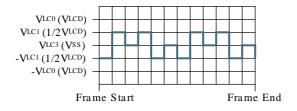
Figure 29.23. LCD 1/2 Bias and Triplex Multiplexing - LCD_SEG0-LCD_COM1



1/2 bias and triplex multiplexing - LCD_SEG0-LCD_COM2

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.4 \times V_{LCD OUT}$
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM2 will be OFF with this waveform

Figure 29.24. LCD 1/2 Bias and Triplex Multiplexing - LCD_SEG0-LCD_COM2



29.3.3.5 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. Figures show 1/3 bias and triplex multiplexing (waveforms show two frames).

Figure 29.25. LCD 1/3 Bias and Triplex Multiplexing - LCD_COM0

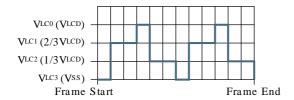




Figure 29.26. LCD 1/3 Bias and Triplex Multiplexing - LCD_COM1

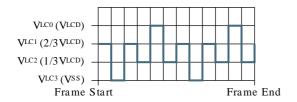
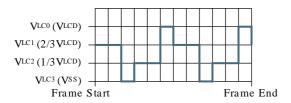


Figure 29.27. LCD 1/3 Bias and Triplex Multiplexing - LCD_COM2



1/3 bias and triplex multiplexing - LCD_SEG0

The LCD_SEG0 waveform illustrates how different segment waveforms can be multiplexed with the COM lines in order to turn on and off LCD pixels. As illustrated in the figures below, this waveform will turn ON pixels connected to LCD_COM1, while pixels connected to LCD_COM0 and LCD_COM2 will be turned OFF.

Figure 29.28. LCD 1/3 Bias and Triplex Multiplexing - LCD_SEG0

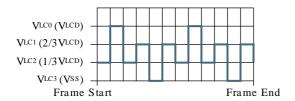
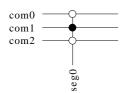


Figure 29.29. LCD 1/3 Bias and Triplex Multiplexing - LCD_SEG0 Connection

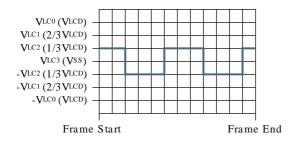


1/3 bias and triplex multiplexing - LCD_SEG0-LCD_COM0

- DC voltage = 0 (over one frame)
- V_{RMS} = 0.33 V_{LCD} OUT
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM0 will be OFF with this waveform



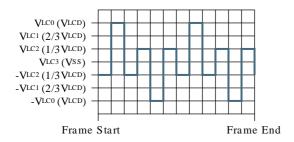
Figure 29.30. LCD 1/3 Bias and Triplex Multiplexing - LCD_SEG0-LCD_COM0



1/3 bias and triplex multiplexing - LCD_SEG0-LCD_COM1

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.64 \times V_{LCD_OUT}$
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM1 will be ON with this waveform

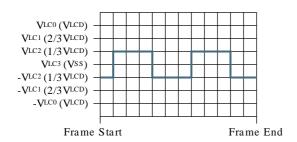
Figure 29.31. LCD 1/3 Bias and Triplex Multiplexing - LCD_SEG0-LCD_COM1



1/3 bias and triplex multiplexing - LCD_SEG0-LCD_COM2

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.33 \times V_{LCD OUT}$
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM2 will be OFF with this waveform

Figure 29.32. LCD 1/3 Bias and Triplex Multiplexing - LCD_SEG0-LCD_COM2



29.3.3.6 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. Figures show 1/3 bias and quadruplex multiplexing (waveforms show two frames).



Figure 29.33. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_COM0

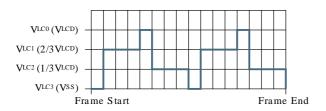


Figure 29.34. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_COM1

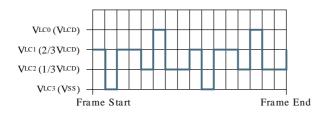


Figure 29.35. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_COM2

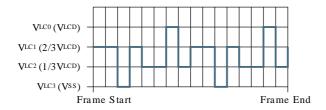
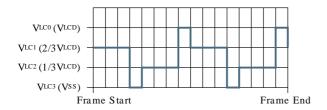


Figure 29.36. LCD 1/3 Bias and Quadruplex Multiplexing - LCD COM3



1/3 bias and quadruplex multiplexing - LCD_SEG0

The LCD_SEG0 waveform on the left 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 the figures below, 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 29.37. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_SEG0

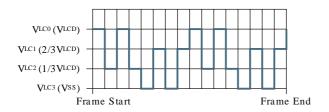
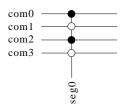


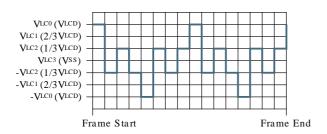
Figure 29.38. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_SEG0 Connection



1/3 bias and quadruplex multiplexing - LCD_SEG0-LCD_COM0

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.58 \times V_{LCD\ OUT}$
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM0 will be ON with this waveform

Figure 29.39. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_SEG0-LCD_COM0

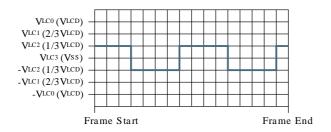


1/3 bias and quadruplex multiplexing - LCD_SEG0-LCD_COM1

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.33 \times V_{LCD_OUT}$
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM1 will be OFF with this waveform



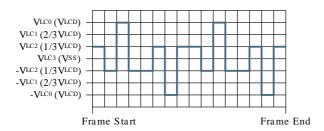
Figure 29.40. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_SEG0-LCD_COM1



1/3 bias and quadruplex multiplexing - LCD_SEG0-LCD_COM2

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.58 \times V_{LCD\ OUT}$
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM2 will be ON with this waveform

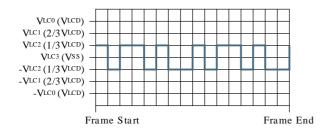
Figure 29.41. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_SEG0-LCD_COM2



1/3 bias and quadruplex multiplexing - LCD SEG0-LCD COM2

- DC voltage = 0 (over one frame)
- $V_{RMS} = 0.33 \times V_{LCD OUT}$
- The LCD display pixel that is connected to LCD_SEG0 and LCD_COM3 will be OFF with this waveform

Figure 29.42. LCD 1/3 Bias and Quadruplex Multiplexing- LCD_SEG0-LCD_COM3



29.3.4 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_OUT} . The contrast is set by CONLEV in LCD_DISPCTRL, and can be adjusted relative to either V_{DD} (V_{LCD}) or Ground using CONCONF in LCD_DISPCTRL. See Table 29.4 (p. 437) and Table 29.5 (p. 437), Table 29.5 (p. 437) and Table 29.6 (p. 438).

Table 29.4. LCD Contrast

BIAS	CONLEV	Equation	Range
00	00000-11111	$V_{LCD_OUT} = V_{LCD} \times (0.61 \times (1 + CONLEV/(2^5 - 1)))$	$CONLEV = 0 \Rightarrow V_{LCD_OUT} = 0.61V_{LCD}$
			CONLEV = 31 => V _{LCD_OUT} = V _{LCD}
01	00000-11111	V _{LCD_OUT} = V _{LCD} x (0.53 x (1 + CONLEV/(2 ⁵ - 1)))	CONLEV = 0 => V _{LCD_OUT} = 0.53V _{LCD}
			CONLEV = 31 => V _{LCD_OUT} = V _{LCD}
10	00000-11111	V _{LCD_OUT} = V _{LCD} x (0.61 x (1 + CONLEV/(2 ⁵ - 1)))	CONLEV = 0 => V _{LCD_OUT} = 0.61V _{LCD}
			CONLEV = 31 => V _{LCD_OUT} = V _{LCD}

Note

Reset value is maximum contrast

Table 29.5. LCD Contrast Function

CONCONF Function	
0	Contrast is adjusted relative to V _{DD} (V _{LCD})
1 Contrast is adjusted relative to Ground	



Table 29.6. LCD Principle of Contrast Adjustment for Different Bias Settings.

	Contrast adjustment relative to V _{DD} (V _{LCD}) (CONCONF = 0)	Contrast adjustment relative to GND (CONCONF = 1)	No contrast adjustment (CONLEV = 11111)
1/3 bias	RX VLCD RX VLCO R0 VLC1 R1 W VLC2 R2 W VLC3	RO W VICO RO W VICO R1 W VICD_OUT R2 W VIC3 Rx W	R0
1/2 bias	RX VICD RX VICO RO VICO VIC	RO VICO VICO VICO VICO VICO VICO VICO VICO	RO VICE VICE VICE OUT R1 VICE VICE VICE OUT
Static	RX VICED RX VICED VIC	RO VICO VICO VICO_OUT Rx	VLCD VLCO VLCO VLCO OUT

R0 = R1 = R2 = R3 in the figure, while Rx is adjusted by changing the CONLEV bits.

29.3.5 V_{LCD} Selection

By default, the LCD driver runs on main external power ($V_{LCD} = V_{DD}$), see Table 29.7 (p. 439). An internal boost circuit can be enabled by setting VBOOSTEN in CMU_LCDCTRL and selecting the boosted voltage by setting VLCDSEL in LCD_DISPCTRL. This will boosts V_{LCD} to V_{BOOST} . V_{BOOST} can be selected in the range of 3.0 V – 3.6 V by configuring VBLEV in LCD_DISPCTRL. Note that the boost circuit is not designed to operate with the selected boost voltage, V_{BOOST} , smaller than V_{DD} . The boost circuit can boost the V_{LCD} up to 3.6 V when V_{DD} is as low as 2.0 V.

When using the voltage booster, the LCD_BEXT pin must be connected through a 1 μ F capacitor to VSS, and the LCD_BCAP_P and LCD_BCAP_N pins must be connected to each other through a 22 nF capacitor.

It is also possible to connect a dedicated power supply to the LCD module. The LCD external power supply must be connected to the LCD_BEXT pin and VLCDSEL in LCD_DISPCTRL must be set. In this mode, the voltage booster should be disabled.



Table 29.7. LCD V_{LCD}

VLCDSEL	Mode	V _{LCD}	
0 VDD		V _{DD} (same as main external power)	
1 VBOOST		Voltage booster/External V _{DD}	

29.3.6 VBOOST Control

The boost voltage is configurable. By programming the VBLEV bits in LCD_DISPCTRL, the boost voltage level can be adjusted between 3.0V and 3.6V.

The boost circuit will use an update frequency given by the VBFDIV bits in CMU_LCDCTRL, see Table 29.8 (p. 439)). It is possible to adjust the frequency to optimize performance for all kinds of LCD panels (large capacitors may require less frequent updates, while small capacitors may require more frequent updates). A lower update frequency would in general lead to smaller current consumption.

Table 29.8. LCD V_{BOOST} Frequency

VBFDIV	V _{BOOST} Update Frequency
000	LFACLK
001	LFACLK/2
010	LFACLK/4
011	LFACLK/8
100	LFACLK/16
101	LFACLK/32
110	LFACLK/64
111	LFACLK/128

29.3.7 Frame rate

It is important to choose the correct frame rate for the LCD display. Normally, the frame rate should be between 30 and 100 Hz. A frame rate below 30 Hz may lead to flickering, while a frame rate above 100 Hz may lead to ghostering and unnecessarily high power consumption.

29.3.7.1 Clock Selection and Prescaler

The LFACLK is prescaled to LFACLK_{LCDpre}in the CMU. The available prescaler settings are:

- LFCLK16: LFACLK_{LCDpre} = LFACLK/16
- LFCLK32: LFACLK_{LCDpre} = LFACLK/32
- LFCLK64: LFACLK_{LCDpre} = LFACLK/64
- LFCLK128: LFACLK_{LCDpre} = LFACLK/128

In addition to selecting the correct prescaling, the clock source can be selected in the CMU.

To use this module, the LE interface clock must be enabled in CMU_HFCORECLKEN0, in addition to the module clock.

29.3.7.2 Frame rate Division Register

The frame rate is set in the CMU by programming the frame rate division bits FDIV in CMU_LCDCTRL. This setting should not be changed while the LCD driver is running. The equation for calculating the resulting frame rate is given from Equation 29.1 (p. 440)

(29.1)



LCD Frame rate Calculation

$$LFACLK_{LCD} = LFACLK_{LCDpre}/(1 + FDIV)$$

Table 29.9. LCD Frame rate Conversion Table

		Resulting Frame rate, CLK _{FRAME} (Hz)							
MUX Mode	Frame- rate formula	LFACLK _{LCDpre} = 2 kHz		LFACLK _{LCDpre} = 1 kHz		LFACLK _{LCDpre} = 0.5 kHz		LFACLK _{LCDpre} = 0.25 kHz	
		Min	Max	Min	Max	Min	Max	Min	Max
Static	LFACLK _{LCD} /2	128	1024	64	512	32	256	16	128
Duplex	LFACLK _{LCD} /4	64	512	32	256	16	128	8	64
Triplex	LFACLK _{LCD} /6	43	341	21	171	11	85	5	43
Quadruplex	LFACLK _{LCD} /8	32	256	16	128	8	64	4	32

Table settings: Min: FDIV = 7, Max: FDIV = 0

29.3.8 Data Update

The LCD Driver logic that controls the output waveforms is clocked on LFACLK_{LCDpre}. The LCD data and Control Registers are clocked on the HFCORECLK. 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

The different methods to update data are controlled by the UDCTRL bits in LCD_CTRL.

Table 29.10. 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 Section 29.3.9 (p. 440) 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.

29.3.9 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 UDCTRL = 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 eight frame

This is controlled by the FCPRESC in LCD_BACTRL, see Table 29.11 (p. 441)

Table 29.11. FCPRESC

FCPRESC	Mode	Description	General equation
00	Div1	CLK _{FRAME} /1	
01	Div2	CLK _{FRAME} /2	$CLK_{FC} = CLK_{FRAME}/2^{FCPRESC}$
10	Div4	CLK _{FRAME} /4	GLRFC = GLRFRAME/2
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.

LCD Event Frequency Equation
$$CLK_{EVENT} = CLK_{FC}/(1 + FCTOP[5:0]) Hz$$
 (29.2)

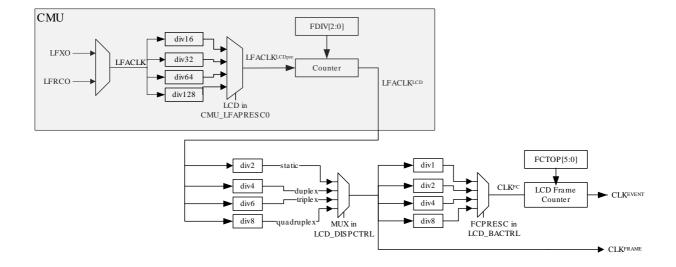
The above equation shows how to set-up the LCD event frequency. In this example, the frame rate is 64Hz, and the LCD event frequency should be set-up to 2 seconds.

Example 29.1. LCD Event Frequency Example

- Write FCPRESC to 3 => CLK_{FC} = 8Hz (0.125 seconds)
- Write FCTOP to 15 => CLK_{EVENT} = 0.5Hz (2 seconds)

If higher resolution is required, configure a lower prescaler value and increase the FCPRESC in LCD_BACTRL accordingly (e.g. FCPRESC = 2, FCTOP = 31).

Figure 29.43. LCD Clock System in LCD Driver





29.3.10 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 FC bit in LCD IEN.

29.3.11 Blink, Blank, and Animation Features

29.3.11.1 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 Section 29.3.9 (p. 440). See Section 29.3.8 (p. 440) for details regarding synchronization of the blink feature. The FC must be on for blink to work.

29.3.11.2 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. See Section 29.3.8 (p. 440) for details regarding synchronization of blank.

29.3.11.3 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 shift operations applied to the shift registers are controlled by AREGASC and AREGBSC in LCD_BACTRL as shown in the table below. Note also that the FC must be on for animation to work, as it is the FC event that drives the animation state machine.

Table 29.12. 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 is controlled by ALOGSEL in LCD_BACTRL as shown in Table 29.13 (p. 442). In addition, the regular segment data SEGD0[7:0] is OR'ed with the animation pattern to generate the resulting output.

Table 29.13. 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 Section 29.3.9 (p. 440). By reading ASTATE in LCD_STATUS, software can identify which state that is currently active in the state sequence. Note that



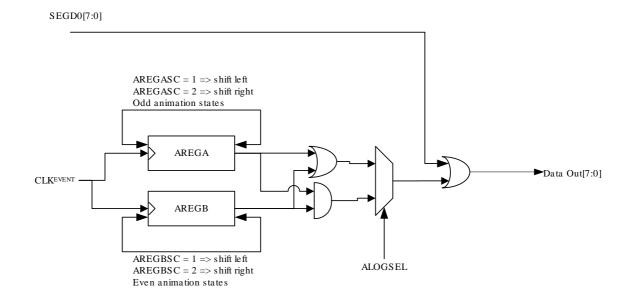
the shifting operation is performed on internal registers that are not accessible in SW (when reading LCD_AREGA and LCD_AREGB, the data that was original written will also be read back). The SW must utilize the knowledge about the current state (ASTATE) to calculate what is currently output. ASTATE is cleared when LCD_AREGA or LCD_AREGB are updated with new values. See Table 29.14 (p. 443) for an example.

Table 29.14. 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
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	0000011	00000011
13	10000001	0000011	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 29.44. LCD Block Diagram of the Animation Circuit





Example 29.2. LCD Animation Enable 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.

Example 29.3. LCD Animation Dependence 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.

29.3.12 LCD in Low Energy Modes

As long as the LFACLK is running (EM0-EM2), 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

29.3.13 Register access

Since this module is a Low Energy Peripheral, and runs off a clock which is asynchronous to the HFCORECLK, special considerations must be taken when accessing registers. Please refer to Section 5.3 (p. 19) for a description on how to perform register accesses to Low Energy Peripherals.



29.4 Register Map

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

Offset	Name	Туре	Description
0x000	LCD_CTRL	RW	Control Register
0x004	LCD_DISPCTRL	RW	Display Control Register
0x008	LCD_SEGEN	RW	Segment Enable Register
0x00C	LCD_BACTRL	RW	Blink and Animation Control Register
0x010	LCD_STATUS	R	Status Register
0x014	LCD_AREGA	RW	Animation Register A
0x018	LCD_AREGB	RW	Animation Register B
0x01C	LCD_IF	R	Interrupt Flag Register
0x020	LCD_IFS	W1	Interrupt Flag Set Register
0x024	LCD_IFC	W1	Interrupt Flag Clear Register
0x028	LCD_IEN	RW	Interrupt Enable Register
0x040	LCD_SEGD0L	RW	Segment Data Low Register 0
0x044	LCD_SEGD1L	RW	Segment Data Low Register 1
0x048	LCD_SEGD2L	RW	Segment Data Low Register 2
0x04C	LCD_SEGD3L	RW	Segment Data Low Register 3
0x050	LCD_SEGD0H	RW	Segment Data High Register 0
0x054	LCD_SEGD1H	RW	Segment Data High Register 1
0x058	LCD_SEGD2H	RW	Segment Data High Register 2
0x05C	LCD_SEGD3H	RW	Segment Data High Register 3
0x060	LCD_FREEZE	RW	Freeze Register
0x064	LCD_SYNCBUSY	R	Synchronization Busy Register

29.5 Register Description

29.5.1 LCD_CTRL - Control Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

Offset								,	,						Bi	t Po	siti	on													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	œ	7	9	2	4	ю	2 +	0
Reset			•	,											,							•	•						,	0x0	0
Access																														RW	RW
Name																														UDCTRL	E S

Bit	Name	Reset	Access	Description
31:3	Reserved	To ensure con	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
2:1	UDCTRL	0x0	RW	Update Data Control

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
1	FCEVENT	The data transfer is done at the next event triggered by the Frame Counter



Bit	Name	Reset	Acce	ss Description
	Value	Mode		Description
	2	FRAMESTART		The data transfer is done continuously at every LCD frame start
0	EN	0	RW	LCD Enable
	When this bit	is set, the LCD driver is enab	led and the	driver will start outputting waveforms on the com/segment lines.

29.5.2 LCD_DISPCTRL - Display Control Register

29. 3.2	Z L'	Uυ	'_L	JI	אכ	C	ΙK	L	- L	<i>7</i> 15	ρι	1 y	C	OI	llr	OI	Κŧ	: g	151	lei												
Offset															Bi	t Po	siti	on														
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	=	10	6	8	7	9	2	4	8	2	-	0
Reset													0x3			0	0					0x1F						0	()	000	OXO	
Access													RW			RW	RW					RW						RW	2	2	Z.	-
Name													VBLEV			VLCDSEL	CONCONF					CONLEV						WAVE	OVI	SKIG	×	<u> </u>
Bit	Na	Name Reset									Access Description																					
31:21	Re	serv	ed					То	ensi	ure c	comp	atibi	ility	with	futu	ire d	evice	es, a	lwa	ys v	vrite	bits	to 0.	Mor	e inf	orm	atio	n in (Sect	ion 2	.1 (p	. 3)
20:18	VB	VBLEV 0x3								R	W		Voltage Boost Level																			

These bits control Voltage Boost level. Please refer to datasheet for further details of the boost levels.

Value	Mode	Description
0	LEVEL0	Minimum boost level
1	LEVEL1	
2	LEVEL2	
3	LEVEL3	
4	LEVEL4	
5	LEVEL5	
6	LEVEL6	
7	LEVEL7	Maximum boost level

17 Reserved To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

16 VLCDSEL 0 RW V_{LCD} Selection

This bit controls which Voltage source that is connected to V_{LCD}.

Value	Mode	Description
0	VDD	VDD
1	VEXTBOOST	Voltage Booster/External VDD

15 CONCONF 0 RW Contrast Configuration

This bit selects whether the contrast adjustment is done relative to V_{LCD} or Ground.

Value	Mode	Description
0	VLCD	Contrast is adjusted relative to V _{LCD}
1	GND	Contrast is adjusted relative to Ground

14:13 Reserved To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)

12:8 CONLEV 0x1F RW Contrast Level

These bits control the contrast setting according to this formula: $V_{LCD_OUT} = V_{LCD} \times 0.5(1 + CONLEV/31)$.

Value	Mode	Description
0	MIN	Minimum contrast
31	MAX	Maximum contrast

7:5 Reserved To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)



Bit	Name	Reset	Access	Description
4	WAVE	0	RW	Waveform Selection
	This bit config	gures the output waveform.		
	Value	Mode	С	Description
	0	LOWPOWER	L	ow power waveform
	1	NORMAL	١	lormal waveform
3:2	BIAS	0x0	RW	Bias Configuration
	These bits se	et the bias mode for the LCD I	Oriver.	
	Value	Mode	С	Description
	0	STATIC	S	Static
	1	ONEHALF	1	/2 Bias
	2	ONETHIRD	1	/3 Bias
1:0	MUX	0x0	RW	Mux Configuration
	These bits se	et the multiplexing mode for th	e LCD Driver.	
	Value	Mode	С	Description
	0	STATIC	S	Static
	1	DUPLEX	[Duplex
	2	TRIPLEX	Т	riplex
	3	QUADRUPLEX	C	Quadruplex

29.5.3 LCD_SEGEN - Segment Enable Register

Offset															Bi	t Po	siti	on														
0x008	31	30	29	28	27	56	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	တ	∞	7	9	2	4	ю	2	-	0
Reset																												000x0				
Access																												₩				
Name																												SEGEN				

Bit	Name	Reset	Access	Description
31:10	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
9:0	SEGEN	0x000	RW	Segment Enable
		output on segme	nt lines 4,5,6 and	resents a group of 4 segment lines. To enable segment lines X to X+3, 7, set bit 1. Each LCD segment pin can also be individually disabled by IO pin configuration.

29.5.4 LCD_BACTRL - Blink and Animation Control Register (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19) .



Offset															Bi	t Po	siti	on														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	7	10	6	8	7	9	2	4	3	2	-	0
Reset															2	3								0	0		0×0	2	2	0	0	0
Access															2	2								S.	RW		S S	<u> </u>	^	RW	RW	RW
Name											FCTOP					- CTN E 3 C								FCEN	ALOGSEL	1	AREGBSC	C0 V C1 d V	NEGAS	AEN	BLANK	BLINKEN

				5	FCP		Ĕ	ALO	ARE	ARE	4	BLIN
										l		
Bit	Name	Re	eset	Acce	ss D	escription						
31:24	Reserved	То	ensure com	patibility w	vith future o	devices, always write bits to 0.	More	e info	rmatio	n in Sect	tion 2	.1 (p. 3
23:18	FCTOP	0x0	00	RW	Fr	ame Counter Top Value						
	These bits con	tain the Top Valu	e for the Fra	me Count	er: CLK _{EVI}	$E_{\text{ENT}} = \text{CLK}_{\text{FC}} / (1 + \text{FCTOP}[5:0]$)]).					
17:16	FCPRESC	0x0)	RW	Fr	ame Counter Prescaler						
	These bits con	trols the prescalir	ng value for t	the Frame	Counter in	nput clock.						
	Value	Mode			Description	n						
	0	DIV1			CLK _{FC} = 0	CLK _{FRAME} / 1						
	1	DIV2			CLK _{FC} = 0	CLK _{FRAME} / 2						
	2	DIV4			CLK _{FC} = 0	CLK _{FRAME} / 4						
	3	DIV8			CLK _{FC} = 0	CLK _{FRAME} / 8						
15:9	Reserved	То	ensure com	patibility w	vith future o	devices, always write bits to 0.	More	e info	rmation	n in Sect	tion 2	.1 (p. 3)
8	FCEN	0		RW	Fr	ame Counter Enable						
	When this bit is	s set, the frame co	ounter is ena	abled.								
7	ALOGSEL	0		RW	Aı	nimate Logic Function Selec	t					
	When this bit is	set, the animatio	n registers a	re AND'ed	I together.	When this bit is cleared, the ar	nimat	tion r	egister	s are OR	ed t	ogether
	Value	Mode			Description	n						
	0	AND			AREGA a	nd AREGB AND'ed						
	1	OR			AREGA a	nd AREGB OR'ed						
6:5	AREGBSC	0x0)	RW	Aı	nimate Register B Shift Cont	trol					
	These bits con	trols the shift ope	ration that is	performe	d on Anim	ation register B.						
	Value	Mode			Description	n						
	0	NOSHIFT			No Shift o	peration on Animation Register B						
	1	SHIFTLEFT			Animation	Register B is shifted left						
	2	SHIFTRIGHT			Animation	Register B is shifted right						
4:3	AREGASC	0x0)	RW	Aı	nimate Register A Shift Cont	trol					
	These bits con	trols the shift ope	ration that is	performe	d on Anim	ation register A.						
	Value	Mode			Description	n						
	0	NOSHIFT			No Shift o	peration on Animation Register A						
	1	SHIFTLEFT			Animation	Register A is shifted left						
	2	SHIFTRIGHT			Animation	Register A is shifted right						
2	AEN	0		RW	Aı	nimation Enable						
	When this bit is	s set, the animate	function is	enabled.								
1	BLANK	0		RW	ВІ	ank Display						
	When this bit is affected when		t output wav	reforms ar	e configur	ed to blank the LCD display.	The	Segn	nent Da	ata Regi	sters	are no
	Value	Descr	ription									
	-											

Display is not "blanked"

Display is "blanked"



Bit	Name	Reset	Access	Description
0	BLINKEN	0	RW	Blink Enable
	When this bit is set, the Blir	k function is enable	d. Every "ON"	segment will alternate between on and off at every Frame Counter Event.

29.5.5 LCD_STATUS - Status Register

Offset															Bi	t Po	siti	on														
0x010	31	30	29	28	27	56	22	24	23	22	21	20	19	92	17	16	15	4	13	12	=	9	6	∞	7	9	2	4	ო	7	-	0
Reset																								0						2	3	
Access																								2						۵	<u> </u>	
Name																								BLINK						ASTATE	20.7	

Bit	Name	Reset	Access	Description
31:9	Reserved	To ensure co	ompatibility with fu	uture devices, always write bits to 0. More information in Section 2.1 (p. 3)
8	BLINK	0	R	Blink State
	This hits indicates	de la la Parla de Cara de Maria	1. St. Ca. A. (41)	or the second of
	to 1 are on.	the blink status. If this	bit is 1, all segm	nents are off. If this bit is 0, the segments(LCD_SEGDxn) which are set
7:4				ature devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:4 3:0	to 1 are on.			

29.5.6 LCD_AREGA - Animation Register A (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

Offset															Bi	t Pc	siti	on														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	=	10	6	8	7	9	2	4	8	2	1	0
Reset																													0x0			
Access																													X N			
Name																													AREGA			

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:0	AREGA	0x00	RW	Animation Register A Data
	This register contains th	e A data for gener	rating animation p	attern.

29.5.7 LCD_AREGB - Animation Register B (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).



Offset									,						Bi	t Po	siti	on														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	8	7	9	2	4	8	2	-	0
Reset																													0000			
Access																												:	S.			
Name																													AREGB			

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure compa	ntibility with futu	re devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:0	AREGB	0x00	RW	Animation Register B Data
	This register contains the B	data for generating	g animation pat	tern.

29.5.8 LCD_IF - Interrupt Flag Register

Offset															Bi	it Po	siti	on														
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	1	10	6	8	7	9	2	4	8	2	-	0
Reset																																0
Access																																~
Name																																FC

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure compa	ntibility with futu	re devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	FC	0	R	Frame Counter Interrupt Flag
	Set when Frame Counter is	zero.		

29.5.9 LCD_IFS - Interrupt Flag Set Register

Offset															Bi	t Po	siti	on														
0x020	31	30	59	28	27	56	25	24	23	22	21	20	19	8	17	16	15	4	13	12	=	10	တ	80	7	9	2	4	က	7	-	0
Reset																																0
Access																																%
Name																																5

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	FC	0	W1	Frame Counter Interrupt Flag Set
	Write to 1 to set FC interrup	ot flag.		



29.5.10 LCD_IFC - Interrupt Flag Clear Register

Offset															Bi	t Po	siti	on														
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	80	7	9	2	4	က	2	-	0
Reset																																0
Access																																W W
Name																																FC

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure compa	tibility with futu	re devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	FC	0	W1	Frame Counter Interrupt Flag Clear
	Write to 1 to clear FC interre	upt flag.		

29.5.11 LCD_IEN - Interrupt Enable Register

Offset								,				,			Bi	t Pc	siti	on				,			,							
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	1	10	6	œ	7	9	2	4	ю	2	-	0
Reset			•	,											,			•			•		•									0
Access																																R W
Name																																FC

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
0	FC	0	RW	Frame Counter Interrupt Enable
	Set to enable inter	rupt on frame counter in	nterrupt flag.	

29.5.12 LCD_SEGD0L - Segment Data Low Register 0 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

Offset															Bi	t Po	siti	on													
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	: [10	6	8	7	9	2	4	3	2	-	0
Reset																0000000															
Access																Š	2														
Name																וטעטפט)))														



Bit	Name	Reset	Access	Description						
31:0	SEGD0L	0x00000000	RW	COM0 Segment Data Low						
	This register contains segn	This register contains segment data for segment lines 0-31 for COM0.								

29.5.13 LCD_SEGD1L - Segment Data Low Register 1 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

Offset															Bi	t Pc	siti	on														
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	1	10	6	8	7	9	2	4	ю	2	-	0
Reset																00000000	000000000															
Access																<u> </u>	À															
Name																200	35.00							_								
Bit	Na	me						Re	set			A	CC	ess		De	scri	iptio	on													
31:0	SE	GD1	L					0x0	0000	0000)	R	W			СО	M1 \$	Segi	me	ent D	ata l	Lov	v									
	Thi	s reg	giste	r cor	ntain	s se	egm	ent c	data 1	for s	egm	ent l	ines	s 0-3	31 fo	r CC	DM1.															

29.5.14 LCD_SEGD2L - Segment Data Low Register 2 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

Offset														Bit	Pos	sitio	on														
0x048	31	59	78	27	26	25	24	23	22	21	20	19	8	17	16	12	4	13	12	=	10	6	∞	7	9	2	4	ო	2	-	0
Reset															00000000x0																
Access															RW																
Name															SEGD2L																
Bit	Name						Re	set			A	CCE	ess		Des	scri	ptic	on													
31:0	SEGD	2L					0x0	0000	0000		R	W			CON	/12 5	Segr	ner	nt D	ata I	_ow										
	This re	giste	r cor	ntain	IS S	egm	ent c	lata f	or se	egm	ent l	lines	s 0-3	31 fo	r CO	M2.															

29.5.15 LCD_SEGD3L - Segment Data Low Register 3 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).



Offset															Ri	t Po	siti	on														
																1					l		1						1			
0x04C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	=	10	6	∞	7	9	2	4	က	2	-	0
Reset																OOOOOOOXO																
Access		≥ 2																														
Name																וציטטאַ																
Bit	Na	Name Reset Access Description																														
31:0	SE	SEGD3L 0x00000000 RW COM3 Segment Data Low																														
	This	This register contains segment data for segment lines 0-31 for COM3.																														

29.5.16 LCD_SEGD0H - Segment Data High Register 0 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19) .

Offset													•		Bi	t Po	siti	on														
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	=	10	6	ω	7	9	2	4	ю	2	-	0
Reset																													00X0			
Access																													& ≷			
Name																													SEGD0H			

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	mpatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:0	SEGD0H	0x00	RW	COM0 Segment Data High
	This register contain	ns segment data for se	gment lines 32-3	9 for COM0.

29.5.17 LCD_SEGD1H - Segment Data High Register 1 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19) .

Offset								·						Bi	t Pc	siti	on														
0x054	31	30	29	28	27	26	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	6	80	7	9	2	4	က	2	-	0
Reset																												0x00			
Access																											;	W.			
Name																												SEGD1H			



Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:0	SEGD1H	0x00	RW	COM1 Segment Data High
	This register contains	s segment data for se	egment lines 32-3	9 for COM1.

29.5.18 LCD_SEGD2H - Segment Data High Register 2 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

Offset															Bi	t Pc	siti	on														
0x058	31	30	29	28	27	26	22	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	တ	∞	7	9	2	4	က	2	1	0
Reset																												0	0000			
Access																												i	≥ Y			
Name																												- - - - - - -	SEGDZH			

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure compa	atibility with futu	re devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:0	SEGD2H	0x00	RW	COM2 Segment Data High
	This register contains segm	ent data for segme	ent lines 32-39	for COM2.

29.5.19 LCD_SEGD3H - Segment Data High Register 3 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 19).

Offset															Bi	t Po	siti	on														
0x05C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	တ	œ	7	9	2	4	ю	2	-	0
Reset																													0x0			
Access																													S N			
Name																													SEGD3H			

Bit	Name	Reset	Access	Description
31:8	Reserved	To ensure compa	atibility with fut	ure devices, always write bits to 0. More information in Section 2.1 (p. 3)
7:0	SEGD3H	0x00	RW	COM3 Segment Data High
	This register contains segment	nent data for segme	ent lines 32-39	for COM3.



29.5.20 LCD_FREEZE - Freeze Register

Offset															Bi	t Po	siti	on														
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	41	13	12	7	10	6	8	7	9	2	4	က	2	-	0
Reset																																0
Access																																RW
Name																																REGFREEZE

Bit	Name	Reset	Access	Description
31:1	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3,
0	REGFREEZE	0	RW	Register Update Freeze
-		-		regions. Spaces 1.0020
		odate of the LCD is postpo		is cleared. Use this bit to update several registers simultaneously.
		odate of the LCD is postpo	oned until this bit i	•
	When set, the up		oned until this bit in Des	is cleared. Use this bit to update several registers simultaneously.

29.5.21 LCD_SYNCBUSY - Synchronization Busy Register

Offset															Bi	t Po	siti	on														
0x064	31	30	29	28	27	56	22	24	23	22	21	20	19	18	17	16	15	4	13	12	7	10	တ	∞	7	9	2	4	က	2	-	0
Reset																					0	0	0	0	0	0	0	0	0	0	0	0
Access																					~	8	~	~	~	2	~	~	22	~	~	22
Name																					SEGD3H	SEGD2H	SEGD1H	SEGDOH	SEGD3L	SEGD2L	SEGD1L	SEGDOL	AREGB	AREGA	BACTRL	CTRL

Bit	Name	Reset	Access	Description
31:12	Reserved	To ensure co	ompatibility with fu	ture devices, always write bits to 0. More information in Section 2.1 (p. 3)
11	SEGD3H	0	R	SEGD3H Register Busy
	Set when the value	written to SEGD3H is	being synchronize	ed.
10	SEGD2H	0	R	SEGD2H Register Busy
	Set when the value	written to SEGD2H is	being synchronize	ed.
9	SEGD1H	0	R	SEGD1H Register Busy
	Set when the value	written to SEGD1H is	being synchronize	ed.
8	SEGD0H	0	R	SEGD0H Register Busy
	Set when the value	written to SEGD0H is	being synchronize	ed.
7	SEGD3L	0	R	SEGD3L Register Busy
	Set when the value	written to SEGD3L is	being synchronize	ed.
6	SEGD2L	0	R	SEGD2L Register Busy
	Set when the value	written to SEGD2L is	being synchronize	ed.
5	SEGD1L	0	R	SEGD1L Register Busy
	Set when the value	written to SEGD1L is	being synchronize	ed.
4	SEGD0L	0	R	SEGD0L Register Busy
	Set when the value	written to SEGD0L is	being synchronize	ed.



Bit	Name	Reset	Access	Description
3	AREGB	0	R	AREGB Register Busy
	Set when the value writt	ten to AREGB is b	eing synchronized	,
2	AREGA	0	R	AREGA Register Busy
	Set when the value writt	ten to AREGA is b	eing synchronized	
1	BACTRL	0	R	BACTRL Register Busy
	Set when the value writt	ten to BACTRL is	being synchronize	d.
0	CTRL	0	R	CTRL Register Busy
	Set when the value writt	ten to CTRL is bei	ing synchronized.	



30 Revision History

30.1 Revision 1.31

February 23rd, 2017

Updated memory system map

Replaced static bit write instruction with reference to the Cortex-M3 manual

Changed gpio pin configuration figure from esd diode to esd black-box to remove incorrect note about LCD overvoltage

Added full wafer as a package option.

Corrected bit alignment in PID0 register in section 3.

Changes in the I²C section

- Updated note.
- Updated Clock Generation section.

Added AUXHFRCO to block diagram.

Added notes in the DMA Controller section.

Updated the register description of LEUARTn_CTRL.

Corrected the DAC f_{sine} equation.

Added and modified notes in the WDOG Clock Source and Register Access sections.

Modified a note in the PCNT Clock Sources section.

Updated the register description of MSC_WDATA.

Updated the register descriptions of USARTn_IF, USARTn_TXDATAX and USARTn_TXDOUBLEX.

Corrected the RMU Reset Input Sources and Connections figure.

Updated the MSC Erase and Write Operations section.

Updated recommendations regarding BIASPROG and ADC_CLK in the ADC chapter.

Updated to revision E.

30.2 Revision 1.30

July 2nd, 2014

Updated current numbers and voltage supply range.

Moved chapter "Device Revision" to section 3.

30.3 Revision 1.20

August 22nd, 2013



Fixed description of ADDRSET, RDSTRB, and WRSTRB fields in EBI Timing section.

Corrected I2C pull-up resistor equation.

Added bus matrix arbitration scheme description.

Added GPIO state retention description.

Updated info page size for Flash memory.

Updated available package options.

Updated product overview section with new parts.

Updated HFXO/LFXO startup description.

Updated the I2C Clock Mode table and added the Maximum Data Hold Time formula.

Added the minimum HFPERCLK requirement for I2C Slave Operation.

Added a new register access type RW1H.

Updated CMU_CALCNT description.

Updated DMA_CHENC register description.

Added LPFMODE recommendation for the ADC Input Filtering.

Updated WRITEONCE bitfield description in MSC_WRITECMD register.

Updated the DMA access description.

Updated trademark, disclaimer and contact information.

Other minor corrections.

30.4 Revision 1.10

April 12th, 2011

Added information about backpowering the MCU if V_{dd} drops below SCL and SDA lines voltage.

Added information on behavior after trying to write to locked pages.

Added information on ACMP warm up with LPREF.

Changed formula in VDDLEVEL bitfield in ACMPn_INPUTSEL.

Added sine wave minimum amplitude to BUFEXTCLK.

Changed description of IRQERASEABORT.

Updated description of WARMUPMODE in ADC section.

Fixed description for REFSEL field in CMU_CALCTRL.

Fixed description of RXDATAV and TXBL interrupt flags in CMU.

Added documentation for DMA_CHREQSTATUS, DMA_CHSREQSTATUS.

Renamed DMA_WAITSTATUS to DMA_CHWAITSTATUS and updated bit fields.



Fixed description of ACMP pin output, the GPIO pin must also be set as output.

Removed reference to the DAC LPF and LPFFREQ and LPFEN bitfields in DACn_CTRL.

Added revision C to Table 3.3 (p. 10).

Changed REFSEL to UPSEL in Figure 11.6 (p. 101).

Added information to the USART chapter that TXTRI is read as 0 if AUTOTRI is set.

Updated general description of bus system.

Updated frequency limitations when clocking TIMER from external source.

Updated information on disabling of individual LCD segment lines.

30.5 Revision 1.00

September 6th, 2010

Changed PCNT_TOP reset value.

Parity bits not available for USART synchronous mode.

Corrected Scaled VDD equation in Section 23.3.4 (p. 332).

DACOUT0 and DACOUT1 in ADCn_SINGLECTRL renamed to DACOOUT0 and DACOOUT1.

CH4 in ADCn_SINGLECTRL under DIFF = 1 renamed to DIFF0.

Changed note about minimum acquisition time when sampling V_{dd}/3 in Section 25.3.4 (p. 352) .

Added information about new individual LCD pin disable feature.

Switched LPFMODE DECAP and RCFILT in ADCn_CTRL register description.

Added EBI Regions and Peripheral Bit Band Alias to System Address Space in Figure 5.2 (p. 16).

Changed VCMP_INPUTCTRL to VCMP_INPUTSEL in Section 24.3.4 (p. 342) , it now complies with register description.

Corrected conversion time numbers in Section 25.3.2 (p. 350).

Changed ENERGYMODE to WARMUPMODE in Section 25.3.3 (p. 351).

Added Result Resolution column in Table 25.3 (p. 357).

Changed ADC calibration routines in Section 25.3.10 (p. 358).

Added table with ADC calibration register effect (Table 25.5 (p. 358)).

Improved ADC Input Filter description and added Figure 25.4 (p. 353).

Added minimum supply voltage restrictions when using the 2.5 V and 5 V bandgap references.

Added note about FULLBIAS and hysteresis level in Section 23.3.2 (p. 331).

Removed V_{ss} as possible negative input selection for the analog comparator in Figure 23.1 (p. 330) .

Improved register description on SCANGAIN, SCANOFFSET, SINGLEGAIN and SINGLEOFFSET fields in ADCn_CAL.



HPROT[3] and HPROT[2] were removed because there is no cache and bufferable implementation in the system.

CHPROT is not only 1 bit for the above reason.

DMA_CONFIG register is W and not RW.

On the PCNT module, the user does not have to issue LTOPBIM command to load TOPB to TOP so this bit has no effect.

Corrected AES 128/256 encryption/decryption duration to 54/75 cycles.

Corrected description of AES byte order for data and key.

QEM in TIMERn_CTRL renamed to QDM.

Changed description of COIST in TIMERn_CCx_CTRL.

Changed DATA0 to CH0DATA and added COMBDATA in PRSEN field description in DACn_CH0CTRL.

Changed DATA1 to CH1DATA and added COMBDATA in PRSEN field description in DACn_CH1CTRL.

Renamed Sine Generation Mode to Sine Generator Mode.

Updated Sine Generator Mode description and added Hi-Z output to Figure 26.3 (p. 377).

Changed Table 7.1 (p. 30), Device Information is not writable by software or debug.

Removed ATESTIN option from INPUTSEL in ADCn_SINGLECTRL.

Corrected reset value for PCNTxCLKEN bits in CMU_PCNTCTRL to 0.

30.6 Revision 0.84

February 19th, 2010

EXTIPSEL16 bitfield in GPIO_EXTIPSELH, renamed to EXTIPSEL15.

AAP information moved from MSC to Debug chapter.

Added description of how to read out device revision number to MSC chapter.

Inserted Links from Register Map to Register Description for each module.

Updated DI table and moved to "Memory and Bus System" Section 5.6 (p. 22).

Updated Section 11.3.3.2 (p. 100) to include information about AUXHFRCO.

EMU_ATESTCTRL register removed.

AUX field in EMU_AUXCTRL renamed to HRCCLR and shrinked to 1 bit.

All DMA channel registers split into separate bit fields.

All PRS channel registers split into separate bit fields.

SINGLEREP in ADCn SINGLECTRL renamed to REP.

SINGLEDIFF in ADCn_SINGLECTRL renamed to DIFF.

SINGLEADJ in ADCn_SINGLECTRL renamed to ADJ.



SINGLERES in ADCn_SINGLECTRL renamed to RES.

SINGLESEL in ADCn_SINGLECTRL renamed to INPUTSEL.

SINGLEREF in ADCn_SINGLECTRL renamed to REF.

SINGLEAT in ADCn_SINGLECTRL renamed to AT.

SINGLEPRSEN in ADCn_SINGLECTRL renamed to PRSEN.

SINGLEPRSSEL in ADCn_SINGLECTRL renamed to PRSSEL.

SCANREP in ADCn_SCANCTRL renamed to REP.

SCANDIFF in ADCn_SCANCTRL renamed to DIFF.

SCANADJ in ADCn_SCANCTRL renamed to ADJ.

SCANRES in ADCn_SCANCTRL renamed to RES.

SCANMASK in ADCn_SCANCTRL renamed to INPUTMASK.

SCANREF in ADCn_SCANCTRL renamed to REF.

SCANAT in ADCn_SCANCTRL renamed to AT.

SCANPRSEN in ADCn_SCANCTRL renamed to PRSEN.

SCANPRSSEL in ADCn_SCANCTRL renamed to PRSSEL.

SINGLEDATA in ADCn SINGLEDATA renamed to DATA.

SCANDATA in ADCn_SCANDATA renamed to DATA.

SINGLEDATA in ADCn_SINGLEDATAP renamed to DATAP.

SCANDATAP in ADCn_SCANDATAP renamed to DATAP.

OSRSEL in ADCn_CTRL renamed to OVSRSEL.

Enumeration of OVSRSEL in ADCn_CTRL changed.

Enumeration of RES in ADCn_SINGLECTRL changed.

Enumeration of RES in ADCn_SCANCTRL changed.

Changed access types for RH registers to R (read only).

Enumeration of UDCTRL in LCD_CTRL changed.

CH0EN in DACn_CH0CTRL renamed to EN.

CHOREFREN in DACn_CHOCTRL renamed to REFFREN.

CH0PRSEN in DACn_CH0CTRL renamed to PRSEN.

CH0PRSSEL in DACn_CH0CTRL renamed to PRSSEL.

CH1EN in DACn_CH1CTRL renamed to EN.

CH1REFREN in DACn_CH1CTRL renamed to REFFREN.



CH1PRSEN in DACn_CH1CTRL renamed to PRSEN.

CH1PRSSEL in DACn_CH1CTRL renamed to PRSSEL.

Enumeration of POSSEL in ACMPn_INPUTSEL changed.

Enumeration of NEGSEL in ACMPn_INPUTSEL changed.

Renamed SWVPEN in GPIO_ROUTE to SWOPEN.

Enumeration of MODE in PCNTn_CTRL changed.

Enumeration of REF in ADCn_SINGLECTRL/ADCn_SCANCTRL changed.

Split DTOGEN in TIMERO_DTOGEN into single bits.

Split DTFSEN in TIMER0_DTFC into single bits.

Split DTFS in TIMER0_DTFAULT into single bits.

Split DTFSC in TIMER0_DTFAULTC into single bits.

DTPRSFSEL0 in TIMER0_DTFC renamed to DTPRS0FSEL.

DTPRSFSEL1 in TIMER0_DTFC renamed to DTPRS1FSEL.

30.7 Revision 0.83

January 25th, 2010

ENERGYMODE bitfield in ADCn_CTRL, renamed to WARMUPMODE.

Updated enumeration for SCANMASK in ADCn_SCANCTRL.

Updated enumeration for SINGLESEL in ADCn_SINGLECTRL.

Updated enumeration for SCANDATASRC in ADCn_STATUS.

Specified default drive strength for GPIO pins in Section 28.3.1 (p. 402) .

Extracted I²C Slave State Machine into separate section (Section 15.3.9 (p. 162)).

Moved specification of resistance values of CSRESEL in ACMPn_CTRL to datasheets.

Corrected DAC clock prescaling equation (Equation 26.1 (p. 375)).

30.8 Revision 0.82

November 20th, 2009

Description of LFXOSEL and LFRCOSEL bits of CMU_STATUS register corrected.

Updated description of EM4 sequence in Table 10.2 (p. 89).

Updated documentation of WORDTIMEOUT and WDATAREADY in MSC_STATUS.

30.9 Revision 0.81

November 13th, 2009



Note added to Section 7.3.5 (p. 32).

Note added to Section 7.3.5 (p. 32).

Internal reference added to Section 5.6 (p. 22) .

DMA_CHx_CTRL register description updated.

Reference to synchronous pin interrupts removed from Chapter 10 (p. 85).

ACMP wakeup triggering updated in Chapter 10 (p. 85).

Internal reference added to note in Section 11.3.1.2 (p. 97) .

Figure 11.4 (p. 100) and Figure 11.5 (p. 100) added.

Section 15.3.7 (p. 155) updated.

Note added in Section 18.3.3 (p. 226).

Section 25.3.6 (p. 354) added and ADCn_BIASPROG register added.

Section 26.3.3 (p. 375) added and DACn_BIASPROG register added.

Section 26.3.8 (p. 377) updated.

Glitch suppression filter added to Figure 28.1 (p. 402), Figure 28.2 (p. 403) and Figure 28.4 (p. 404).

Section 29.3.5 (p. 438) and Section 29.3.6 (p. 439) updated.

LCD_DISPCTRL register updated.

Added PRS example in Section 13.3.4 (p. 131).

Split CCPEN and CDTIPEN bits in TIMERn_ROUTE into CCxPEN and CDTIxPEN bits.

Description and enumeration of EMVREG in EMU_CTRL updated.

30.10 Revision 0.80

October 19th, 2009

Initial preliminary revision.



A Abbreviations

A.1 Abbreviations

This section lists abbreviations used in this document.

Table A.1. Abbreviations

Abbreviation	Description
ACMP	Analog Comparator
ADC	Analog to Digital Converter
АНВ	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".
ALE	Address Latch Enable
AUXHFRCO	Auxiliary High Frequency RC Oscillator.
СС	Compare / Capture
CLK	Clock
CMD	Command
СМИ	Clock Management Unit
CTRL	Control
DAC	Digital to Analog Converter
DBG	Debug
DMA	Direct Memory Access
DRD	Dual Role Device
DTI	Dead Time Insertion
EBI	External Bus Interface
EFM	Energy Friendly Microcontroller
EM	Energy Mode
EM0	Energy Mode 0 (also called active mode)
EM1 to EM4	Energy Mode 1 to Energy Mode 4 (also called low energy modes)
EMU	Energy Management Unit
ENOB	Effective Number of Bits
FS	Full-speed
GPIO	General Purpose Input / Output
HFRCO	High Frequency RC Oscillator
HFXO	High Frequency Crystal Oscillator
HW	Hardware
l ² C	Inter-Integrated Circuit interface
LCD	Liquid Crystal Display
LETIMER	Low Energy Timer



Abbreviation	Description
LEUART	Low Energy Universal Asynchronous Receiver Transmitter
LFRCO	Low Frequency RC Oscillator
LFXO	Low Frequency Crystal Oscillator
LS	Low-speed
MAC	Media Access Controller
NVIC	Nested Vector Interrupt Controller
OSR	Oversampling Ratio
OTG	On-the-go
PCNT	Pulse Counter
PHY	Physical Layer
PRS	Peripheral Reflex System
PWM	Pulse Width Modulation
RC	Resistance and Capacitance
RMU	Reset Management Unit
RTC	Real Time Clock
SAR	Successive Approximation Register
SOF	Start of Frame
SPI	Serial Peripheral Interface
SW	Software
UART	Universal Asynchronous Receiver Transmitter
USART	Universal Synchronous Asynchronous Receiver Transmitter
USB	Universal Serial Bus
VCMP	Voltage supply Comparator
WDOG	Watchdog timer
XTAL	Crystal



B Disclaimer and Trademarks

B.1 Disclaimer

Silicon Laboratories intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Laboratories products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Laboratories reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Laboratories shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products must not be used within any Life Support System without the specific written consent of Silicon Laboratories. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Laboratories products are generally not intended for military applications. Silicon Laboratories products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

B.2 Trademark Information

Silicon Laboratories Inc., Silicon Laboratories, Silicon Labs, SiLabs and the Silicon Labs logo, CMEMS®, EFM, EFM32, EFR, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZMac®, EZRadio®, EZRadioPRO®, DSPLL®, ISOmodem®, Precision32®, ProSLIC®, SiPHY®, USBXpress® and others are trademarks or registered trademarks of Silicon Laboratories Inc. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.



C Contact Information

Silicon Laboratories Inc.

400 West Cesar Chavez Austin, TX 78701

Please visit the Silicon Labs Technical Support web page: http://www.silabs.com/support/pages/contacttechnicalsupport.aspx and register to submit a technical support request.



Table of Contents

1.		Friendly Microcontrollers	
		Typical Applications	
2		EFM32G Development	
۷.		This Document	_
		Related Documentation	
3		Overview	
Ο.		Introduction	
		Block Diagram	
		Features	
		Energy Modes	
		Product Overview	
	3.6.	Device Revision	. 9
4.	System	rocessor	11
		Introduction	
		Features	
_		Functional Description	
5.		y and Bus System	
		Introduction	
	5.Z.	Functional Description	10
		Flash	
		SRAM	
		Device Information (DI) Page	
6		Debug Interface	
٠.		Introduction	
		Features	
		Functional Description	
		Debug Lock and Device Erase	
		Register Map	
		Register Description	
7.		Memory System Controller	
		Introduction	
		Features	
		Functional Description	
		Register Map	
0		Register Description	
ο.		Introduction	
		Features	
		Block Diagram	
		Functional Description	
		Examples	
		Register Map	
	8.7.	Register Description	62
9.	RMU -	Reset Management Unit	79
	9.1.	Introduction	79
		Features	
		Functional Description	
		Register Map	
4.0		Register Description	
10		- Energy Management Unit	
		I. Introduction	
		2. Features	
		I. Register Map	
		5. Register Description	
11		- Clock Management Unit	
•		I. Introduction	
		2. Features	
		3. Functional Description	
		Register Map 1	
			104
12		G - Watchdog Timer 1	
		I. Introduction	
			123
		·	123
		Register Map	
۰.		5. Register Description	
13		Peripheral Reflex System	
	13.1	I. Introduction	ι∠≿



	13.2.	Features	128
		Functional Description	
		Register Map	
		Register Description	
14.	EBI - E	xternal Bus Interface	136
		Introduction	
		Features	
		Functional Description	
		Register Map	
15	14.5.	Register Description	144
15.	151	Introduction	140
		Features	
		Functional Description	
		Register Map	
		Register Description	
16.	USART	- Universal Synchronous Asynchronous Receiver/Transmitter	182
	16.1.	Introduction	182
		Features	
		Functional Description	
	16.4.	Register Map	204
	16.5.	Register Description	204
17.		· Universal Asynchronous Receiver/Transmitter	
		Introduction	
		Features	
		Functional Description	
		Register Description	
4.0	17.5.	Register Map	223
18.	LEUAR	T - Low Energy Universal Asynchronous Receiver/Transmitter	224
		Introduction	
		Features	
		Functional Description	
		Register Description	
10		- Timer/Counter	
19.		Introduction	
		Features	
		Functional Description	
		Register Map	
		Register Description	
20.	RTC -	Real Time Counter	285
	20.1.	Introduction	285
		Features	
		Functional Description	
	20.4.	Register Map	289
	20.5.	Register Description	289
21.		ER - Low Energy Timer	
		Introduction	
		Features	-
		Register Map	
		Register Description	
22		- Pulse Counter	
		Introduction	
		Features	316
	22.3.	Functional Description	316
	22.4.	Register Map	322
		Register Description	
23.	ACMP	- Analog Comparator	329
	23.1.	Introduction	
		Features	329
		Functional Description	330
		Register Map	
0.4		Register Description	
∠ 4.		- Voltage Comparator	
		Features	340
		Functional Description	341
		Register Map	-
		Register Description	
25.		Analog to Digital Converter	
		Introduction	
	25.2.	Features	348
	25.3.	Functional Description	349



	25.4. Register Map	360
	25.5. Register Description	360
26.	DAC - Digital to Analog Converter	373
	26.1. Introduction	
	26.2. Features	373
	26.3. Functional Description	374
	26.4. Register Map	379
	26.5. Register Description	379
27.	AES - Advanced Encryption Standard Accelerator	
	27.1. Introduction	388
	27.2. Features	388
	27.3. Functional Description	388
	27.4. Register Map	392
	27.5. Register Description	392
28.	GPIO - General Purpose Input/Output	400
	28.1. Introduction	400
	28.2. Features	
	28.3. Functional Description	
	28.4. Register Map	407
	28.5. Register Description	408
29.	LCD - Liquid Crystal Display Driver	422
	29.1. Introduction	
	29.2. Features	
	29.3. Functional Description	423
	29.4. Register Map	445
	29.5. Register Description	
30.	Revision History	457
	30.1. Revision 1.31	457
	30.2. Revision 1.30	
	30.3. Revision 1.20	457
	30.4. Revision 1.10	458
	30.5. Revision 1.00	459
	30.6. Revision 0.84	460
	30.7. Revision 0.83	462
	30.8. Revision 0.82	462
	30.9. Revision 0.81	
	30.10. Revision 0.80	
A. <i>A</i>	Abbreviations	464
	A.1. Abbreviations	
В. [Disclaimer and Trademarks	466
	B.1. Disclaimer	
	B.2. Trademark Information	466
C. (Contact Information	
	C 1	467



List of Figures

3.1. Block Diagram of EFM32G	
3.2. Energy Mode Indicator	
3.3. Revision Number Extraction	
5.1. EFM32G Bus System	
5.2. System Address Space	
5.3. Write operation to Low Energy Peripherals	. 20
5.4. Read operation from Low Energy Peripherals	21
6.1. AAP - Authentication Access Port	. 25
6.2. Device Unlock	
8.1. DMA Block Diagram	
8.2. Polling flowchart	
8.3. Ping-pong example	
8.5. Peripheral scatter-gather example	. 48 51
8.6. Memory map for 8 channels, including the alternate data structure	53
8.7. Detailed memory map for the 8 channels, including the alternate data structure	. 54
8.8. channel cfg bit assignments	. 55
9.1. RMU Reset Input Sources and Connections.	80
9.2. RMU Power-on Reset Operation	
9.3. RMU Brown-out Detector Operation	81
10.1. EMU Overview	86
10.2. EMU Energy Mode Transitions	
11.2. CMU Switching from HFRCO to HFXO before HFXO is ready	90
11.3. CMU Switching from HFRCO to HFXO after HFXO is ready	93 90
11.4. HFXO Pin Connection	
11.5. LFXO Pin Connection	100
11.6. HW-support for RC Oscillator Calibration	101
11.7. Single Calibration (CONT=0)	101
13.1. PRS Overview	129
13.2. TIMER0 overflow starting ADC0 single conversions through PRS channel 5.	
14.1. EBI Overview	
14.2. EBI Non-multiplexed 8-bit Data, 8-bit Address Write Operation	138
14.4. EBI Address Latch Setup	
14.5. EBI Multiplexed 16-bit Data, 16-bit Address Read Operation	
14.6. EBI Multiplexed 16-bit Data, 16-bit Address Write Operation	139
14.7. EBI Multiplexed 8-bit Data, 24-bit Address Read Operation	139
14.8. EBI Multiplexed 8-bit Data, 24-bit Address Write Operation	140
14.9. EBI Default Memory Map (ALTMAP = 0)	141
14.10. EBI Alternative Memory Map (ALTMAP = 1)	142
15.1. I C Overview	
15.3. I ² C START and STOP Conditions	150
15.4. I ² C Bit Transfer on I ² C-Bus	
15.5. I ² C Single Byte Write to Slave	151
15.6. I ² C Double Byte Read from Slave	
15.7. I ² C Single Byte Write, then Repeated Start and Single Byte Read	151
15.8. I ² C Master Transmitter/Slave Receiver with 10-bit Address	152
15.9. I ² C Master Receiver/Slave Transmitter with 10-bit Address	152
15.10. I ² C Master State Machine	
15.11. I ² C Slave State Machine	
16.2. USART Asynchronous Frame Format	
16.3. USART Transmit Buffer Operation	
16.4. USART Receive Buffer Operation	
16.5. USART Sampling of Start and Data Bits	191
16.6. USART Sampling of Stop Bits when Number of Stop Bits are 1 or More	
16.7. USART Local Loopback	
16.8. USART Half Duplex Communication with External Driver	
16.9. USART Transmission of Large Frames	195
16.11. USART Reception of Large Frames	
16.12. USART ISO 7816 Data Frame Without Error	197
16.13. USART ISO 7816 Data Frame With Error	198
16.14. USART SmartCard Stop Bit Sampling	
16.15. USART SPI Timing	200
	203
18.1. LEUART Overview	225
18.2. LEUART Asynchronous Frame Format	225



18.3. LEUART Transmitter Overview	
18.4. LEUART Receiver Overview	
18.5. LEUART Local Loopback	. 232
18.6. LEUART Half Duplex Communication with External Driver	
18.7. LEUART - NRZ vs. RZI	
19.1. TIMER Block Overview	
19.2. TIMER Hardware Timer/Counter Control	
19.3. TIMER Clock Selection	
19.4. TIMER Connections	. 253
19.5. TIMER TOP Value Update Functionality	. 253
19.6. TIMER Quadrature Encoded Inputs	. 254
19.7. TIMER Quadrature Decoder Configuration	. 254
19.8. TIMER X2 Decoding Mode	. 255
19.9. TIMER X4 Decoding Mode	
19.10. TIMER Input Pin Logic	. 256
19.11. TIMER Input Capture Buffer Functionality	. 257
19.12. TIMER Output Compare/PWM Buffer Functionality	. 257
19.13. TIMER Input Capture	. 258
19.14. TIMER Period and/or Pulse width Capture	. 258
19.15. TIMER Block Diagram Showing Comparison Functionality	
19.16. TIMER Output Logic	. 259
19.17. TIMER Up-count Frequency Generation	. 260
19.18. TIMER Up-count PWM Generation	. 260
19.19. TIMER Up/Down-count PWM Generation	261
19.20. TIMER Dead-Time Insertion Unit Overview	. 261
19.21. TIMER Triple Half-Bridge	. 262
19.22. TIMER Overview of Dead-Time Insertion Block for a Single PWM channel	. 262
19.23. TIMER Polarity of Both Signals are Set as Active-High	. 263
19.24. TIMER Output Polarities	. 264
20.1. RTC Overview	. 286
21.1. LETIMER Overview	. 295
21.2. LETIMER State Machine for Free-running Mode	. 297
21.3. LETIMER One-shot Repeat State Machine	. 298
21.4. LETIMER Buffered Repeat State Machine	. 299
21.5. LETIMER Double Repeat State Machine	. 300
21.6. LETIMER Simple Waveforms Output	. 302
21.7. LETIMER Repeated Counting	. 302
21.8. LETIMER Dual Output	. 303
21.9. LETIMER Triggered Operation	
21.10. LETIMER Continuous Operation	. 304
21.11. LETIMER LETIMERn_CNT Not Initialized to 0	. 305
22.1. PCNT Overview	. 317
22.2. PCNT Quadrature Coding	
22.3. PCNT Direction Change Interrupt (DIRCNG) Generation	. 321
23.1. ACMP Overview	. 330
23.2. 20 mV Hysteresis Selected	
23.3. Capacitive Sensing Set-up	. 333
24.1. VCMP Overview	
24.2. VCMP 20 mV Hysteresis Enabled	. 342
25.1. ADC Overview	. 350
25.2. ADC Conversion Timing	. 351
25.3. ADC Analog Power Consumption With Different WARMUPMODE Settings	
25.4. ADC RC Input Filter Configuration	
25.5. ADC Bias Programming	. 354
25.6. ADC Conversion Tailgating	
26.1. DAC Overview	. 374
26.2. DAC Bias Programming	. 376
26.3. DAC Sine Mode	
27.1. AES Key and Data Definitions	
27.2. AES Data and Key Orientation as Defined in the Advanced Encryption Standard	
27.3. AES Data and Key Register Operation	. 390
28.1. Pin Configuration	. 402
28.2. Tristated Output with Optional Pull-up or Pull-down	. 403
28.3. Push-Pull Configuration	. 404
28.4. Open-drain	
28.5. Pin n Interrupt Generation	
29.1. LCD Block Diagram	. 423
29.2. LCD Low-power Waveform for LCD_COM0 in Quadruples Multiplex Mode, 1/3 Bias	
29.3. LCD Normal Waveform for LCD_COM0 in Quadruples Multiplex Mode, 1/3 Bias	. 425
29.4. LCD Static Bias and Multiplexing - LCD_COM0	. 425
29.5. LCD 1/2 Bias and Duplex Multiplexing - LCD_COM0	. 426
29.6. LCD 1/2 Bias and Duplex Multiplexing - LCD_COM1	
29.7. LCD 1/2 Bias and Duplex Multiplexing - LCD_SEG0	. 426
29.8. LCD 1/2 Bias and Duplex Multiplexing - LCD_SEG0 Connection	



29.9. LCD 1/2 Bias and Duplex Multiplexing - LCD_SEG0-LCD_COM0	427
29.10. LCD 1/2 Bias and Duplex Multiplexing - LCD_SEG0-LCD_COM1	427
29.11. LCD 1/3 Bias and Duplex Multiplexing - LCD_COM0	427
29.12. LCD 1/3 Bias and Duplex Multiplexing - LCD_COM1	428
29.13. LCD 1/3 Bias and Duplex Multiplexing - LCD_SEG0	428
29.14. LCD 1/3 Bias and Duplex Multiplexing - LCD_SEG0 Connection	428
29.15. LCD 1/3 Bias and Duplex Multiplexing - LCD_SEG0-LCD_COM0	
29.16. LCD 1/3 Bias and Duplex Multiplexing - LCD_SEG0-LCD_COM1	429
29.17. LCD 1/2 Bias and Triplex Multiplexing - LCD_COM0	429
29.18. LCD 1/2 Bias and Triplex Multiplexing - LCD_COM1	429
29.19. LCD 1/2 Bias and Triplex Multiplexing - LCD_COM2	430
29.20. LCD 1/2 Bias and Triplex Multiplexing - LCD_SEG0	
29.21. LCD 1/2 Bias and Triplex Multiplexing - LCD_SEG0 Connection	
29.22. LCD 1/2 Bias and Triplex Multiplexing - LCD_SEG0-LCD_COM0	
29.23. LCD 1/2 Bias and Triplex Multiplexing - LCD_SEG0-LCD_COM1	431
29.24. LCD 1/2 Bias and Triplex Multiplexing - LCD_SEG0-LCD_COM2	431
29.25. LCD 1/3 Bias and Triplex Multiplexing - LCD_COM0	431
29.26. LCD 1/3 Bias and Triplex Multiplexing - LCD_COM1	432
29.27. LCD 1/3 Bias and Triplex Multiplexing - LCD_COM2	432
29.28. LCD 1/3 Bias and Triplex Multiplexing - LCD_SEG0	432
29.29. LCD 1/3 Bias and Triplex Multiplexing - LCD_SEG0 Connection	
29.30. LCD 1/3 Bias and Triplex Multiplexing - LCD_SEG0-LCD_COM0	433
29.31. LCD 1/3 Bias and Triplex Multiplexing - LCD_SEG0-LCD_COM1	433
29.32. LCD 1/3 Bias and Triplex Multiplexing - LCD_SEG0-LCD_COM2	433
29.33. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_COM0	
29.34. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_COM1	434
29.35. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_COM2	434
29.36. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_COM3	434
29.37. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_SEG0	
29.38. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_SEG0 Connection	
29.39. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_SEG0-LCD_COM0	435
29.40. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_SEG0-LCD_COM1	436
29.41. LCD 1/3 Bias and Quadruplex Multiplexing - LCD_SEG0-LCD_COM2	436
29.42. LCD 1/3 Bias and Quadruplex Multiplexing- LCD_SEG0-LCD_COM3	436
29.43. LCD Clock System in LCD Driver	441
29.44 LCD Block Diagram of the Animation Circuit	113



List of Tables

2.1. Register Access Types	
3.1. Energy Mode Description	8
3.2. EFM32G Microcontroller Series	8
3.3. Minor Revision Number Interpretation	
4.1. Interrupt Request Lines (IRQ)	12 17
5.2. Memory System Low Energy Peripherals	17
5.3. Memory System Peripherals	
5.4. Device Information Page Contents	22
7.1. MSC Flash Memory Mapping	30
7.2. Lock Bits Page Structure	31
8.1. AHB bus transfer arbitration interval	
8.2. DMA channel priority	43
8.3. DMA cycle types	45
8.5. channel_cfg for a primary data structure, in peripheral scatter-gather mode	40 50
8.6. Address bit settings for the channel control data structure	53
8.7. src_data_end_ptr bit assignments	
8.8. dst_data_end_ptr bit assignments	55
8.9. channel_cfg bit assignments	55
8.10. DMA cycle of six words using a word increment	58
8.11. DMA cycle of 12 bytes using a halfword increment	59
9.1. RMU Reset Cause Register Interpretation	
10.1. EMU Energy Mode Overview	
10.3. EMU Wakeup Triggers from Low Energy Modes	os
13.1. Reflex Producers	129
13.2. Reflex Consumers	
15.1. I ² C Reserved I ² C Addresses	151
15.2. I ² C High and Low Periods for Low CLKDIV	153
15.3. I ² C Clock Mode	154
15.4. I ² C Interactions in Prioritized Order	
15.5. I ² C Master Transmitter	
15.7. I ² C STATE Values	162
15.8. I ₂ ^C Transmission Status	
15.9. I ² Ç Slave Transmitter	165
15.10. I ² C - Slave Receiver	166
15.11. I ² C Bus Error Response	167
16.1. USART Asynchronous vs. Synchronous Mode	184
16.2. USART Pin Usage	
16.3. USART Data Bits	
16.5. USART Parity Bits	186
16.6. USART Oversampling	
16.7. USART Baud Rates @ 4MHz Peripheral Clock	187
16.8. USART SPI Modes	
16.9. USART IrDA Pulse Widths	
17.1. UART Limitations	
18.1. LEUART Parity Bit	-
19.1. TIMER Counter Response in X2 Decoding Mode	
19.2. TIMER Counter Response in X4 Decoding Mode	
19.3. TIMER Events	
20.1. RTC Resolution Vs Overflow	287
21.1. LETIMER Repeat Modes	
21.2. LETIMER Underflow Output Actions	
22.1. PCNT QUAD Mode Counter Control Function	
23.1. Bias Configuration	
24.1. Bias Configuration	
25.2. ADC Differential Conversion	
25.3. Oversampling Result Shifting and Resolution	
25.4. ADC Results Representation	357
25.5. Calibration Register Effect	358
28.1. Pin Configuration	
29.1. LCD Mux Settings	
29.2. LCD BIAS Settings	
29.4. LCD Contrast	
29.5. LCD Contrast Function	



29.6. LCD Principle of Contrast Adjustment for Different Bias Settings.	438
29.7. LCD V _{LCD}	
29.8. LCD V _{BOOST} Frequency	
29.9. LCD Frame rate Conversion Table	
29.10. LCD Update Data Control (UDCTRL) Bits	
29.11. FCPRESC	
29.12. LCD Animation Shift Register	442
29.13. LCD Animation Pattern	
29.14. LCD Animation Example	443
	161



List of Examples

8.1. DMA Transfer	60
16.1. USART Multi-processor Mode Example	196
19.1. TIMER DTI Example 1	263
19.2. TIMER DTI Example 2	263
21.1. LETIMER Triggered Output Generation	
21.2. LETIMER Continuous Output Generation	304
21.3. LETIMER PWM Output	
21.4. LETIMER PWM Output	
27.1. AES Cipher Block Chaining	391
28.1. GPIO Interrupt Example	406
29.1. LCD Event Frequency Example	441
29.2. LCD Animation Enable Example	444
	444



List of Equations

5.1. Memory SRAM Area Set/Clear Bit	16
5.2. Memory Peripheral Area Bit Modification	. 17
5.3. Memory Wait Cycles with Clock Equal or Faster than HFCORECLK	. 19
5.4. Memory Wait Cycles with Clock Slower than CPU	
12.1. WDOG Timeout Equation	
15.1. I ² C Pull-up Resistor Equation	
15.2. I ² C Maximum Transmission Rate	
15.3. I ² C High and Low Cycles Equations	
15.4. Maximum Data Hold Time	
16.1. USART Baud Rate	186
16.2. USART Desired Baud Rate	186
16.3. USART Synchronous Mode Bit Rate	199
16.4. USART Synchronous Mode Clock Division Factor	199
18.1. LEUART Baud Rate Equation	226
18.2. LEUART CLKDIV Equation	
18.3. LEUART Optimal Sampling Point	230
18.4. LEUART Actual Sampling Point	
19.1. TIMER Rotational Position Equation	
19.2. TIMER Up-count Frequency Generation Equation	
19.3. TIMER Up-count PWM Resolution Equation	
19.4. TIMER Up-count PWM Frequency Equation	
19.5. TIMER Up-count Duty Cycle Equation	
19.6. TIMER Up/Down-count PWM Resolution Equation	
19.7. TIMER Up/Down-count PWM Frequency Equation	261
19.8. TIMER Up/Down-count Duty Cycle Equation	
20.1. RTC Frequency Equation	
21.1. LETIMER Clock Frequency	
23.1. V _{DD} Scaled	
24.1. VCMP V _{DD} Trigger Level	
25.1. ADC Total Conversion Time (in ADC_CLK cycles) Per Output	
25.2. ADC Temperature Measurement	
26.1. DAC Clock Prescaling	
26.2. DAC Single Ended Output Voltage	
26.3. DAC Differential Output Voltage	
26.4. DAC Sine Generation	-
29.1. LCD Frame rate Calculation	-
20.2 LCD Event Frequency Equation	441

SI20S.COM











