Chipsmall

Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

EFM32GG Reference Manual

Giant Gecko Series

- 32-bit ARM Cortex-M3 processor running at up to 48 MHz
- Up to 1024 kB Flash and 128 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 EFM32GG 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. EFM32GG devices consume as little as 219 µA/MHz in run mode.

EFM32GG's low energy consumption outperforms any other available 8-, 16-, and 32-bit solution. The EFM32GG 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 EFM32GG 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 EFM32GG microcontrollers are perfect for:

- Gas metering
- Energy metering
- Water metering
- Smart metering
- Alarm and security systems
- Health and fitness applications
- Industrial and home automation
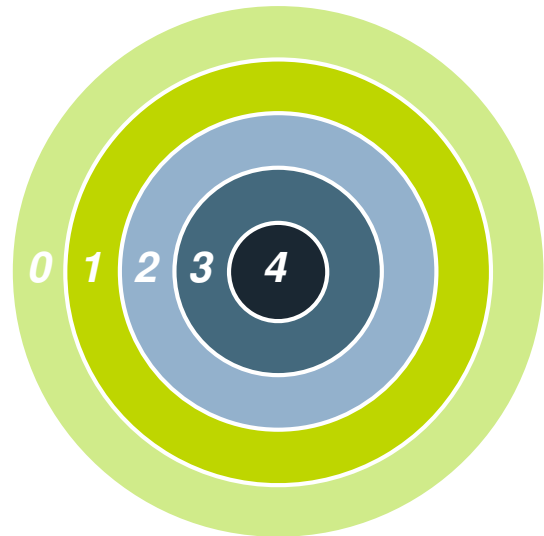
1.2 EFM32GG Development

Because EFM32GG 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 EFM32GG devices ensure easy migration and feature upgrade possibilities.

2 About This Document

This document contains reference material for the EFM32GG series of microcontrollers. All modules and peripherals in the EFM32GG 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 EFM32GG 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 EFM32GG 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 EFM32GG. The color indicates peripheral availability in the different energy modes, described in Section 3.4 (p. 7) .
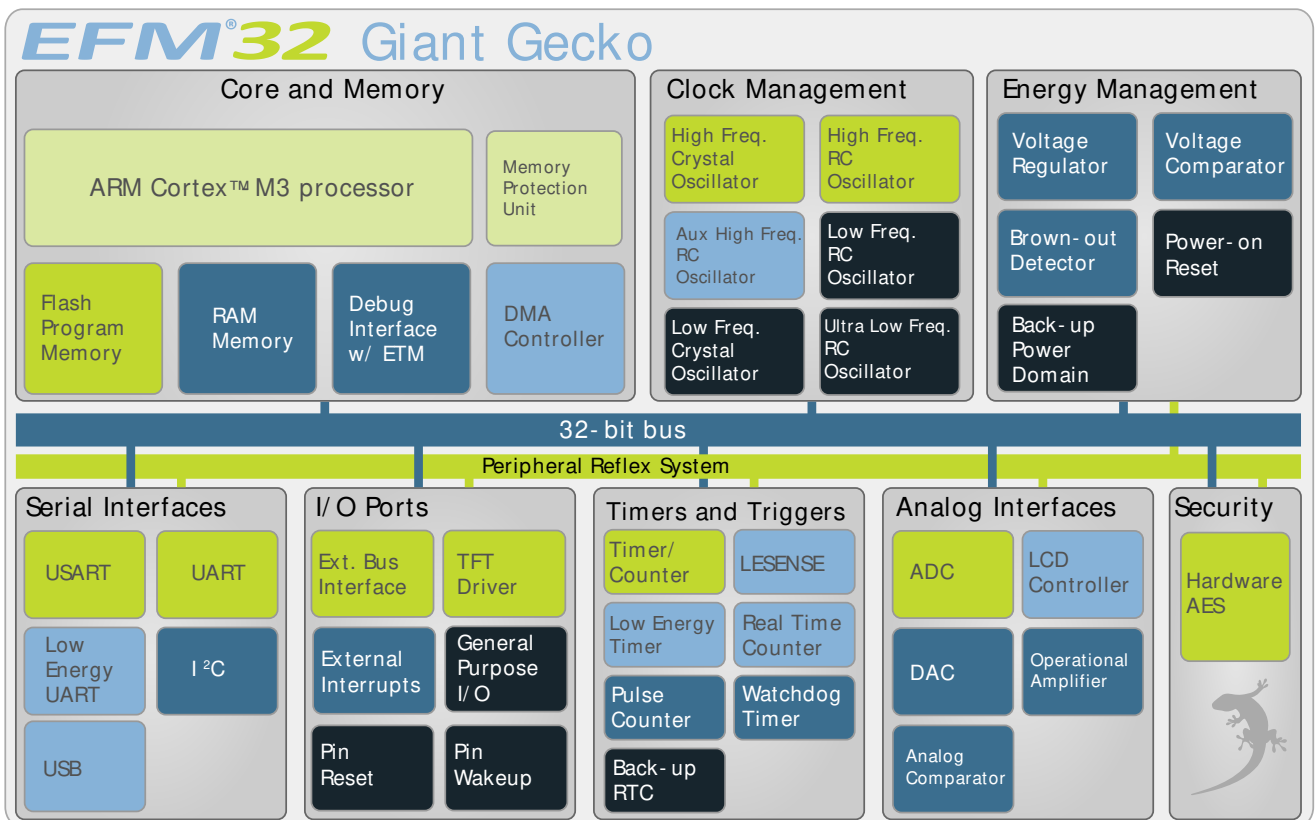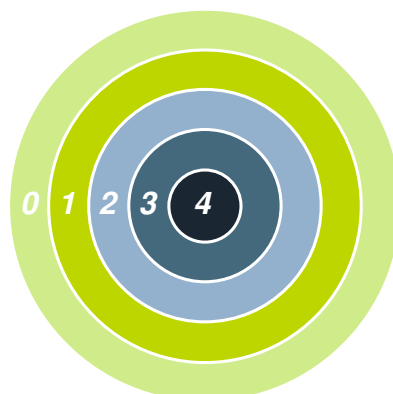
Figure 3.1. Block Diagram of EFM32GG

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 48 MHz
  - Memory Protection Unit
  - Wake-up Interrupt Controller
- **Flexible Energy Management System**
  - 20 nA @ 3 V Shutoff Mode
  - 0.4 µA @ 3 V Shutoff Mode with RTC
  - 0.8 µA @ 3 V Stop Mode, including Power-on Reset, Brown-out Detector, RAM and CPU retention
  - 1.1 µA @ 3 V Deep Sleep Mode, including RTC with 32768 Hz oscillator, Power-on Reset, Brown-out Detector, RAM and CPU retention
  - 80 µA/MHz @ 3 V Sleep Mode
  - 219 µA/MHz @ 3 V Run Mode, with code executed from flash
- **1024/512 KB Flash**
  - Read-while-write support
- **128 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
  - Output state retention and wake-up from Shutoff Mode
- **12 Channel DMA Controller**
  - Alternate/primary descriptors with scatter-gather/ping-pong operation
- **12 Channel Peripheral Reflex System**
  - Autonomous inter-peripheral signaling enables smart operation in low energy modes
- **External Bus Interface (EBI)**
  - Up to 4x256 MB of external memory mapped space
  - TFT Controller supporting Direct Drive
- **Universal Serial Bus (USB)**
  - Fully USB 2.0 compliant
  - On-chip PHY and embedded 5V to 3.3V regulator
- **Integrated LCD Controller for up to 8×36 Segments**
  - Voltage boost, adjustable contrast adjustment and autonomous animation feature
- **Hardware AES with 128/256-bit Keys in 54/75 cycles**
- **Communication interfaces**
  - 3× Universal Synchronous/Asynchronous Receiver/Transmitter
    - UART/SPI/SmartCard (ISO 7816)/IrDA (USART0)/I2S (USART1+USART2)
    - Triple buffered full/half-duplex operation
    - 4-16 data bits
  - 2× Universal Asynchronous Receiver/Transmitter
    - Triple buffered full/half-duplex operation
    - 8-9 data bits
  - 2× Low Energy UART
    - Autonomous operation with DMA in Deep Sleep Mode
  - 2× I^{2}C Interface with SMBus support
    - Address recognition in Stop Mode

- **Timers/Counters**
  - 4× 16-bit Timer/Counter
    - 3 Compare/Capture/PWM channels
    - Dead-Time Insertion on TIMER0
  - 16-bit Low Energy Timer
  - 1× 24-bit and 1× 32-bit Real-Time Counter
  - 3× 8/16-bit Pulse Counter
    - Asynchronous pulse counting/quadrature decoding
  - Watchdog Timer with dedicated RC oscillator @ 50 nA
- **Backup Power Domain**
  - RTC and retention registers in a separate power domain, available in all energy modes
  - Operation from backup battery when main power drains out
- **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
  - Up to 3 Operational Amplifiers
    - Supports rail-to-rail inputs and outputs
    - Programmable gain
  - 2× Analog Comparator
    - Programmable speed/current
    - Capacitive sensing with up to 8 inputs
  - Supply Voltage Comparator
- **Ultra low power sensor interface**
  - Autonomous sensor monitoring in Deep Sleep Mode
  - Wide range of sensors supported, including LC sensors and capacitive buttons

3.3.2 System Features

- **Ultra efficient Power-on Reset and Brown-Out Detector**
- **Debug Interface**
  - 2-pin Serial Wire Debug interface
    - 1-pin Serial Wire Viewer
  - Embedded Trace Module v3.5 (ETM)
- **Temperature range -40 - 85°C**
- **Single power supply 1.98 - 3.8 V**
- **Packages**
  - QFN64
  - TQFP64
  - LQFP100
  - LFBGA112
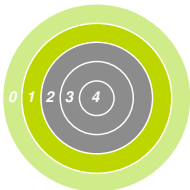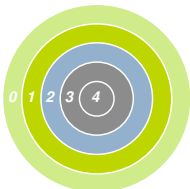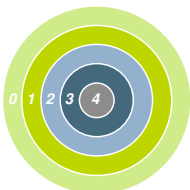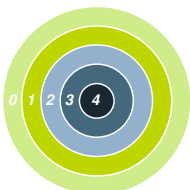 - VFBGA120
  - Full wafer

3.4 Energy Modes

There are five different Energy Modes (EM0-EM4) in the EFM32GG, see Table 3.1 (p. 8). The EFM32GG 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 219 µA/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 80 µA/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, I^2C, LESENSE, OPAMP, USB, WDOG and ACMP) are still available. This gives a high degree of autonomous operation with a current consumption as low as 1.1 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, EM4 wake-up and Brown-out Detection, with a consumption of only 0.8 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, GPIO pin wake-up, GPIO pin retention, Backup RTC (including retention RAM) 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 EFM32GG Microcontroller Series, including peripheral functionality. For more information, the reader is referred to the device specific datasheets.

Table 3.2. EFM32GG Microcontroller Series

EFM32GG Part #	Flash	RAM	GPIO(pins)	USB	LCD	USART+UART	LEUART	I²C	Timer(PWM)	LETIMER	RTC	PCNT	Watchdog	ADC(pins)	DAC(pins)	ACMP(pins)	AES	EBI	LESENSE	Op-Amps	Package
230F512	512	128	56	-	-	3	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	-	Y	3	QFN64

EFM32GG Part #	Flash	RAM	GPIO(pins)	USB	LCD	USART+UART	LEUART	I²C	Timer(PWM)	LETIMER	RTC	PCNT	Watchdog	ADC(pins)	DAC(pins)	ACMP(pins)	AES	EBI	LESENSE	Op-Amps	Package
230F1024	1024	128	56	-	-	3	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	-	Y	3	QFN64
232F512	512	128	53	-	-	3	2	2	4 (11)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	-	Y	3	TQFP64
232F1024	1024	128	53	-	-	3	2	2	4 (11)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	-	Y	3	TQFP64
280F512	512	128	86	-	-	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y	Y	3	LQFP100
280F1024	1024	128	86	-	-	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y	Y	3	LQFP100
290F512	512	128	90	-	-	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y	Y	3	LFBGA112
290F1024	1024	128	90	-	-	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y	Y	3	LFBGA112
295F512	512	128	93	-	-	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y	Y	3	VFBGA120
295F1024	1024	128	93	-	-	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y	Y	3	VFBGA120
330F512	512	128	53	Y	-	3	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (12)	Y	-	Y	3	QFN64
330F1024	1024	128	53	Y	-	3	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (12)	Y	-	Y	3	QFN64
332F512	512	128	50	Y	-	3	2	2	4 (11)	1	1	3	1	1 (8)	2 (2)	1 (4)	Y	-	Y	3	TQFP64
332F1024	1024	128	50	Y	-	3	2	2	4 (11)	1	1	3	1	1 (8)	2 (2)	1 (4)	Y	-	Y	3	TQFP64
380F512	512	128	83	Y	-	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (12)	Y	Y	Y	3	LQFP100
380F1024	1024	128	83	Y	-	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (12)	Y	Y	Y	3	LQFP100
390F512	512	128	87	Y	-	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (12)	Y	Y	Y	3	LFBGA112
390F1024	1024	128	87	Y	-	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (12)	Y	Y	Y	3	LFBGA112
395F512	512	128	93	Y	-	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y	Y	3	VFBGA120
395F1024	1024	128	93	Y	-	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y	Y	3	VFBGA120
840F512	512	128	56	-	8x20	3	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (8)	Y	-	Y	3	QFN64
840F1024	1024	128	56	-	8x20	3	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (8)	Y	-	Y	3	QFN64
842F512	512	128	53	-	8x18	3	2	2	4 (11)	1	1	3	1	1 (8)	2 (2)	2 (8)	Y	-	Y	3	TQFP64
842F1024	1024	128	53	-	8x18	3	2	2	4 (11)	1	1	3	1	1 (8)	2 (2)	2 (8)	Y	-	Y	3	TQFP64
880F512	512	128	86	-	8x36	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y[1]	Y	3	LQFP100
880F1024	1024	128	86	-	8x36	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y[1]	Y	3	LQFP100
890F512	512	128	90	-	8x36	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y[1]	Y	3	LFBGA112

EFM32GG Part #	Flash	RAM	GPIO(pins)	USB	LCD	USART+UART	LEUART	I²C	Timer(PWM)	LETIMER	RTC	PCNT	Watchdog	ADC(pins)	DAC(pins)	ACMP(pins)	AES	EBI	LESENSE	Op-Amps	Package
890F1024	1024	128	90	-	8x36	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y[1]	Y	3	LFBGA112
895F512	512	128	93	-	8x36	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y[1]	Y	3	VFBGA120
895F1024	1024	128	93	-	8x36	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y[1]	Y	3	VFBGA120
900F512	512	128	93	Y	8x36	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y[1]	Y	3	Wafer
900F1024	1024	128	93	Y	8x36	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y[1]	Y	3	Wafer
940F512	512	128	53	Y	8x18	3	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	1 (4)	Y	-	Y	3	QFN64
940F1024	1024	128	53	Y	8x18	3	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	1 (4)	Y	-	Y	3	QFN64
942F512	512	128	50	Y	8x16	3	2	2	4 (11)	1	1	3	1	1 (8)	2 (2)	1 (4)	Y	-	Y	3	TQFP64
942F1024	1024	128	50	Y	8x16	3	2	2	4 (11)	1	1	3	1	1 (8)	2 (2)	1 (4)	Y	-	Y	3	TQFP64
980F512	512	128	83	Y	8x34	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (12)	Y	Y[1]	Y	3	LQFP100
980F1024	1024	128	83	Y	8x34	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (12)	Y	Y[1]	Y	3	LQFP100
990F512	512	128	87	Y	8x34	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (12)	Y	Y[1]	Y	3	LFBGA112
990F1024	1024	128	87	Y	8x34	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (12)	Y	Y[1]	Y	3	LFBGA112
995F512	512	128	93	Y	8x36	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y[1]	Y	3	VFBGA120
995F1024	1024	128	93	Y	8x36	3+2	2	2	4 (12)	1	1	3	1	1 (8)	2 (2)	2 (16)	Y	Y[1]	Y	3	VFBGA120

[1]EBI and LCD overlaps 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]	

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

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

PID1 (0xE00FFFE4)	
31:4	3:0
	Fam[5:2]
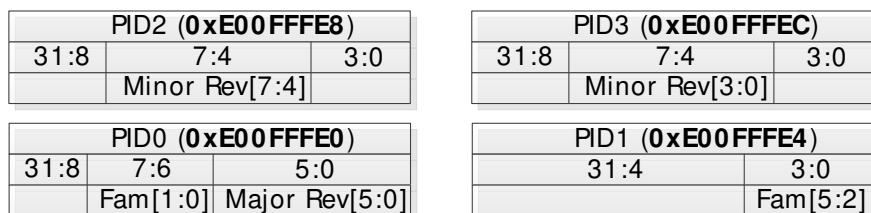
For the latest revision of the Giant Gecko family, the chip family number is 0x02 and the major revision number is 0x01. The minor revision number is to be interpreted according to Table 3.3 (p. 11) .

Table 3.3. Minor Revision Number Interpretation

Minor Rev[7:0]	Revision
0x00	A
0x01	B
0x02	C
0x03	D
0x04	E

3.6.1 Revision Specific Behaviour

Some functional differences exist between the revisions and the user is referred to the errata document for an overview of erratas that apply to the EFM32GG. This document can be found in Simplicity Studio and online at:
http://www.silabs.com/support/pages/document-library.aspx?p=MCUs--32-bit

In addition, there are a couple of differences not covered by any errata, as new functionality is added to later revisions. Those differences are listed here.

3.6.1.1 Revision E

EMU_BUCTRL_BUMODEBODEN is added. This enables the BUBODUNREG-bod to be sensing on BU_VIN in backup mode, and when it senses a brown-out it should trigger a reset (and switch back to main power). Setting this bit on previous revisions will not enable this BOD in backup mode.

4 System Processor
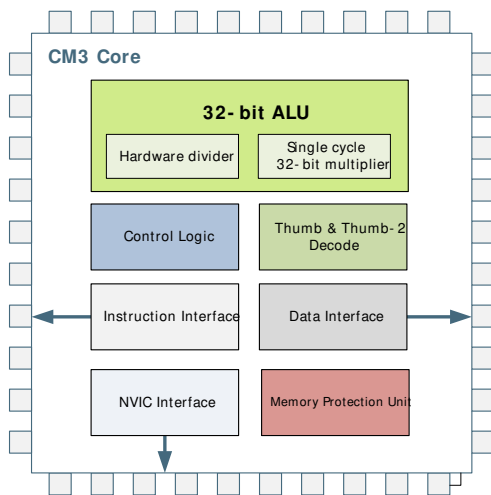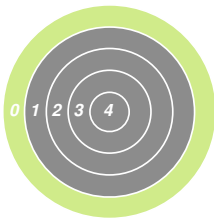
4.1 Introduction

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 r2p1.

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 (r2p1) implementation in the EFM32GG family, the reader is referred to the *EFM32 Cortex-M3 Reference Manual*.

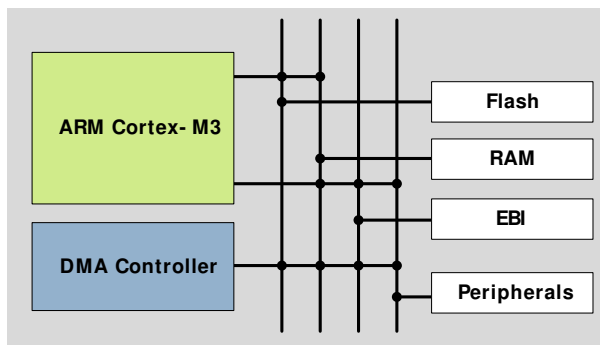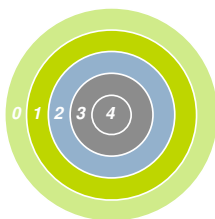## 4.3.1 Interrupt Operation

Figure 4.1. Interrupt Operation
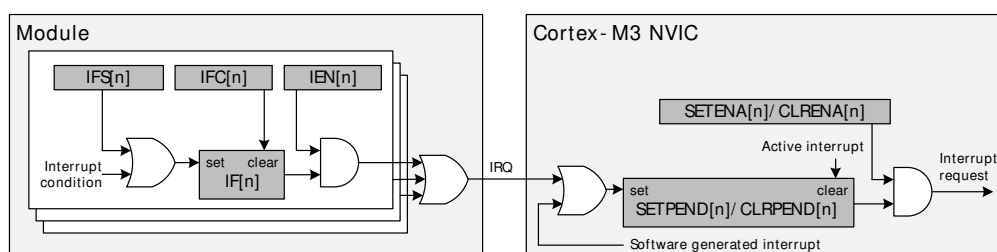
The EFM32GG devices have up to 38 interrupt request lines (IRQ) which are connected to the Cortex-M3. Each of these lines (shown in Table 4.1 (p. 13)) are connected to one or more interrupt flags in one or more modules. The interrupt flags are set by hardware on an interrupt condition. It is also possible 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/CLRPEND bits in ISPR0/ICPR0) in the Cortex-M3 NVIC. The pending bit is then qualified with an enable bit (set/cleared with SETENA/CLRENA bits in ISER0/ICER0) before generating an interrupt request to the core. Figure 4.1 (p. 13) illustrates the interrupt system. For more information on how the interrupts are handled inside the Cortex-M3, the reader is referred to the *EFM32 Cortex-M3 Reference Manual*.

Table 4.1. Interrupt Request Lines (IRQ)

IRQ #	Source
0	DMA
1	GPIO_EVEN
2	TIMER0
3	USART0_RX
4	USART0_TX
5	USB
6	ACMP0/ACMP1
7	ADC0
8	DAC0
9	I2C0

IRQ #	Source
10	I2C1
11	GPIO_ODD
12	TIMER1
13	TIMER2
14	TIMER3
15	USART1_RX
16	USART1_TX
17	LESENSE
18	USART2_RX
19	USART2_TX
20	UART0_RX
21	UART0_TX
22	UART1_RX
23	UART1_TX
24	LEUART0
25	LEUART1
26	LETIMER0
27	PCNT0
28	PCNT1
29	PCNT2
30	RTC
31	BURTC
32	CMU
33	VCMP
34	LCD
35	MSC
36	AES
37	EBI
38	EMU

5 Memory and Bus System

5.1 Introduction

The EFM32GG 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. 16)). 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).
- **USB DMA:** Can access EBI, SRAM and Flash (0x80000000 - 0xDFFFFFFF, 0x00000000 - 0x3FFFFFFF), and the AHB-peripherals: USB and AES.
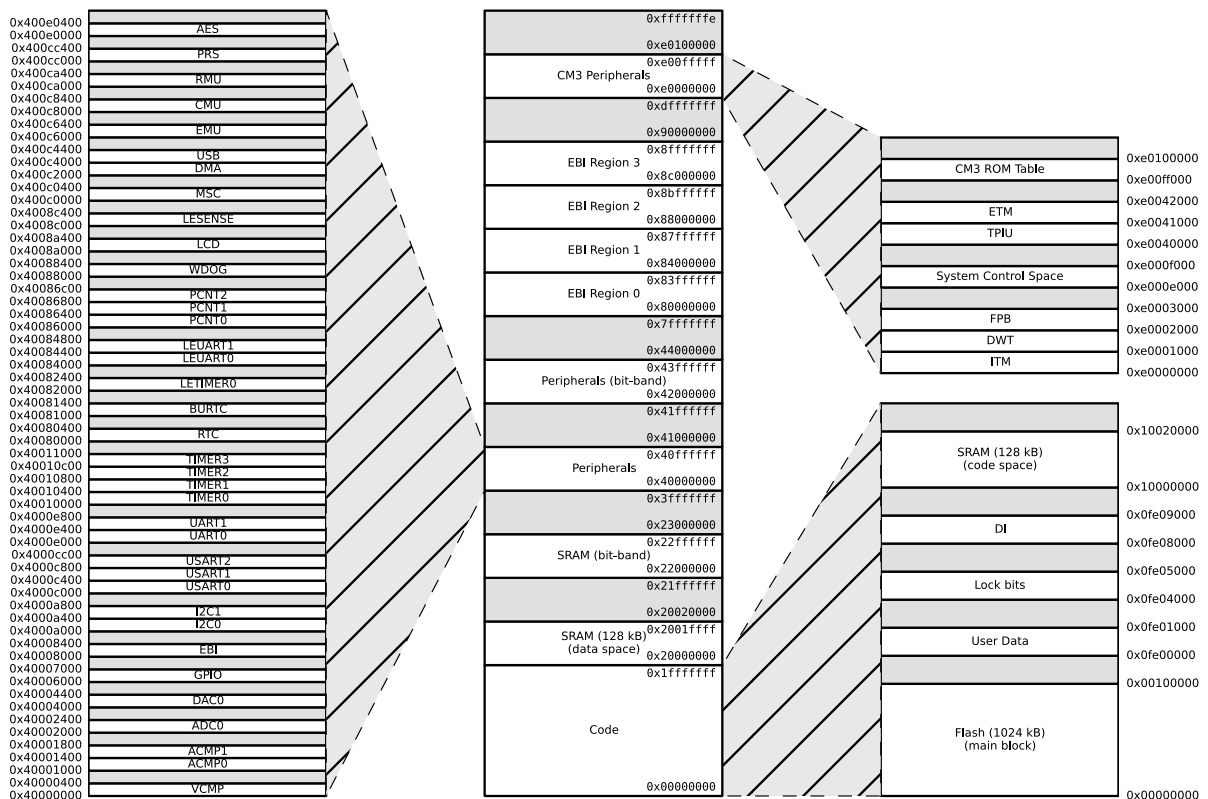
Figure 5.1. EFM32GG Bus System
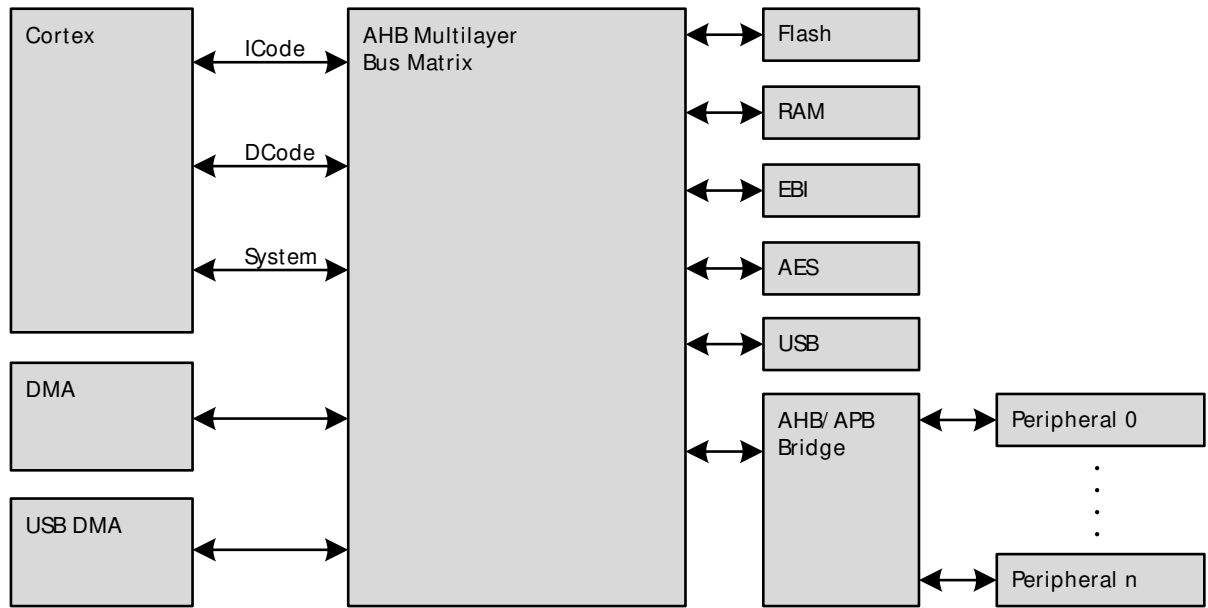
5.2 Functional Description

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. 17)

Figure 5.2. System Address Space

The embedded SRAM is located at address 0x20000000 in the memory map of the EFM32GG. 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 EFM32GG.

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

$$bit\_address = 0x22000000 + (address - 0x20000000) \times 32 + bit \times 4, \tag{5.1}$$

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

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

where *address* and *bit* are defined as above.

Note that the AHB-peripherals USB and 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. 18) , Table 5.2 (p. 18) and Table 5.3 (p. 19) .

Table 5.1. Memory System Core Peripherals

Core peripherals	
Address Range	**Module Name**
0xE0041000 - 0xE0080FFF	ETM
0x400E0000 - 0x400E03FF	AES
0x400CA000 - 0x400CA3FF	RMU
0x400C8000 - 0x400C83FF	CMU
0x400C6000 - 0x400C63FF	EMU
0x400C4000 - 0x400C43FF	USB
0x400C2000 - 0x400C3FFF	DMA
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	I2C0
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
• USB DMA: Access to EBI, SRAM and Flash

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

$$N_{cycles} = 2 + N_{slave\ cycles}, \tag{5.3}$$

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

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

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

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

More details on clocks and prescaling can be found in Chapter 11 (p. 126) .

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
- Low Energy Sensor Interface - LESENSE
- Backup RTC - BURTC

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. There are two different synchronization mechanisms on the Giant Gecko; immediate synchronization, and delayed synchronization. Immediate synchronization is available for the RTC, LETIMER and LESENSE, and results in an immediate update of the target registers. Delayed synchronization is used for the other Low Energy Peripherals, and for these peripherals, a write operation requires 3 positive edges on the clock of the Low Energy Peripheral being accessed. Registers requiring synchronization are marked "Asynchronous" in their description header.

5.3.1.1.1 Delayed synchronization

After writing data to a register which value is to be synchronized into the Low Energy Peripheral using delayed synchronization, a corresponding busy flag in the <module_name>_SYNCBUSY register (e.g. LEUART_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.

See Figure 5.3 (p. 21) for a more detailed overview of the write operation.

Figure 5.3. Write operation to Low Energy Peripherals
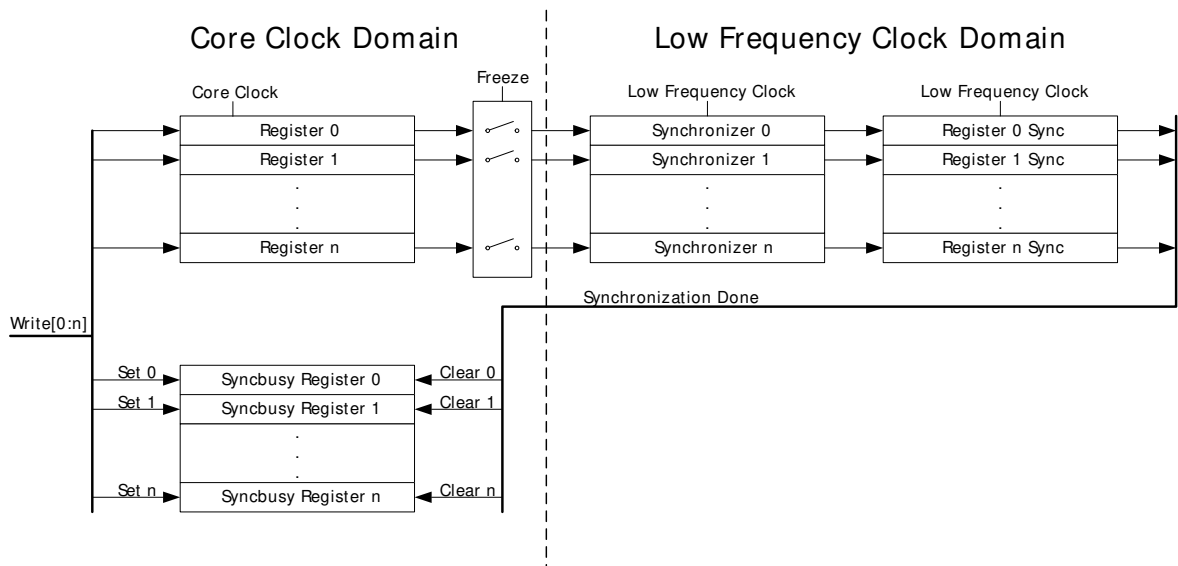
5.3.1.1.2 Immediate synchronization

Contrary to the peripherals with delayed synchronization, data written to peripherals with immediate synchronization, takes effect in the peripheral immediately. They are updated immediately on the peripheral write access. If a write is set up close to a peripheral clock edge, the write is delayed to after the clock edge. This will introduce wait-states on peripheral access. In the worst case, there can be three wait-state cycles of the HFCORECLK_LE and an additional wait-state equivalent of up to 315 ns.

For peripherals with immediate synchronization, the SYNCBUSY registers are still present and serve two purposes: (1) commands written to a peripheral with immediate synchronization are not executed before the first peripheral clock after the write. During this period, the SYNCBUSY flag in the command register is set, indicating that the command has not yet been executed; (2) to maintain backwards compatibility with the EFM32G series, SYNCBUSY registers are also present for other registers. These are however, always 0, indicating that register writes are always safe.

Note

> If the application must be compatible with the EFM32G series, all Low Energy Peripherals should be accessed as if they only had delayed synchronization, i.e. using SYNCBUSY.

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. 22) 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
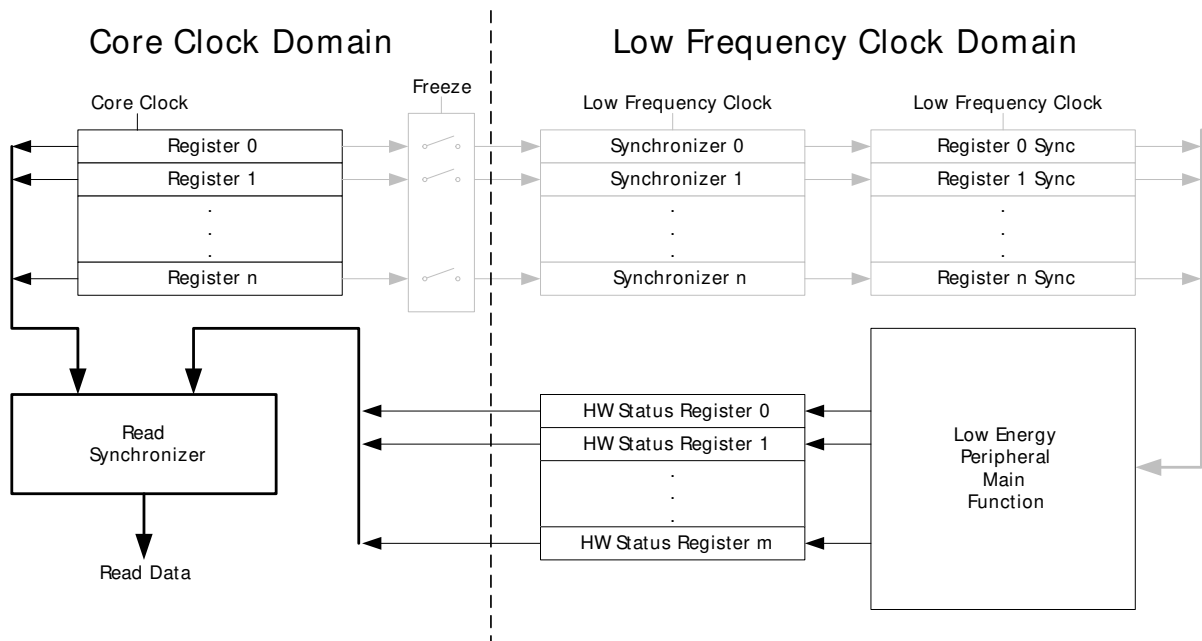
5.3.2 FREEZE register

For Low Energy Peripherals with delayed synchronization 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.

Note

> The FREEZE register is also present on peripherals with immediate synchronization, but has no effect.

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 1024 kB of memory
• Page size of 4096 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 128 kB memory
- Bit-band access support
- 32 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.
0x0FE080A0	DAC0_OPACTRL	Register reset value.
0x0FE080A8	DAC0_OPAOFFSET	Register reset value.
0x0FE080B0	EMU_BUINACT	Register reset value.
0x0FE080B8	EMU_BUACT	Register reset value.
0x0FE080C0	EMU_BUBODBUVINCAL	Register reset value.
0x0FE080C8	EMU_BUBODUNREGCAL	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.

DI Address	Register	Description
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	AUXHFRCO_CALIB_BAND_1	[7:0]: Tuning for the 1.2 MHz AUXHFRCO band.
0x0FE081D5	AUXHFRCO_CALIB_BAND_7	[7:0]: Tuning for the 6.6 MHz AUXHFRCO band.
0x0FE081D6	AUXHFRCO_CALIB_BAND_11	[7:0]: Tuning for the 11 MHz AUXHFRCO band.
0x0FE081D7	AUXHFRCO_CALIB_BAND_14	[7:0]: Tuning for the 14 MHz AUXHFRCO band.
0x0FE081D8	AUXHFRCO_CALIB_BAND_21	[7:0]: Tuning for the 21 MHz AUXHFRCO band.
0x0FE081D9	AUXHFRCO_CALIB_BAND_28	[7:0]: Tuning for the 28 MHz AUXHFRCO band.
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.
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). Ie. 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
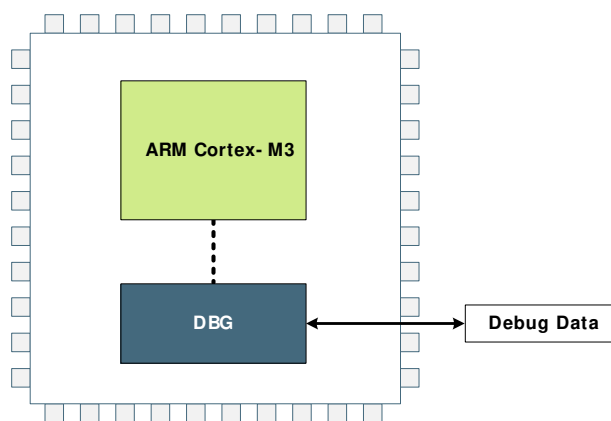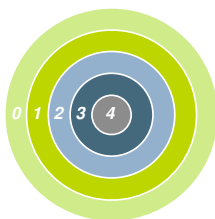
What?

The DBG (Debug Interface) is used to program and debug EFM32GG devices.

Why?

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

How?

The Cortex-M3 supports advanced debugging features. EFM32GG 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 EFM32GG devices include hardware debug support through a 2-pin serial-wire debug (SWD) interface and an Embedded Trace Module (ETM) for data/instruction tracing. In addition, there is also a Serial Wire Viewer pin which can be used to output profiling information, data trace and software-generated messages.

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

• ARM Cortex-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
• Embedded Trace Macrocell v3.5 (ETM)
  • Real time instruction and data trace information of the processor

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: