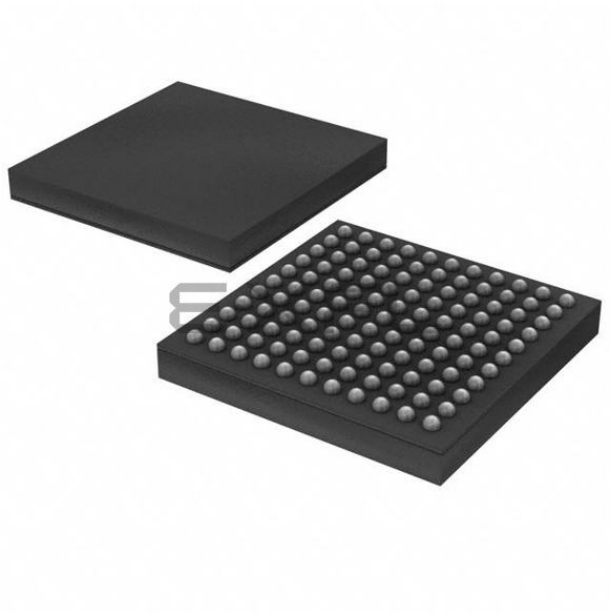


Welcome to E-XFL.COM

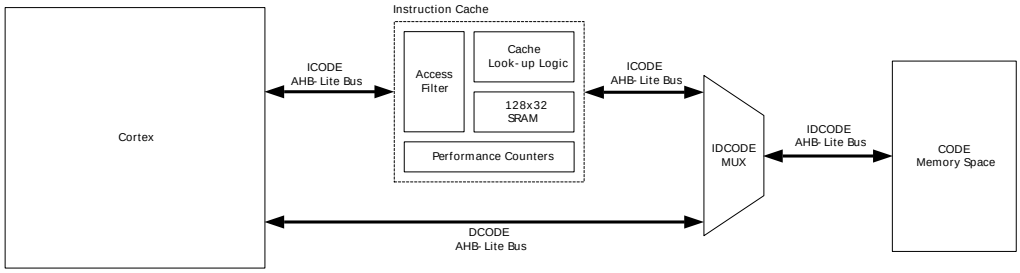
What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Discontinued at Digi-Key
Core Processor	ARM® Cortex®-M3
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	EBI/EMI, I²C, IrDA, SmartCard, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, LCD, POR, PWM, WDT
Number of I/O	93
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128K x 8
Voltage - Supply (Vcc/Vdd)	1.85V ~ 3.8V
Data Converters	A/D 8x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	120-VFBGA
Supplier Device Package	120-BGA (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/efm32gg995f512-bga120

Figure 7.1. Instruction Cache



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

In general it is highly recommended to keep the cache enabled all the time. However, for some sections of code with very low cache hit-rate more energy-efficient execution can be achieved by disabling the cache temporarily. To measure the hit-rate of a code-section, the built-in performance counters can be used. Before the section, start the performance counters by writing 1 to STARTPC in MSC_CMD. This starts the performance counters, counting from 0. At the end of the section, stop the performance counters by writing 1 to STOPPC in MSC_CMD. The number of cache hits and cache misses for that section can then be read from MSC_CACHEHITS and MSC_CACHEMISSES respectively. The total number of 32-bit instruction fetches will be MSC_CACHEHITS + MSC_CACHEMISSES. Thus, the cache hit-ratio can be calculated as MSC_CACHEHITS / (MSC_CACHEHITS + MSC_CACHEMISSES). When MSC_CACHEHITS overflows the CHOF interrupt flag is set. When MSC_CACHEMISSES overflows the CMOF interrupt flag is set. These flags must be cleared explicitly by software. The range of the performance counters can thus be extended by increasing a counter in the MSC interrupt routine. The performance counters only count when a cache lookup is performed. If the lookup fails, MSC_CACHEMISSES is increased. If the lookup is successful, MSC_CACHEHITS is increased. For example, a cache lookup is not performed if the cache is disabled or the code is executed from RAM outside the code space. When caching of vector fetches and instructions in interrupt routines is disabled (ICCDIS in MSC_READCTRL is set), the performance counters do not count when these types of fetches occur (i.e. while in interrupt context).

By default, interrupt vector fetches and instructions in interrupt routines are also cached. Some applications may get better cache utilization by not caching instructions in interrupt context. This is done by setting ICCDIS in MSC_READCTRL. You should only set this bit based on the results from a cache hit ratio measurement. In general, it is recommended to keep the ICCDIS bit cleared. Note that lookups in the cache are still performed, regardless of the ICCDIS setting - but instructions are not cached when cache misses occur inside the interrupt routine. So, for example, if a cached function is called from the interrupt routine, the instructions for that function will be taken from the cache.

The cache content is not retained in EM2, EM3 and EM4. The cache is therefore invalidated regardless of the setting of AIDIS in MSC_READCTRL when entering these energy modes. Applications that switch frequently between EM0 and EM2/3 and execute the very same non-looping code almost every time will most likely benefit from putting this code in RAM. The interrupt vectors can also be put in RAM to reduce current consumption even further.

The cache also supports caching of instruction fetches from the external bus interface (EBI) when accessing the EBI through code space. By default, this is enabled, but it can be disabled by setting EBICDIS in MSC_READCTRL.

7.3.4.7 Instruction Prefetch

The MSC also includes instruction prefetch capability for the internal flash memory. This feature is by default disabled, but can be enabled by setting PREFETCH in MSC_READCTRL. The prefetcher works by the assumption that the next instruction word will be needed in the next fetch. This next word is fetched before the word is actually needed. If it turns out that this next word is actually needed by the CPU, the prefetched word can be returned in one clock cycle, removing the wait-states that would otherwise potentially be needed by a flash read. With the prefetcher enabled, the number of waitstates to the flash when accessing code that is not in the cache is effectively halved.

7.3.5 Erase and Write Operations

The AUXHFRCO is used for timing during flash write and erase operations. To achieve correct timing, the MSC_TIMEBASE register has to be configured according to the settings in CMU_AUXHFRCOCTRL. BASE in MSC_TIMEBASE defines how many AUXCLK cycles - 1 there is in 1 us or 5 us, depending on the configuration of PERIOD. To ensure that timing of flash write and erase operations is within the specification of the flash, the value written to BASE should give at least a 10% margin with respect to the period, i.e. for the 1 us PERIOD, the number of cycles should at least span 1.1 us, and for the 5 us period they should span at least 5.5 us. For the 1 MHz band, PERIOD in MSC_TIMEBASE should be set to 5US, while it should be set to 1US for all other AUXHFRCO bands.

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. To abort an ongoing erase, set the ERASEABORT bit in the MSC_WRITECMD register.

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

During a write or erase, flash read accesses not subject to read-while-write will be stalled, effectively halting code execution from flash. Code execution continues upon write/erase completion. Code residing in RAM may be executed during a write/erase operation regardless of whether read-while-write is enabled or not.

7.4 Register Map

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

Offset	Name	Type	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
0x040	MSC_CMD	W1	Command Register
0x044	MSC_CACHEHITS	R	Cache Hits Performance Counter
0x048	MSC_CACHEMISSES	R	Cache Misses Performance Counter
0x050	MSC_TIMEBASE	RW	Flash Write and Erase Timebase
0x054	MSC_MASSLOCK	RW	Mass Erase 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	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																1
Access																																RW
Name																																BUSFAULT

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future 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 generates bus error response.

Value	Mode	Description
0	GENERATE	A bus fault is generated on access to unmapped code and system space.
1	IGNORE	Accesses to unmapped address space is ignored.

Bit	Name	Description
[13:4]	n_minus_1	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. The 10-bit value indicates the number of DMA transfers, minus one. The possible values are: b000000000 = 1 DMA transfer b000000001 = 2 DMA transfers b000000010 = 3 DMA transfers b000000011 = 4 DMA transfers b000000100 = 5 DMA transfers . . . b111111111 = 1024 DMA transfers. The controller updates this field immediately prior to it entering the arbitration process. This enables the controller to store the number of outstanding DMA transfers that are necessary to complete the DMA cycle.
[3]	next_useburst	Controls if the chnl_useburst_set [C] bit is set to a 1, when the controller is performing a peripheral scatter-gather and is completing a DMA cycle that uses the alternate data structure. ***Note*** Immediately 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. In peripheral scatter-gather DMA cycle then after the DMA cycle that uses the alternate data structure 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 scatter-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. 1 = the controller sets the chnl_useburst_set [C] bit to a 1. Therefore, for the remaining DMA cycles in the peripheral scatter-gather transaction, the controller only responds to requests on `dma_req[ ]`, when it performs a DMA cycle that uses an alternate data structure.
[2:0]	cycle_ctrl	The operating mode of the DMA cycle. The modes are: b000 Stop. Indicates that the data structure is invalid. b001 Basic. The controller must receive a new request, prior to it entering the arbitration process, to enable the DMA cycle to complete. b010 Auto-request. The controller automatically inserts a request for the appropriate channel during the arbitration process. This means that the initial request is sufficient to enable the DMA cycle to complete. b011 Ping-pong. The controller performs a DMA cycle using one of the data structures. After the DMA cycle completes, it performs a DMA cycle using the other data structure. After the DMA cycle completes and provided that the host processor has updated the original data structure, it performs a DMA cycle using the original data structure. The controller continues to perform DMA cycles until it either reads an invalid data structure or the host processor changes the cycle_ctrl bits to b001 or b010. See Section 8.4.2.3.4 (p. 54) . b100 Memory scatter/gather. See Section 8.4.2.3.5 (p. 56) . When the controller operates in memory scatter-gather mode, you must only use this value in the primary data structure. b101 Memory scatter/gather. See Section 8.4.2.3.5 (p. 56) . When the controller operates in memory scatter-gather mode, you must only use this value in the alternate data structure. b110 Peripheral scatter/gather. See Section 8.4.2.3.6 (p. 59) . 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. 59) .

11.5.11 CMU_LFCLKSEL - Low Frequency Clock Select Register

Offset	Bit Position																															
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset												0				0													0x1		0x1	
Access												RW				RW													RW		RW	
Name												LFBE				LFAE													LFB		LFA	

Bit	Name	Reset	Access	Description
31:21	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
20	LFBE	0	RW	**Clock Select for LFB Extended**

This bit redefines the meaning of the LFB field.

Value	Mode	Description
0	DISABLED	LFBCLK is disabled (when LFB = DISABLED).
1	ULFRCO	ULFRCO selected as LFBCLK (when LFB = DISABLED).

19:17	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
16	LFAE	0	RW	**Clock Select for LFA Extended**

This bit redefines the meaning of the LFA field.

Value	Mode	Description
0	DISABLED	LFACLK is disabled (when LFA = DISABLED).
1	ULFRCO	ULFRCO selected as LFACLK (when LFA = DISABLED).

15:4	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
3:2	LFB	0x1	RW	**Clock Select for LFB**

Selects the clock 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 clock 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

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	DDJIT	0	R	**Direct Drive Jitter Interrupt Flag**
	Set when DCLKPERIOD is not met.			
4	DDEMPTY	0	R	**Direct Drive Data Empty Interrupt Flag**
	Set when Direct Drive engine EBI_TFTDD data is empty.			
3	VFPORCH	0	R	**Vertical Front Porch Interrupt Flag**
	Set at beginning of Vertical Front Porch.			
2	VBPORCH	0	R	**Vertical Back Porch Interrupt Flag**
	Set at end of Vertical Back Porch.			
1	HSYNC	0	R	**Horizontal Sync Interrupt Flag**
	Set at Horizontal Sync pulse.			
0	VSYNC	0	R	**Vertical Sync Interrupt Flag**
	Set at Vertical Sync pulse.			

14.5.40 EBI_IFS - Interrupt Flag Set Register

Offset	Bit Position																													
0x09C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0	0	0	0	0	0
Access																											W1	W1	W1	W1	W1	W1
Name																											DDJIT	DDEMPTY	VFPORCH	VBPORCH	HSYNC	VSYNC

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	DDJIT	0	W1	**Direct Drive Jitter Interrupt Flag Set**
	Write to 1 to set Direct Drive Jitter Interrupt flag.			
4	DDEMPTY	0	W1	**Direct Drive Data Empty Interrupt Flag Set**
	Write to 1 to set Direct Drive Data Empty Interrupt flag.			
3	VFPORCH	0	W1	**Vertical Front Porch Interrupt Flag Set**
	Write to 1 to set Vertical Front Porch Interrupt flag.			
2	VBPORCH	0	W1	**Vertical Back Porch Interrupt Flag Set**
	Write to 1 to set Vertical Back Porch Interrupt flag.			
1	HSYNC	0	W1	**Horizontal Sync Interrupt Flag Set**
	Write to 1 to set Horizontal Sync interrupt flag.			
0	VSYNC	0	W1	**Vertical Sync Interrupt Flag Set**
	Write to 1 to set Vertical Sync interrupt flag.			

Offset	Name	Type	Description
...	USB_HCx_INT	RW1H	Host Channel x Interrupt Register
...	USB_HCx_INTMSK	RW	Host Channel x Interrupt Mask Register
...	USB_HCx_TSIZ	RW	Host Channel x Transfer Size Register
...	USB_HCx_DMAADDR	RW	Host Channel x DMA Address Register
0x3C6A0	USB_HC13_CHAR	RWH	Host Channel x Characteristics Register
0x3C6A8	USB_HC13_INT	RW1H	Host Channel x Interrupt Register
0x3C6AC	USB_HC13_INTMSK	RW	Host Channel x Interrupt Mask Register
0x3C6B0	USB_HC13_TSIZ	RW	Host Channel x Transfer Size Register
0x3C6B4	USB_HC13_DMAADDR	RW	Host Channel x DMA Address Register
0x3C800	USB_DCFG	RW	Device Configuration Register
0x3C804	USB_DCTL	RWH	Device Control Register
0x3C808	USB_DSTS	R	Device Status Register
0x3C810	USB_DIEPMSK	RW	Device IN Endpoint Common Interrupt Mask Register
0x3C814	USB_DOEPMSK	RW	Device OUT Endpoint Common Interrupt Mask Register
0x3C818	USB_DAINT	R	Device All Endpoints Interrupt Register
0x3C81C	USB_DAINTMSK	RW	Device All Endpoints Interrupt Mask Register
0x3C828	USB_DVBUSDIS	RW	Device VBUS Discharge Time Register
0x3C82C	USB_DVBUSPULSE	RW	Device VBUS Pulsing Time Register
0x3C834	USB_DIEPEMPMSK	RW	Device IN Endpoint FIFO Empty Interrupt Mask Register
0x3C900	USB_DIEP0CTL	RWH	Device IN Endpoint 0 Control Register
0x3C908	USB_DIEP0INT	RWH	Device IN Endpoint 0 Interrupt Register
0x3C910	USB_DIEP0TSIZ	RW	Device IN Endpoint 0 Transfer Size Register
0x3C914	USB_DIEP0DMAADDR	RW	Device IN Endpoint 0 DMA Address Register
0x3C918	USB_DIEP0TXFSTS	R	Device IN Endpoint 0 Transmit FIFO Status Register
0x3C920	USB_DIEP0_CTL	RWH	Device IN Endpoint x+1 Control Register
0x3C928	USB_DIEP0_INT	RWH	Device IN Endpoint x+1 Interrupt Register
0x3C930	USB_DIEP0_TSIZ	RW	Device IN Endpoint x+1 Transfer Size Register
0x3C934	USB_DIEP0_DMAADDR	RW	Device IN Endpoint x+1 DMA Address Register
0x3C938	USB_DIEP0_TXFSTS	R	Device IN Endpoint x+1 Transmit FIFO Status Register
0x3C940	USB_DIEP1_CTL	RWH	Device IN Endpoint x+1 Control Register
0x3C948	USB_DIEP1_INT	RWH	Device IN Endpoint x+1 Interrupt Register
0x3C950	USB_DIEP1_TSIZ	RW	Device IN Endpoint x+1 Transfer Size Register
0x3C954	USB_DIEP1_DMAADDR	RW	Device IN Endpoint x+1 DMA Address Register
0x3C958	USB_DIEP1_TXFSTS	R	Device IN Endpoint x+1 Transmit FIFO Status Register
0x3C960	USB_DIEP2_CTL	RWH	Device IN Endpoint x+1 Control Register
0x3C968	USB_DIEP2_INT	RWH	Device IN Endpoint x+1 Interrupt Register
0x3C970	USB_DIEP2_TSIZ	RW	Device IN Endpoint x+1 Transfer Size Register
0x3C974	USB_DIEP2_DMAADDR	RW	Device IN Endpoint x+1 DMA Address Register
0x3C978	USB_DIEP2_TXFSTS	R	Device IN Endpoint x+1 Transmit FIFO Status Register
0x3C980	USB_DIEP3_CTL	RWH	Device IN Endpoint x+1 Control Register
0x3C988	USB_DIEP3_INT	RWH	Device IN Endpoint x+1 Interrupt Register
0x3C990	USB_DIEP3_TSIZ	RW	Device IN Endpoint x+1 Transfer Size Register

15.6.55 USB_DIEPx_CTL - Device IN Endpoint x+1 Control Register

The application uses this register to control the behavior of each logical endpoint other than endpoint 0.

Offset	Bit Position																														
0x3C920	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0	0	0	0	0	0	0x0				0		0x0		0	0	0					0x000										
Access	RW1H	RW1H	W1	W1	W1	W1	RW				RW1H		RW		R	R	RW					RW										
Name	EPENA	EPDIS	SETD1PIDOF	SETD0PIDEF	SNAK	CNAK	TXFNUM				STALL		EPTYPE		NAKSTS	DPIDEOF	USBACTEP					MPS										

Bit	Name	Reset	Access	Description
31	EPENA	0	RW1H	**Endpoint Enable**
	In DMA mode for IN endpoints, this bit indicates that data is ready to be transmitted on the endpoint. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done, Endpoint Disabled, Transfer Completed. For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory.			
30	EPDIS	0	RW1H	**Endpoint Disable**
	The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.			
29	SETD1PIDOF	0	W1	**Set DATA1 PID / Odd Frame**
	For bulk and interrupt endpoints writing this field sets the Endpoint Data PID / Even or Odd Frame (DPIDEOF) field in this register to DATA1ODD. For isochronous endpoints writing this field sets the Endpoint Data PID / Even or Odd Frame (DPIDEOF) field to odd (DATA1ODD).			
28	SETD0PIDEF	0	W1	**Set DATA0 PID / Even Frame**
	For bulk and interrupt endpoints writing this field sets the Endpoint Data PID / Even or Odd Frame (DPIDEOF) field in this register to DATA0EVEN. For isochronous endpoints writing this field sets the Endpoint Data PID / Even or Odd Frame (DPIDEOF) field to odd (DATA0EVEN).			
27	SNAK	0	W1	**Set NAK**
	A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint.			
26	CNAK	0	W1	**Clear NAK**
	A write to this bit clears the NAK bit for the endpoint.			
25:22	TXFNUM	0x0	RW	**TxFIFO Number**
	These bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints.			
21	STALL	0	RW1H	**Handshake**
	For bulk and interrupt endpoints: The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. In this case only the application can clear this bit, never the core. When control endpoint: The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.			
20	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
19:18	EPTYPE	0x0	RW	**Endpoint Type**
	This is the transfer type supported by this logical endpoint.			

Value	Mode	Description
0	CONTROL	Control Endpoint.

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
7:1	ADDR	0x00	RW	**Slave address**
	Specifies the slave address of the device.			
0	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		

16.5.7 I2Cn_SADDRMASK - Slave Address Mask Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x00							
Access																									RW							
Name																									MASK							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future 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 bits of the slave address. Setting the mask to 0x00 will match all addresses, while setting it to 0x7F will only match the exact address specified by ADDR.			
0	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		

16.5.8 I2Cn_RXDATA - Receive Buffer Data Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x00							
Access																									R							
Name																									RXDATA							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future 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 read from the receive buffer. Buffer is emptied on read access.			

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 :

LEUART Baud Rate Equation

$$br = fLEUARTn/(1 + LEUARTn\_CLKDIV/256) \quad (19.1)$$

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

$$LEUARTn\_CLKDIV = 256 \times (fLEUARTn/br_{DESIRED} - 1) \quad (19.2)$$

Table 19.2 (p. 500) 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 19.2. 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

19.3.4 Data Transmission

Data transmission is initiated by writing data to the transmit buffer using one of the methods described in Section 19.3.4.1 (p. 500) . 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.

19.3.4.1 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.

The PWM frequency is given by Equation 20.7 (p. 535) :

TIMER 2x Mode PWM Frequency Equation(Up/Down-count)

$$f_{PWM_{2xmode}} = f_{HFPERCLK} / TOP \quad (20.13)$$

The high duty cycle is given by Equation 20.14 (p. 537)

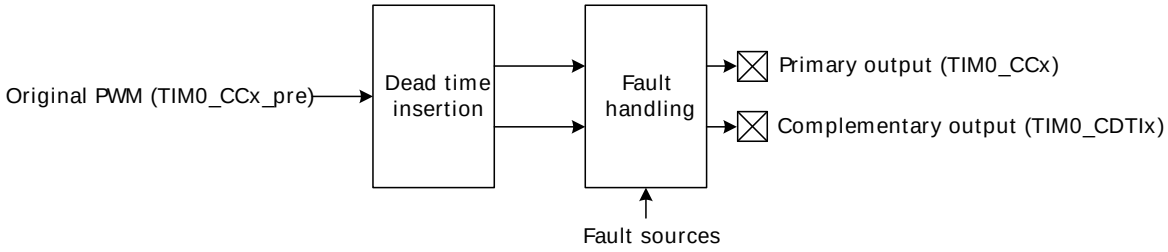
TIMER 2x Mode Duty Cycle Equation

$$DS_{2xmode} = CCVx/TOP \quad (20.14)$$

20.3.3 Dead-Time Insertion Unit (TIMER0 only)

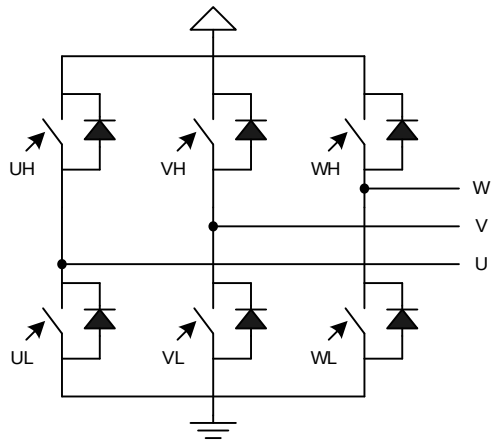
The Dead-Time Insertion Unit aims to make control of BLDC motors safer and more efficient by introducing complementary PWM outputs with dead-time insertion and fault handling, see Figure 20.22 (p. 537) .

Figure 20.22. 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 20.23 (p. 537)). 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 20.23. 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

Bit	Name	Reset	Access	Description
31:11	*Reserved*	*To ensure compatibility with future 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 that a new capture value has pushed an unread value out of the TIMERn_CC2_CCV/TIMERn_CC2_CCVB register pair.			
9	ICBOF1	0	R	**CC Channel 1 Input Capture Buffer Overflow Interrupt Flag**
	This bit indicates that a new capture value has pushed an unread value out of the TIMERn_CC1_CCV/TIMERn_CC1_CCVB register pair.			
8	ICBOF0	0	R	**CC Channel 0 Input Capture Buffer Overflow Interrupt Flag**
	This bit indicates that a new capture value has pushed an unread value out of the TIMERn_CC0_CCV/TIMERn_CC0_CCVB register pair.			
7	*Reserved*	*To ensure compatibility with future 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 that there has been an interrupt event on Compare/Capture channel 2.			
5	CC1	0	R	**CC Channel 1 Interrupt Flag**
	This bit indicates that there has been an interrupt event on Compare/Capture channel 1.			
4	CC0	0	R	**CC Channel 0 Interrupt Flag**
	This bit indicates that there has been an interrupt event on Compare/Capture channel 0.			
3:2	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
1	UF	0	R	**Underflow Interrupt Flag**
	This bit indicates that there has been an underflow.			
0	OF	0	R	**Overflow Interrupt Flag**
	This bit indicates that there has been an overflow.			

20.5.6 TIMERn_IFS - Interrupt Flag Set Register

Offset	Bit Position																												
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																						0	0	0		0	0	0			0	0
Access																						W1	W1	W1		W1	W1	W1			W1	W1
Name																						ICBOF2	ICBOF1	ICBOF0		CC2	CC1	CC0			UF	OF

Bit	Name	Reset	Access	Description
31:11	*Reserved*	*To ensure compatibility with future 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 bit will set Compare/Capture channel 2 input capture buffer overflow interrupt flag.			
9	ICBOF1	0	W1	**CC Channel 1 Input Capture Buffer Overflow Interrupt Flag Set**
	Writing a 1 to this bit will set Compare/Capture channel 1 input capture buffer overflow interrupt flag.			
8	ICBOF0	0	W1	**CC Channel 0 Input Capture Buffer Overflow Interrupt Flag Set**
	Writing a 1 to this bit will set Compare/Capture channel 0 input capture buffer overflow interrupt flag.			
7	*Reserved*	*To ensure compatibility with future 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 bit will set Compare/Capture channel 2 interrupt flag.			
5	CC1	0	W1	**CC Channel 1 Interrupt Flag Set**

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	CNT	0x0000	RWH	**Counter Value**
	These bits hold the counter value.			

20.5.11 TIMERn_ROUTE - I/O Routing Register

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0							0	0	0						0	0	0
Access															RW							RW	RW	RW						RW	RW	RW
Name															LOCATION							CDTI2PEN	CDTI1PEN	CDTI0PEN						CC2PEN	CC1PEN	CC0PEN

Bit	Name	Reset	Access	Description
31:19	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
18:16	LOCATION	0x0	RW	**I/O Location**
	Decides the location of the CC and CDTI pins.			

Value	Mode	Description
0	LOC0	Location 0
1	LOC1	Location 1
2	LOC2	Location 2
3	LOC3	Location 3
4	LOC4	Location 4
5	LOC5	Location 5

15:11	*Reserved*	*To ensure compatibility 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 complementary dead-time insertion output connection to pin.			
9	CDTI1PEN	0	RW	**CC Channel 1 Complementary Dead-Time Insertion Pin Enable**
	Enable/disable CC channel 1 complementary dead-time insertion output connection to pin.			
8	CDTI0PEN	0	RW	**CC Channel 0 Complementary Dead-Time Insertion Pin Enable**
	Enable/disable CC channel 0 complementary dead-time insertion output connection to pin.			
7:3	*Reserved*	*To ensure compatibility 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 connection to pin.			
1	CC1PEN	0	RW	**CC Channel 1 Pin Enable**
	Enable/disable CC channel 1 output/input connection to pin.			
0	CC0PEN	0	RW	**CC Channel 0 Pin Enable**
	Enable/disable CC Channel 0 output/input connection to pin.			

Bit	Name	Reset	Access	Description

Value	Mode	Description
8	PRSCH8	PRS Channel 8 selected as input
9	PRSCH9	PRS Channel 9 selected as input
10	PRSCH10	PRS Channel 10 selected as input
11	PRSCH11	PRS Channel 11 selected as input

15:14 Reserved — *To ensure compatibility with 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 underflow.

Value	Mode	Description
0	NONE	No action on counter underflow
1	TOGGLE	Toggle output on counter underflow
2	CLEAR	Clear output on counter underflow
3	SET	Set output on counter underflow

11:10 COFOA — 0x0 — RW — **Counter Overflow Output Action**

Select output action on counter overflow.

Value	Mode	Description
0	NONE	No action on counter overflow
1	TOGGLE	Toggle output on counter overflow
2	CLEAR	Clear output on counter overflow
3	SET	Set output on counter overflow

9:8 CMOA — 0x0 — RW — **Compare Match Output Action**

Select output action on compare match.

Value	Mode	Description
0	NONE	No action on compare match
1	TOGGLE	Toggle output on compare match
2	CLEAR	Clear output on compare match
3	SET	Set output on compare match

7:5 Reserved — *To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*

4 COIST — 0 — RW — **Compare Output Initial State**

This bit is only used in Output Compare and PWM mode. When this bit is set in compare mode,the output is set high when the counter is disabled. When counting resumes, this value will represent the initial value for the output. If the bit is cleared, the output will be cleared when the counter is disabled. In PWM mode, the output will always be low when disabled, regardless of this bit. However, this bit will represent the initial value of the output, once it is enabled.

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

2 OUTINV — 0 — RW — **Output Invert**

Setting this bit inverts the output from the CC channel (Output compare,PWM).

1:0 MODE — 0x0 — RW — **CC Channel Mode**

These bits select the mode for Compare/Capture channel.

Value	Mode	Description
0	OFF	Compare/Capture channel turned off
1	INPUTCAPTURE	Input capture
2	OUTPUTCOMPARE	Output compare
3	PWM	Pulse-Width Modulation

20.5.19 TIMERn_DTOGEN - DTI Output Generation Enable Register

Offset	Bit Position																													
0x07C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0	0	0	0	0	0
Access																											RW	RW	RW	RW	RW	RW
Name																											DTOGCDTI2EN	DTOGCDTI1EN	DTOGCDTI0EN	DTOGCC2EN	DTOGCC1EN	DTOGCC0EN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write 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 generation for the CDTI2 output from the DTI.			
4	DTOGCDTI1EN	0	RW	**DTI CDTI1 Output Generation Enable**
	This bit enables/disables output generation for the CDTI1 output from the DTI.			
3	DTOGCDTI0EN	0	RW	**DTI CDTI0 Output Generation Enable**
	This bit enables/disables output generation for the CDTI0 output from the DTI.			
2	DTOGCC2EN	0	RW	**DTI CC2 Output Generation Enable**
	This bit enables/disables output generation for the CC2 output from the DTI.			
1	DTOGCC1EN	0	RW	**DTI CC1 Output Generation Enable**
	This bit enables/disables output generation for the CC1 output from the DTI.			
0	DTOGCC0EN	0	RW	**DTI CC0 Output Generation Enable**
	This bit enables/disables output generation for the CC0 output from the DTI.			

20.5.20 TIMERn_DTFAULT - DTI Fault Register

Offset	Bit Position																													
0x080	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0	0	0	0
Access																													R	R	R	R
Name																													DTLOCKUPF	DTDBGF	DTPRS1F	DTPRS0F

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future 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 core lockup fault has occurred and DTLOCKUPFEN is set to 1. The TIMER0_DTFAULTC register can be used to clear fault bits.			
2	DTDBGF	0	R	**DTI Debugger Fault**
	This bit is set to 1 if a debugger fault has occurred and DTDBGFEN is set to 1. The TIMER0_DTFAULTC register can be used to clear fault bits.			
1	DTPRS1F	0	R	**DTI PRS 1 Fault**
	This bit is set to 1 if a PRS 1 fault has occurred and DTPRS1FEN is set to 1. The TIMER0_DTFAULTC register can be used to clear fault bits.			

22.5.15 BURTC_FREEZE - Freeze Register

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

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
0	REGFREEZE	0	RW	**Register Update Freeze**

When set, the update of the BURTC is postponed until this bit is cleared. Use this bit to update several registers simultaneously.

Value	Mode	Description
0	UPDATE	Each write access to an BURTC register is updated into the Low Frequency domain as soon as possible.
1	FREEZE	The BURTC is not updated with the new written value until the freeze bit is cleared.

22.5.16 BURTC_SYNCBUSY - Synchronization Busy Register

Offset	Bit Position																														
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0	0
Access																															R	R
Name																															COMP0	LPMODE

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write bits to 0. More information in Section 2.1 (p. 3)*		
1	COMP0	0	R	**COMP0 Register Busy**
	Set when the value written to COMP0 is being synchronized.			
0	LPMODE	0	R	**LPMODE Register Busy**
	Set when the value written to LPMODE is being synchronized.			

22.5.17 RETx_REG - Retention Register

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

23.5.4 LETIMERn_CNT - Counter Value Register

Offset	Bit Position																															
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0000															
Access																	RWH															
Name																	CNT															

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	CNT	0x0000	RWH	**Counter Value**
	Use to read the current value of the LETIMER.			

23.5.5 LETIMERn_COMP0 - Compare Value Register 0 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 20) .

Offset	Bit Position																															
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0000															
Access																	RW															
Name																	COMP0															

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	COMP0	0x0000	RW	**Compare Value 0**
	Compare and optionally top value for LETIMER			

23.5.6 LETIMERn_COMP1 - Compare Value Register 1 (Async Reg)

For more information about Asynchronous Registers please see Section 5.3 (p. 20) .

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.

26.3.2 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 26.1 (p. 671) 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 26.1. Bias Configuration

BIASPROG	Bias Current (µA), 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

26.3.3 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 26.2 (p. 672)). 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.

Bit	Name	Reset	Access	Description

Value	Mode	Description
0x01	A0	Enable em4 wakeup on pin A0
0x02	A6	Enable em4 wakeup on pin A6
0x04	C9	Enable em4 wakeup on pin C9
0x08	F1	Enable em4 wakeup on pin F1
0x10	F2	Enable em4 wakeup on pin F2
0x20	E13	Enable em4 wakeup on pin E13

32.5.24 GPIO_EM4WUPOL - EM4 Wake-up Polarity Register

Offset	Bit Position																													
0x138	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x00				
Access																											RW				
Name																											EM4WUPOL				

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)*		
5:0	EM4WUPOL	0x00	RW	**EM4 Wake-up Polarity**

Write bit n to 1 for high wake-up request. Write bit n to 0 for low wake-up request

Value	Mode	Description
0x01	A0	Determines polarity on pin A0
0x02	A6	Determines polarity on pin A6
0x04	C9	Determines polarity on pin C9
0x08	F1	Determines polarity on pin F1
0x10	F2	Determines polarity on pin F2
0x20	E13	Determines polarity on pin E13

32.5.25 GPIO_EM4WUCAUSE - EM4 Wake-up Cause Register

Offset	Bit Position																													
0x13C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x00				
Access																											R				
Name																											EM4WUCAUSE				

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)*		
5:0	EM4WUCAUSE	0x00	R	**EM4 wake-up cause**

Bit n indicates which pin the wake-up request occurred.

List of Tables