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	Active
Core Processor	HCS12X
Core Size	16-Bit
Speed	50MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	119
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 24x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LQFP
Supplier Device Package	144-LQFP (20x20)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=s912xeq512bcag

Chapter 22
Timer Module (TIM16B8CV2) Block Description

Chapter 23
Voltage Regulator (S12VREGL3V3V1)

  - Flexible identifier filter programmable as 2 x 32 bit, 4 x 16 bit, or 8 x 8 bit
  - Four separate interrupt channels for Rx, Tx, error, and wake-up
  - Low-pass filter wake-up function
  - Loop-back for self-test operation
- ECT (enhanced capture timer)
  - 8 x 16-bit channels for input capture or output compare
  - 16-bit free-running counter with 8-bit precision prescaler
  - 16-bit modulus down counter with 8-bit precision prescaler
  - Four 8-bit or two 16-bit pulse accumulators
- TIM (standard timer module)
  - 8 x 16-bit channels for input capture or output compare
  - 16-bit free-running counter with 8-bit precision prescaler
  - 1 x 16-bit pulse accumulator
- PIT (periodic interrupt timer)
  - Up to eight timers with independent time-out periods
  - Time-out periods selectable between 1 and 2^{24} bus clock cycles
  - Time-out interrupt and peripheral triggers
- 8 PWM (pulse-width modulator) channels
  - 8 channel x 8-bit or 4 channel x 16-bit Pulse Width Modulator
  - programmable period and duty cycle per channel
  - Center- or left-aligned outputs
  - Programmable clock select logic with a wide range of frequencies
  - Fast emergency shutdown input
- Three Serial Peripheral Interface Modules (SPI)
  - Configurable for 8 or 16-bit data size
- Eight Serial Communication Interfaces (SCI)
  - Standard mark/space non-return-to-zero (NRZ) format
  - Selectable IrDA 1.4 return-to-zero-inverted (RZI) format with programmable pulse widths
- Two Inter-IC bus (IIC) Modules
  - Multi-master operation
  - Software programmable for one of 256 different serial clock frequencies
  - Broadcast mode support
  - 10-bit address support
- On-Chip Voltage Regulator
  - Two parallel, linear voltage regulators with bandgap reference
  - Low-voltage detect (LVD) with low-voltage interrupt (LVI)
  - Power-on reset (POR) circuit
  - 3.3V and 5V range operation
  - Low-voltage reset (LVR)

Table 1-14. Interrupt Vector Locations (Sheet 1 of 4)

Vector Address[1]	XGATE Channel ID[2]	Interrupt Source	CCR Mask	Local Enable	STOP Wake up	WAIT Wake up
Vector base + $F8	—	Unimplemented instruction trap	None	None	—	—
Vector base+ $F6	—	SWI	None	None	—	—
Vector base+ $F4	—	$\overline{\text{XIRQ}}$	X Bit	None	Yes	Yes
Vector base+ $F2	—	$\overline{\text{IRQ}}$	I bit	IRQCR (IRQEN)	Yes	Yes
Vector base+ $F0	$78	Real time interrupt	I bit	CRGINT (RTIE)	Refer to CRG interrupt section	
Vector base+ $EE	$77	Enhanced capture timer channel 0	I bit	TIE (C0I)	No	Yes
Vector base + $EC	$76	Enhanced capture timer channel 1	I bit	TIE (C1I)	No	Yes
Vector base+ $EA	$75	Enhanced capture timer channel 2	I bit	TIE (C2I)	No	Yes
Vector base+ $E8	$74	Enhanced capture timer channel 3	I bit	TIE (C3I)	No	Yes
Vector base+ $E6	$73	Enhanced capture timer channel 4	I bit	TIE (C4I)	No	Yes
Vector base+ $E4	$72	Enhanced capture timer channel 5	I bit	TIE (C5I)	No	Yes
Vector base + $E2	$71	Enhanced capture timer channel 6	I bit	TIE (C6I)	No	Yes
Vector base+ $E0	$70	Enhanced capture timer channel 7	I bit	TIE (C7I)	No	Yes
Vector base+ $DE	$6F	Enhanced capture timer overflow	I bit	TSRC2 (TOF)	No	Yes
Vector base+ $DC	$6E	Pulse accumulator A overflow	I bit	PACTL (PAOVI)	No	Yes
Vector base + $DA	$6D	Pulse accumulator input edge	I bit	PACTL (PAI)	No	Yes
Vector base + $D8	$6C	SPI0	I bit	SPI0CR1 (SPIE, SPTIE)	No	Yes
Vector base+ $D6	$6B	SCI0	I bit	SCI0CR2 (TIE, TCIE, RIE, ILIE)	Yes	Yes
Vector base + $D4	$6A	SCI1	I bit	SCI1CR2 (TIE, TCIE, RIE, ILIE)	Yes	Yes
Vector base + $D2	$69	ATD0	I bit	ATD0CTL2 (ASCIE)	Yes	Yes
Vector base + $D0	$68	ATD1	I bit	ATD1CTL2 (ASCIE)	Yes	Yes
Vector base + $CE	$67	Port J	I bit	PIEJ (PIEJ7-PIEJ0)	Yes	Yes
Vector base + $CC	$66	Port H	I bit	PIEH (PIEH7-PIEH0)	Yes	Yes
Vector base + $CA	$65	Modulus down counter underflow	I bit	MCCTL(MCZI)	No	Yes
Vector base + $C8	$64	Pulse accumulator B overflow	I bit	PBCTL(PBOVI)	No	Yes
Vector base + $C6	$63	CRG PLL lock	I bit	CRGINT(LOCKIE)	Refer to CRG interrupt section	
Vector base + $C4	$62	CRG self-clock mode	I bit	CRGINT (SCMIE)	Refer to CRG interrupt section	
Vector base + $C2	$61	SCI6	I bit	SCI6CR2 (TIE, TCIE, RIE, ILIE)	Yes	Yes
Vector base + $C0	$60	IIC0 bus	I bit	IBCR0 (IBIE)	No	Yes

Table 1-14. Interrupt Vector Locations (Sheet 2 of 4)

Vector Address[1]	XGATE Channel ID[2]	Interrupt Source	CCR Mask	Local Enable	STOP Wake up	WAIT Wake up
Vector base + $BE	$5F	SPI1	I bit	SPI1CR1 (SPIE, SPTIE)	No	Yes
Vector base + $BC	$5E	SPI2	I bit	SPI2CR1 (SPIE, SPTIE)	No	Yes
Vector base + $BA	$5D	FLASH Fault Detect	I bit	FCNFG2 (FDIE)	No	No
Vector base + $B8	$5C	FLASH	I bit	FCNFG (CCIE, CBEIE)	No	Yes
Vector base + $B6	$5B	CAN0 wake-up	I bit	CAN0RIER (WUPIE)	Yes	Yes
Vector base + $B4	$5A	CAN0 errors	I bit	CAN0RIER (CSCIE, OVRIE)	No	Yes
Vector base + $B2	$59	CAN0 receive	I bit	CAN0RIER (RXFIE)	No	Yes
Vector base + $B0	$58	CAN0 transmit	I bit	CAN0TIER (TXEIE[2:0])	No	Yes
Vector base + $AE	$57	CAN1 wake-up	I bit	CAN1RIER (WUPIE)	Yes	Yes
Vector base + $AC	$56	CAN1 errors	I bit	CAN1RIER (CSCIE, OVRIE)	No	Yes
Vector base + $AA	$55	CAN1 receive	I bit	CAN1RIER (RXFIE)	No	Yes
Vector base + $A8	$54	CAN1 transmit	I bit	CAN1TIER (TXEIE[2:0])	No	Yes
Vector base + $A6	$53	CAN2 wake-up	I bit	CAN2RIER (WUPIE)	Yes	Yes
Vector base + $A4	$52	CAN2 errors	I bit	CAN2RIER (CSCIE, OVRIE)	No	Yes
Vector base + $A2	$51	CAN2 receive	I bit	CAN2RIER (RXFIE)	No	Yes
Vector base + $A0	$50	CAN2 transmit	I bit	CAN2TIER (TXEIE[2:0])	No	Yes
Vector base + $9E	$4F	CAN3 wake-up	I bit	CAN3RIER (WUPIE)	Yes	Yes
Vector base+ $9C	$4E	CAN3 errors	I bit	CAN3RIER (CSCIE, OVRIE)	No	Yes
Vector base+ $9A	$4D	CAN3 receive	I bit	CAN3RIER (RXFIE)	No	Yes
Vector base + $98	$4C	CAN3 transmit	I bit	CAN3TIER (TXEIE[2:0])	No	Yes
Vector base + $96	$4B	CAN4 wake-up	I bit	CAN4RIER (WUPIE)	Yes	Yes
Vector base + $94	$4A	CAN4 errors	I bit	CAN4RIER (CSCIE, OVRIE)	No	Yes
Vector base + $92	$49	CAN4 receive	I bit	CAN4RIER (RXFIE)	No	Yes
Vector base + $90	$48	CAN4 transmit	I bit	CAN4TIER (TXEIE[2:0])	No	Yes
Vector base + $8E	$47	Port P Interrupt	I bit	PIEP (PIEP7-PIEP0)	Yes	Yes
Vector base+ $8C	$46	PWM emergency shutdown	I bit	PWMSDN (PWMIE)	No	Yes

2.3.37 Port M Data Register (PTM)

Address 0x0250 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R W	PTM7	PTM6	PTM5	PTM4	PTM3	PTM2	PTM1	PTM0
Altern. Function	TXCAN3	RXCAN3	TXCAN2	RXCAN2	TXCAN1	RXCAN1	TXCAN0	RXCAN0
	—	—	(TXCAN0)	(RXCAN0)	(TXCAN0)	(RXCAN0)	—	—
	(TXCAN4)	(RXCAN4)	(TXCAN4)	(RXCAN4)	—	—	—	—
	—	—	(SCK0)	(MOSI0)	($\overline{SS0}$)	(MISO0)	—	—
	TXD3	RXD3	—	—	—	—	—	—
Reset	0	0	0	0	0	0	0	0

Figure 2-35. Port M Data Register (PTM)

1. Read: Anytime.
   Write: Anytime.

Table 2-33. PTM Register Field Descriptions

Field	Description
7-6 PTM	**Port M general purpose input/output data**—Data Register Port M pins 7 and 6 are associated with TXCAN and RXCAN signals of CAN3 and the routed CAN4, as well as with TXD and RXD signals of SCI3, respectively. The CAN3 function takes precedence over the CAN4, SCI3 and the general purpose I/O function if the CAN3 module is enabled. The CAN4 function takes precedence over the SCI3 and the general purpose I/O function if the CAN4 module is enabled. The SCI3 function takes precedence over the general purpose I/O function if the SCI3 module is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.
5 PTM	**Port M general purpose input/output data**—Data Register Port M pin 5 is associated with the TXCAN signal of CAN2 and the routed CAN4 and CAN0, as well as with SCK signals of SPI0. The CAN2 function takes precedence over the routed CAN0, routed CAN4, the routed SPI0 and the general purpose I/O function if the CAN2 module is enabled. The routed CAN0 function takes precedence over the routed CAN4, the routed SPI0 and the general purpose I/O function if the routed CAN0 module is enabled. The routed CAN4 function takes precedence over the routed SPI0 and general purpose I/O function if the routed CAN4 module is enabled. The routed SPI0 function takes precedence of the general purpose I/O function if the routed SPI0 is enabled. When not used with the alternative function, this pin can be used as general purpose I/O. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

1. Read: Anytime.
   Write: Anytime.

Table 2-51. DDRH Register Field Descriptions

Field	Description
7 DDRH	**Port H data direction**— This register controls the data direction of pin 7. The enabled SCI5 forces the I/O state to be an output. Depending on the configuration of the enabled routed SPI2 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
6 DDRH	**Port H data direction**— This register controls the data direction of pin 6. The enabled SCI5 forces the I/O state to be an input. Depending on the configuration of the enabled routed SPI2 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
5 DDRH	**Port H data direction**— This register controls the data direction of pin 5. The enabled SCI4 forces the I/O state to be an output. Depending on the configuration of the enabled routed SPI2 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
4 DDRH	**Port H data direction**— This register controls the data direction of pin 4. The enabled SCI4 forces the I/O state to be an input. Depending on the configuration of the enabled routed SPI2 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
3 DDRH	**Port H data direction**— This register controls the data direction of pin 3. The enabled SCI7 forces the I/O state to be an output. Depending on the configuration of the enabled routed SPI1 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
2 DDRH	**Port H data direction**— This register controls the data direction of pin 2. The enabled SCI7 forces the I/O state to be an input. Depending on the configuration of the enabled routed SPI1 this pin will be forced to be input or output. In those cases the data direction bits will not change. The DDRM bits revert to controlling the I/O direction of a pin when the associated peripheral module is disabled. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

Expansion of the BDM Local Address Map

PPAGE, RPAGE, and EPAGE registers are also used for the expansion of the BDM local address to the global address. These registers can be read and written by the BDM.

The BDM expansion scheme is the same as the CPU expansion scheme.

3.4.2.2 Global Addresses Based on the Global Page

CPU Global Addresses Based on the Global Page

The seven global page index bits allow access to the full 8 Mbyte address map that can be accessed with 23 address bits. This provides an alternative way to access all of the various pages of FLASH, RAM and EEE as well as additional external memory.

The GPAGE Register is used only when the CPU is executing a global instruction (see Section 3.3.2.3, “Global Page Index Register (GPAGE)). The generated global address is the result of concatenation of the CPU local address [15:0] with the GPAGE register [22:16] (see Figure 3-7).

BDM Global Addresses Based on the Global Page

The seven BDMGPR Global Page index bits allow access to the full 8 Mbyte address map that can be accessed with 23 address bits. This provides an alternative way to access all of the various pages of FLASH, RAM and EEE as well as additional external memory.

The BDM global page index register (BDMGPR) is used only in the case the CPU is executing a firmware command which uses a global instruction (like GLDD, GSTD) or by a BDM hardware command (like WRITE_W, WRITE_BYTE, READ_W, READ_BYTE). See the BDM Block Guide for further details.

The generated global address is a result of concatenation of the BDM local address with the BDMGPR register [22:16] in the case of a hardware command or concatenation of the CPU local address and the BDMGPR register [22:16] in the case of a firmware command (see Figure 3-18).

Address: 0x0128

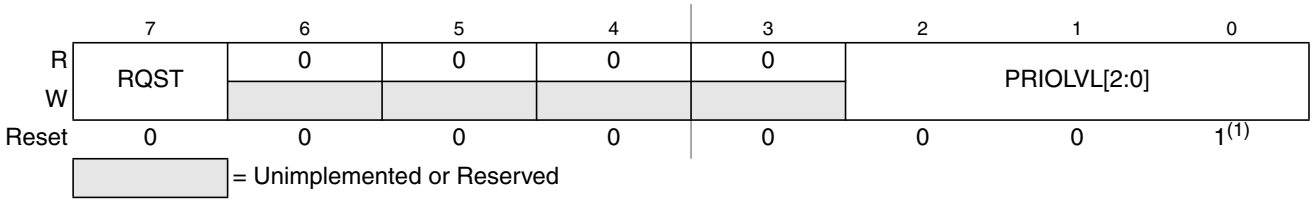


Figure 6-6. Interrupt Request Configuration Data Register 0 (INT_CFDATA0)

1. Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x0129

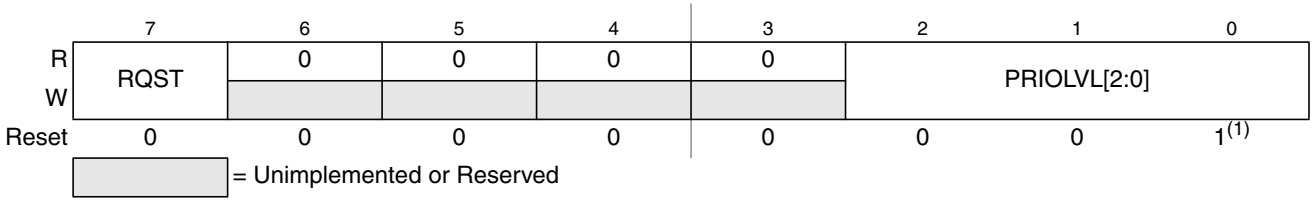


Figure 6-7. Interrupt Request Configuration Data Register 1 (INT_CFDATA1)

1. Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x012A

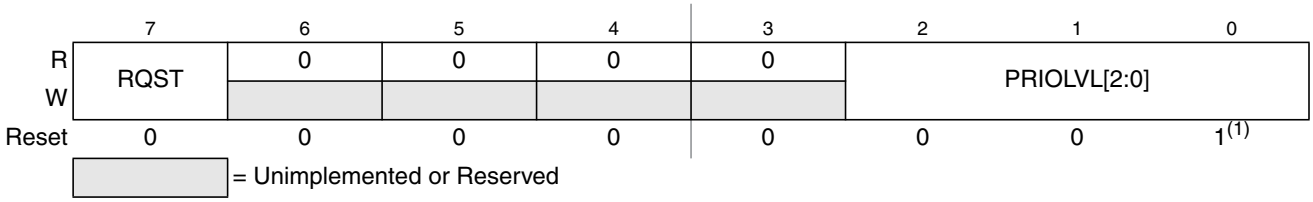


Figure 6-8. Interrupt Request Configuration Data Register 2 (INT_CFDATA2)

1. Please refer to the notes following the PRIOLVL[2:0] description below.

Address: 0x012B

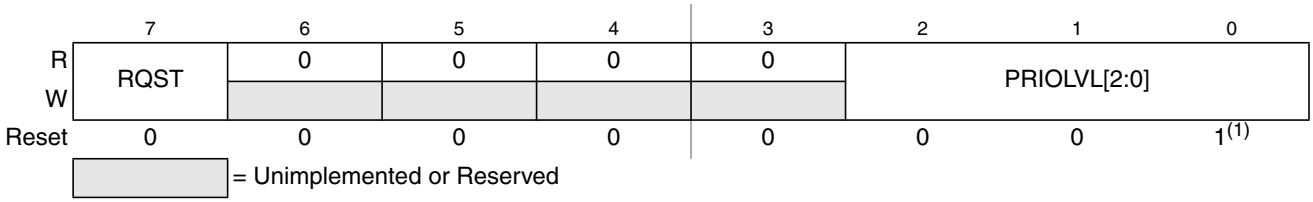


Figure 6-9. Interrupt Request Configuration Data Register 3 (INT_CFDATA3)

1. Please refer to the notes following the PRIOLVL[2:0] description below.

ADC — Add with Carry — ADC

Operation

RS1 + RS2 + C ⇒ RD

Adds the content of register RS1, the content of register RS2 and the value of the Carry bit using binary addition and stores the result in the destination register RD. The Zero Flag is also carried forward from the previous operation allowing 32 and more bit additions.

Example:

```
ADD     R6,R2,R2
ADC     R7,R3,R3 ; R7:R6 = R5:R4 + R3:R2
BCC              ; conditional branch on 32 bit addition
```

CCR Effects

N	Z	V	C
Δ	Δ	Δ	Δ

N: Set if bit 15 of the result is set; cleared otherwise.

Z: Set if the result is $0000 and Z was set before this operation; cleared otherwise.

V: Set if a two´s complement overflow resulted from the operation; cleared otherwise.
RS1[15] & RS2[15] & $\overline{RD[15]}_{new}$ | $\overline{RS1[15]}$ & $\overline{RS2[15]}$ & $RD[15]_{new}$

C: Set if there is a carry from bit 15 of the result; cleared otherwise.
RS1[15] & RS2[15] | RS1[15] & $\overline{RD[15]}_{new}$ | RS2[15] & $\overline{RD[15]}_{new}$

Code and CPU Cycles

Source Form	Address Mode	Machine Code										Cycles
ADC RD, RS1, RS2	TRI	0	0	0	1	1	RD	RS1	RS2	1	1	P

Module Base + 0x00X2

	7	6	5	4	3	2	1	0
R W	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
Reset:	x	x	x	x	x	x	x	x

Figure 16-28. Identifier Register 2 (IDR2) — Extended Identifier Mapping

Table 16-29. IDR2 Register Field Descriptions — Extended

Field	Description
7-0 ID[14:7]	**Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.

Module Base + 0x00X3

	7	6	5	4	3	2	1	0
R W	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR
Reset:	x	x	x	x	x	x	x	x

Figure 16-29. Identifier Register 3 (IDR3) — Extended Identifier Mapping

Table 16-30. IDR3 Register Field Descriptions — Extended

Field	Description
7-1 ID[6:0]	**Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.
0 RTR	**Remote Transmission Request** — This flag reflects the status of the remote transmission request bit in the CAN frame. In the case of a receive buffer, it indicates the status of the received frame and supports the transmission of an answering frame in software. In the case of a transmit buffer, this flag defines the setting of the RTR bit to be sent. 0 Data frame 1 Remote frame

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x000B PWMSCNTB[1]	R	0	0	0	0	0	0	0	0
	W								
0x000C PWMCNT0	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
0x000D PWMCNT1	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
0x000E PWMCNT2	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
0x000F PWMCNT3	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
0x0010 PWMCNT4	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
0x0011 PWMCNT5	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
0x0012 PWMCNT6	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
0x0013 PWMCNT7	R	Bit 7	6	5	4	3	2	1	Bit 0
	W	0	0	0	0	0	0	0	0
0x0014 PWMPER0	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0015 PWMPER1	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0016 PWMPER2	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0017 PWMPER3	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0018 PWMPER4	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								
0x0019 PWMPER5	R	Bit 7	6	5	4	3	2	1	Bit 0
	W								

= Unimplemented or Reserved

Figure 19-2. PWM Register Summary (Sheet 2 of 3)

Module Base + 0x000A, 0x000B

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 19-13. Reserved Registers (PWMSCNTx)

Read: Always read $00 in normal modes

Write: Unimplemented in normal modes

NOTE

Writing to these registers when in special modes can alter the PWM functionality.

19.3.2.12 PWM Channel Counter Registers (PWMCNTx)

Each channel has a dedicated 8-bit up/down counter which runs at the rate of the selected clock source. The counter can be read at any time without affecting the count or the operation of the PWM channel. In left aligned output mode, the counter counts from 0 to the value in the period register - 1. In center aligned output mode, the counter counts from 0 up to the value in the period register and then back down to 0.

Any value written to the counter causes the counter to reset to $00, the counter direction to be set to up, the immediate load of both duty and period registers with values from the buffers, and the output to change according to the polarity bit. The counter is also cleared at the end of the effective period (see Section 19.4.2.5, “Left Aligned Outputs” and Section 19.4.2.6, “Center Aligned Outputs” for more details). When the channel is disabled (PWMEx = 0), the PWMCNTx register does not count. When a channel becomes enabled (PWMEx = 1), the associated PWM counter starts at the count in the PWMCNTx register. For more detailed information on the operation of the counters, see Section 19.4.2.4, “PWM Timer Counters”.

In concatenated mode, writes to the 16-bit counter by using a 16-bit access or writes to either the low or high order byte of the counter will reset the 16-bit counter. Reads of the 16-bit counter must be made by 16-bit access to maintain data coherency.

NOTE

Writing to the counter while the channel is enabled can cause an irregular PWM cycle to occur.

Module Base + 0x000C = PWMCNT0, 0x000D = PWMCNT1, 0x000E = PWMCNT2, 0x000F = PWMCNT3
Module Base + 0x0010 = PWMCNT4, 0x0011 = PWMCNT5, 0x0012 = PWMCNT6, 0x0013 = PWMCNT7

	7	6	5	4	3	2	1	0
R	Bit 7	6	5	4	3	2	1	Bit 0
W	0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0

Figure 19-14. PWM Channel Counter Registers (PWMCNTx)

Read: Anytime

26.4.1.3 Valid Flash Module Commands

Table 26-30. Flash Commands by Mode

FCMD	Command	Unsecured				Secured			
		NS (1)	NX (2)	SS(3)	ST(4)	NS (5)	NX (6)	SS(7)	ST(8)
0x01	Erase Verify All Blocks	*	*	*	*	*	*	*	*
0x02	Erase Verify Block	*	*	*	*	*	*	*	*
0x03	Erase Verify P-Flash Section	*	*	*	*	*			
0x04	Read Once	*	*	*	*	*			
0x05	Load Data Field	*	*	*	*	*			
0x06	Program P-Flash	*	*	*	*	*			
0x07	Program Once	*	*	*	*	*			
0x08	Erase All Blocks			*	*			*	*
0x09	Erase P-Flash Block	*	*	*	*	*			
0x0A	Erase P-Flash Sector	*	*	*	*	*			
0x0B	Unsecure Flash			*	*			*	*
0x0C	Verify Backdoor Access Key	*				*			
0x0D	Set User Margin Level	*	*	*	*	*			
0x0E	Set Field Margin Level			*	*				
0x0F	Full Partition D-Flash			*	*				
0x10	Erase Verify D-Flash Section	*	*	*	*	*			
0x11	Program D-Flash	*	*	*	*	*			
0x12	Erase D-Flash Sector	*	*	*	*	*			
0x13	Enable EEPROM Emulation	*	*	*	*	*	*	*	*
0x14	Disable EEPROM Emulation	*	*	*	*	*	*	*	*
0x15	EEPROM Emulation Query	*	*	*	*	*	*	*	*
0x20	Partition D-Flash	*	*	*	*	*	*	*	*

1. Unsecured Normal Single Chip mode.
2. Unsecured Normal Expanded mode.
3. Unsecured Special Single Chip mode.
4. Unsecured Special Mode.
5. Secured Normal Single Chip mode.
6. Secured Normal Expanded mode.
7. Secured Special Single Chip mode.
8. Secured Special Mode.

Table 27-28. FECCR Index=000 Bit Descriptions

Field	Description
15:8 PAR[7:0]	**ECC Parity Bits** — Contains the 8 parity bits from the 72 bit wide P-Flash data word or the 6 parity bits, allocated to PAR[5:0], from the 22 bit wide D-Flash word with PAR[7:6]=00.
7 XBUS01	**Bus Source Identifier** — The XBUS01 bit determines whether the ECC error was caused by a read access from the CPU or XGATE. 0 ECC Error happened on the CPU access 1 ECC Error happened on the XGATE access
6–0 GADDR[22:16]	**Global Address** — The GADDR[22:16] field contains the upper seven bits of the global address having caused the error.

The P-Flash word addressed by ECCRIX = 001 contains the lower 16 bits of the global address. The following four words addressed by ECCRIX = 010 to 101 contain the 64-bit wide data phrase. The four data words and the parity byte are the uncorrected data read from the P-Flash block.

The D-Flash word addressed by ECCRIX = 001 contains the lower 16 bits of the global address. The uncorrected 16-bit data word is addressed by ECCRIX = 010.

27.3.2.14 Flash Option Register (FOPT)

The FOPT register is the Flash option register.

Offset Module Base + 0x0010

	7	6	5	4	3	2	1	0
R	NV[7:0]							
W								
Reset	F	F	F	F	F	F	F	F

= Unimplemented or Reserved

Figure 27-22. Flash Option Register (FOPT)

All bits in the FOPT register are readable but are not writable.

During the reset sequence, the FOPT register is loaded from the Flash nonvolatile byte in the Flash configuration field at global address 0x7F_FF0E located in P-Flash memory (see Table 27-3) as indicated by reset condition F in Figure 27-22. If a double bit fault is detected while reading the P-Flash phrase containing the Flash nonvolatile byte during the reset sequence, all bits in the FOPT register will be set.

Table 27-29. FOPT Field Descriptions

Field	Description
7–0 NV[7:0]	**Nonvolatile Bits** — The NV[7:0] bits are available as nonvolatile bits. Refer to the device user guide for proper use of the NV bits.

27.3.2.15 Flash Reserved0 Register (FRSV0)

This Flash register is reserved for factory testing.

1. MMCCTL1 register bit

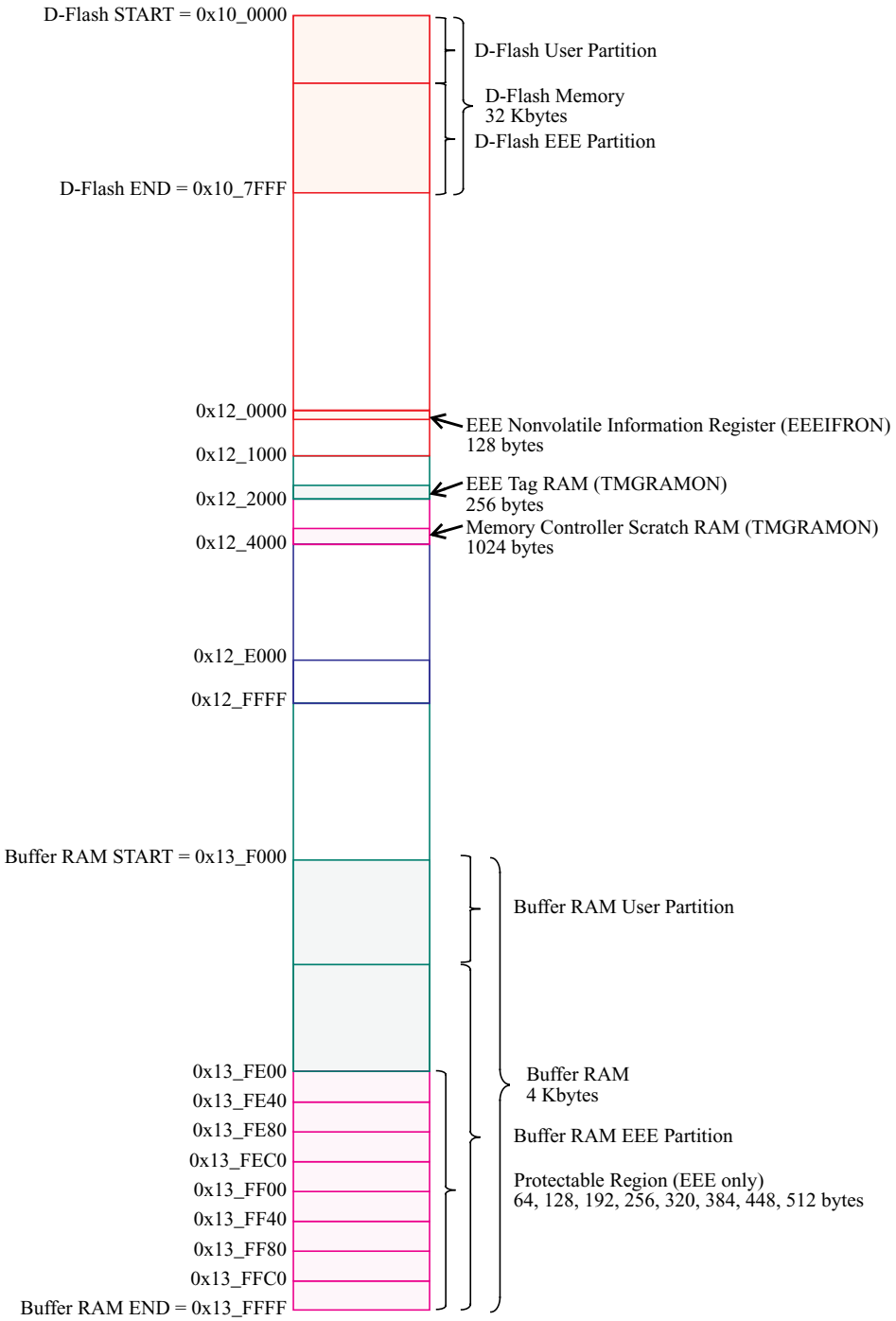


Figure 28-3. EEE Resource Memory Map

Table 29-58. Valid Set User Margin Level Settings

CCOB (CCOBIX=001)	Level Description
0x0002	User Margin-0 Level(2)

1. Read margin to the erased state
2. Read margin to the programmed state

Table 29-59. Set User Margin Level Command Error Handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] != 001 at command launch
		Set if a Load Data Field command sequence is currently active
		Set if command not available in current mode (see Table 29-30)
		Set if an invalid global address [22:16] is supplied
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None
FERSTAT	EPVIOLIF	None

NOTE

User margin levels can be used to check that Flash memory contents have adequate margin for normal level read operations. If unexpected results are encountered when checking Flash memory contents at user margin levels, a potential loss of information has been detected.

29.4.2.14 Set Field Margin Level Command

The Set Field Margin Level command, valid in special modes only, causes the Memory Controller to set the margin level specified for future read operations of a specific P-Flash or D-Flash block.

Table 29-60. Set Field Margin Level Command FCCOB Requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0E	Global address [22:16] to identify the Flash block
001	Margin level setting	

Upon clearing CCIF to launch the Set Field Margin Level command, the Memory Controller will set the field margin level for the targeted block and then set the CCIF flag.

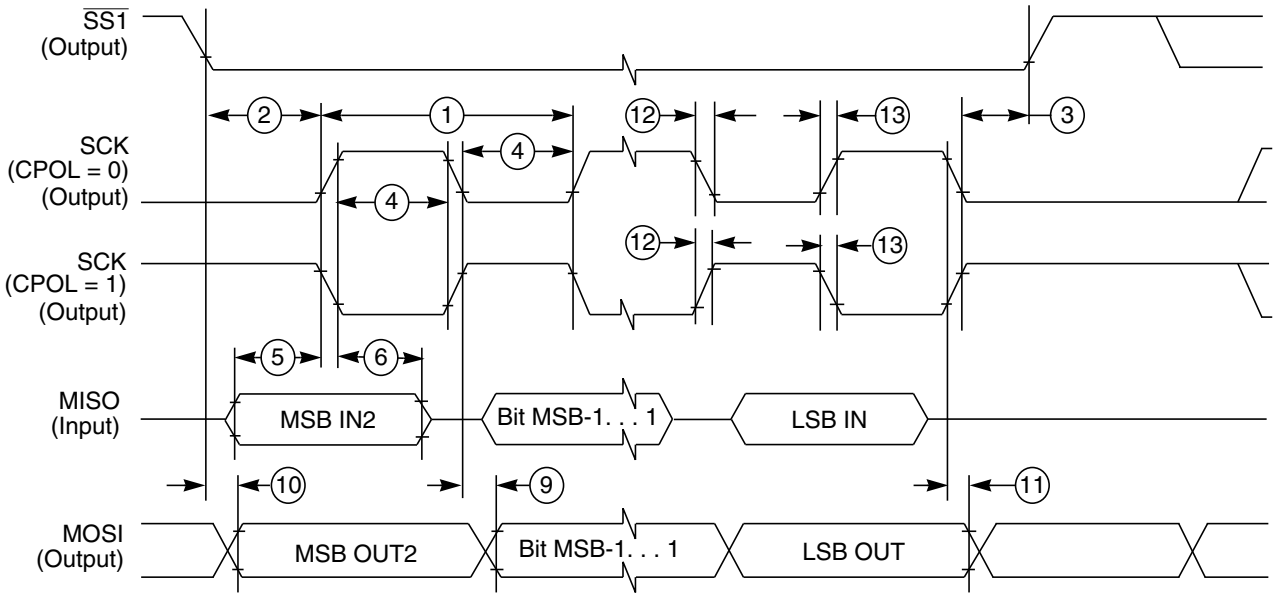
Table A-27. Measurement Conditions

Description	Value	Unit
Drive mode	Full drive mode	—
Load capacitance C_{LOAD}[(1)], on all outputs	50	pF
Thresholds for delay measurement points	(20% / 80%) V_{DDX}	V

1. Timing specified for equal load on all SPI output pins. Avoid asymmetric load.

A.7.2.1 Master Mode

In Figure A-7 the timing diagram for master mode with transmission format CPHA = 0 is depicted.



1. If configured as an output.
2. LSBF = 0. For LSBF = 1, bit order is LSB, bit 1, bit 2... MSB.

Figure A-7. SPI Master Timing (CPHA = 0)

0x01C0–0x01FF MSCAN (CAN2) Map (Sheet 2 of 3)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x01C8	CAN2TARQ	R	0	0	0	0	0	ABTRQ2	ABTRQ1	ABTRQ0
		W								
0x01C9	CAN2TAAK	R	0	0	0	0	0	ABTAK2	ABTAK1	ABTAK0
		W								
0x01CA	CAN2TBSEL	R	0	0	0	0	0	TX2	TX1	TX0
		W								
0x01CB	CAN2IDAC	R	0	0	IDAM1	IDAM0	0	IDHIT2	IDHIT1	IDHIT0
		W								
0x01CC	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x01CD	CAN2MISC	R	0	0	0	0	0	0	0	BOHOLD
		W								
0x01CE	CAN2RXERR	R	RXERR7	RXERR6	RXERR5	RXERR4	RXERR3	RXERR2	RXERR1	RXERR0
		W								
0x01CF	CAN2TXERR	R	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
		W								
0x01D0	CAN2IDAR0	R	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		W								
0x01D1	CAN2IDAR1	R	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		W								
0x01D2	CAN2IDAR2	R	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		W								
0x01D3	CAN2IDAR3	R	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		W								
0x01D4	CAN2IDMR0	R	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		W								
0x01D5	CAN2IDMR1	R	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		W								
0x01D6	CAN2IDMR2	R	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		W								
0x01D7	CAN2IDMR3	R	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		W								
0x01D8	CAN2IDAR4	R	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		W								
0x01D9	CAN2IDAR5	R	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		W								
0x01DA	CAN2IDAR6	R	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		W								
0x01DB	CAN2IDAR7	R	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		W								
0x01DC	CAN2IDMR4	R	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		W								
0x01DD	CAN2IDMR5	R	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		W								
0x01DE	CAN2IDMR6	R	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		W								