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	40MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	44
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	12K x 8
Voltage - Supply (Vcc/Vdd)	1.72V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	64-LQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s12xs256j0mae

Table 1-1. Device Register Memory Map (continued)

Address	Module	Size (Bytes)
0x0368–0x07FF	Reserved	1176

NOTE

Reserved register space shown in Table 1-1 is not allocated to any module. This register space is reserved for future use. Writing to these locations has no effect. Read access to these locations returns zero.

1.1.5 Address Mapping

Figure 1-2 shows S12XS CPU and BDM local address translation to the global memory map. It indicates also the location of the internal resources in the memory map.

Table 2-1. Pin Functions and Priorities (continued)

Port	Pin Name	Pin Function & Priority[1]	I/O	Description	Pin Function after Reset
M	PM[7:6]	GPIO	I/O	General purpose	GPIO
	PM5	(SCK0)	I/O	Serial Peripheral Interface 0 serial clock pin	
		GPIO	I/O	General purpose	
	PM4	(MOSI0)	I/O	Serial Peripheral Interface 0 master out/slave in pin	
		GPIO	I/O	General purpose	
	PM3	($\overline{SS0}$)	I/O	Serial Peripheral Interface 0 slave select output in master mode, input in slave mode or master mode.	
		GPIO	I/O	General purpose	
	PM2	(MISO0)	I/O	Serial Peripheral Interface 0 master in/slave out pin	
		GPIO	I/O	General purpose	
	PM1	TXCAN0	O	MSCAN0 transmit pin	
		(TXD1)	O	Serial Communication Interface 1 transmit pin	
		GPIO	I/O	General purpose	
	PM0	RXCAN0	I	MSCAN0 receive pin	
		(RXD1)	I	Serial Communication Interface 1 receive pin	
		GPIO	I/O	General purpose	

Table 2-39. PTP Register Field Descriptions (continued)

Field	Description
1 PTP	**Port P general purpose input/output data**—Data Register, PWM output, routed TIM output, pin interrupt input/output When not used with the alternative function, the associated pin can be used as general purpose I/O. In general purpose output mode the register bit value is driven to the pin. If the associated data direction bit is set to 1, a read returns the value of the port register bit, otherwise the buffered pin input state is read. • The PWM function takes precedence over the TIM and general purpose I/O function if the related channel is enabled. • The TIM function takes precedence over the general purpose I/O function if the related channel is enabled. • Pin interrupts can be generated if enabled in input or output mode.
0 PTP	**Port P general purpose input/output data**—Data Register, PWM output, routed TIM output, routed SCI1 RXD output, pin interrupt input/output When not used with the alternative function, the associated pin can be used as general purpose I/O. In general purpose output mode the register bit value is driven to the pin. If the associated data direction bit is set to 1, a read returns the value of the port register bit, otherwise the buffered pin input state is read. • The PWM function takes precedence over the TIM, SCI1 and general purpose I/O function if the related channel is enabled. • The TIM function takes precedence over SCI1 and the general purpose I/O function if the related channel is enabled. • The SCI1 function takes precedence over the general purpose I/O function if enabled. • Pin interrupts can be generated if enabled in input or output mode.

2.3.43 Port P Input Register (PTIP)

Address 0x0259 Access: User read[1]

	7	6	5	4	3	2	1	0
R	PTIP7	PTIP6	PTIP5	PTIP4	PTIP3	PTIP2	PTIP1	PTIP0
W								
Reset	u	u	u	u	u	u	u	u

= Unimplemented or Reserved u = Unaffected by reset

Figure 2-41. Port P Input Register (PTIP)

[1] Read: Anytime
Write:Never, writes to this register have no effect

Table 2-40. PTIP Register Field Descriptions

Field	Description
7-0 PTIP	**Port P input data**— A read always returns the buffered input state of the associated pin. It can be used to detect overload or short circuit conditions on output pins.

2.3.57 Port H Interrupt Flag Register (PIFH)

Address 0x0267 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PIFH7	PIFH6	PIFH5	PIFH4	PIFH3	PIFH2	PIFH1	PIFH0
Reset	0	0	0	0	0	0	0	0

Figure 2-55. Port H Interrupt Flag Register (PIFH)

1 Read: Anytime
Write: Anytime

Table 2-54. PIFH Register Field Descriptions

Field	Description
7-0 PIFH	**Port H interrupt flag**— The flag bit is set after an active edge was applied to the associated input pin. This can be a rising or a falling edge based on the state of the polarity select register. Writing a logic “1” to the corresponding bit field clears the flag. 1 Active edge on the associated bit has occurred (an interrupt will occur if the associated enable bit is set) 0 No active edge occurred

2.3.58 Port J Data Register (PTJ)

Address 0x0268 Access: User read/write[1]

	7	6	5	4	3	2	1	0
R	PTJ7	PTJ6	0	0	0	0	PTJ1	PTJ0
W								
Reset	0	0	0	0	0	0	0	0

Figure 2-56. Port J Data Register (PTJ)

1 Read: Anytime, the data source depends on the data direction value
Write: Anytime

Table 2-55. PTJ Register Field Descriptions

Field	Description
7-6, 1-0 PTJ	**Port J general purpose input/output data**—Data Register, pin interrupt input/output The associated pin can be used as general purpose I/O. In general purpose output mode the register bit value is driven to the pin. If the associated data direction bit is set to 1, a read returns the value of the port register bit, otherwise the buffered pin input state is read. • Pin interrupts can be generated if enabled in input or output mode.

4.3.2 Register Descriptions

This section describes in address order all the XINT module registers and their individual bits.

Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0121	IVBR	R	IVB_ADDR[7:0]7							
		W								
0x0126	INT_XGPRIO	R	0	0	0	0	0	XILVL[2:0]		
		W								
0x0127	INT_CFADDR	R	INT_CFADDR[7:4]				0	0	0	0
		W								
0x0128	INT_CFDATA0	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x0129	INT_CFDATA1	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x012A	INT_CFDATA2	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x012B	INT_CFDATA3	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x012C	INT_CFDATA4	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x012D	INT_CFDATA5	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x012E	INT_CFDATA6	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								
0x012F	INT_CFDATA7	R	RQST	0	0	0	0	PRIOLVL[2:0]		
		W								

= Unimplemented or Reserved

Figure 4-2. XINT Register Summary

NOTE

Care must be taken to ensure that all exception requests remain active until the system begins execution of the applicable service routine; otherwise, the exception request may not get processed at all or the result may be a spurious interrupt request (vector at address (vector base + 0x0010)).

4.4.5 Reset Exception Requests

The XINT module supports three system reset exception request types (for details please refer to the Clock and Reset Generator module (CRG)):

1. Pin reset, power-on reset, low-voltage reset, or illegal address reset
2. Clock monitor reset request
3. COP watchdog reset request

4.4.6 Exception Priority

The priority (from highest to lowest) and address of all exception vectors issued by the XINT module upon request by the CPU is shown in Table 4-10. Generally, all non-maskable interrupts have higher priorities than maskable interrupts. Please note that between the three software interrupts (Unimplemented op-code trap request, SWI/BGND request, SYS request) there is no real priority defined because they cannot occur simultaneously (the S12XCPU executes one instruction at a time).

Table 4-10. Exception Vector Map and Priority

Vector Address[1]	Source
0xFFFE	Pin reset, power-on reset, low-voltage reset, illegal address reset
0xFFFC	Clock monitor reset
0xFFFA	COP watchdog reset
(Vector base + 0x00F8)	Unimplemented op-code trap
(Vector base + 0x00F6)	Software interrupt instruction (SWI) or BDM vector request
(Vector base + 0x0012)	System call interrupt instruction (SYS)
(Vector base + 0x0018)	(reserved for future use)
(Vector base + 0x0016)	XGATE Access violation interrupt request[2]
(Vector base + 0x0014)	CPU Access violation interrupt request[3]
(Vector base + 0x00F4)	$\overline{\text{XIRQ}}$ interrupt request
(Vector base + 0x00F2)	$\overline{\text{IRQ}}$ interrupt request
(Vector base + 0x00F0–0x001A)	Device specific I bit maskable interrupt sources (priority determined by the associated configuration registers, in descending order)
(Vector base + 0x0010)	Spurious interrupt

1. 16 bits vector address based
2. only implemented if device features both a Memory Protection Unit (MPU) and an XGATE co-processor
3. only implemented if device features a Memory Protection Unit (MPU)

unimplemented bus, thus preventing proper operation.

The DBGC1_COMRV bits determine which comparator control, address, data and datamask registers are visible in the 8-byte window from 0x0028 to 0x002F as shown in Section Table 6-26.

Table 6-26. Comparator Address Register Visibility

COMRV	Visible Comparator
00	DBGACTL, DBGAAH ,DBGAAM, DBGAAL, DBGADH, DBGADL, DBGADHM, DBGADLM
01	DBGBCTL, DBGBAH, DBGBAM, DBGBAL
10	DBGCCTL, DBGCAH, DBGCAM, DBGCAL, DBGCDH, DBGCDL, DBGCDHM, DBGCDLM
11	DBGDCTL, DBGDAH, DBGDAM, DBGDAL

Table 6-27. DBGXCTL Field Descriptions

Field	Description
7 SZE (Comparators B and D)	**Size Comparator Enable Bit** — The SZE bit controls whether access size comparison is enabled for the associated comparator. This bit is ignored if the TAG bit in the same register is set. 0 Word/Byte access size is not used in comparison 1 Word/Byte access size is used in comparison
6 NDB (Comparators A and C	**Not Data Bus** — The NDB bit controls whether the match occurs when the data bus matches the comparator register value or when the data bus differs from the register value. Furthermore data bus bits can be individually masked using the comparator data mask registers. This bit is only available for comparators A and C. This bit is ignored if the TAG bit in the same register is set. This bit position has an SZ functionality for comparators B and D. 0 Match on data bus equivalence to comparator register contents 1 Match on data bus difference to comparator register contents
6 SZ (Comparators B and D)	**Size Comparator Value Bit** — The SZ bit selects either word or byte access size in comparison for the associated comparator. This bit is ignored if the SZE bit is cleared or if the TAG bit in the same register is set. This bit position has NDB functionality for comparators A and C 0 Word access size will be compared 1 Byte access size will be compared
5 TAG	**Tag Select** — This bit controls whether the comparator match will cause a trigger or tag the opcode at the matched address. Tagged opcodes trigger only if they reach the execution stage of the instruction queue. 0 Trigger immediately on match 1 On match, tag the opcode. If the opcode is about to be executed a trigger is generated
4 BRK	**Break** — This bit controls whether a channel match terminates a debug session immediately, independent of state sequencer state. To generate an immediate breakpoint the module breakpoints must be enabled using DBGBRK. 0 The debug session termination is dependent upon the state sequencer and trigger conditions. 1 A match on this channel terminates the debug session immediately; breakpoints if active are generated, tracing, if active, is terminated and the module disarmed.
3 RW	**Read/Write Comparator Value Bit** — The RW bit controls whether read or write is used in compare for the associated comparator . The RW bit is not used if RWE = 0. 0 Write cycle will be matched 1 Read cycle will be matched
2 RWE	**Read/Write Enable Bit** — The RWE bit controls whether read or write comparison is enabled for the associated comparator. This bit is not used for tagged operations. 0 Read/Write is not used in comparison 1 Read/Write is used in comparison

Read: Anytime

Write: Anytime

Table 8-5. CRGINT Field Descriptions

Field	Description
7 RTIE	**Real Time Interrupt Enable Bit** 0 Interrupt requests from RTI are disabled. 1 Interrupt will be requested whenever RTIF is set.
4 LOCKIE	**Lock Interrupt Enable Bit** 0 LOCK interrupt requests are disabled. 1 Interrupt will be requested whenever LOCKIF is set.
1 SCMIE	**Self Clock Mode Interrupt Enable Bit** 0 SCM interrupt requests are disabled. 1 Interrupt will be requested whenever SCMIF is set.

8.3.2.6 S12XECRG Clock Select Register (CLKSEL)

This register controls S12XECRG clock selection. Refer toFigure 8-16 for more details on the effect of each bit.

Module Base + 0x0005

	7	6	5	4	3	2	1	0
R	PLLSEL	PSTP	XCLKS	0	PLLWAI	0	RTIWAI	COPWAI
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 8-8. S12XECRG Clock Select Register (CLKSEL)

Read: Anytime

Write: Refer to each bit for individual write conditions

Table 8-14. Examples of IPLL Divider Settings[1]

f_{OSC}	REFDIV[5:0]	f_{REF}	REFFRQ[1:0]	SYNDIV[5:0]	f_{VCO}	VCOFRQ[1:0]	POSTDIV[4:0]	f_{PLL}	f_{BUS}
4MHz	$01	2MHz	01	$18	100MHz	11	$00	100MHz	50 MHz
8MHz	$03	2MHz	01	$18	100MHz	11	$00	100MHz	50 MHz
4MHz	$00	4MHz	01	$09	80MHz	01	$00	80MHz	40MHz
8MHz	$00	8MHz	10	$04	80MHz	01	$00	80MHz	40MHz
4MHz	$00	4MHz	01	$03	32MHz	00	$01	16MHz	8MHz
4MHz	$01	2MHz	01	$18	100MHz	11	$01	50MHz	25MHz

1. f_{PLL} and f_{BUS} values in this table may exceed maximum allowed frequencies for some devices. Refer to device information for maximum values.

8.4.1.1.1 IPLL Operation

The oscillator output clock signal (OSCCLK) is fed through the reference programmable divider and is divided in a range of 1 to 64 (REFDIV+1) to output the REFCLK. The VCO output clock, (VCOCLK) is fed back through the programmable loop divider and is divided in a range of 2 to 128 in increments of [2 x (SYNDIV +1)] to output the FBCLK. The VCOCLK is fed to the final programmable divider and is divided in a range of 1,2,4,6,8,... to 62 (2*POSTDIV) to output the PLLCLK. See Figure 8-15.

The phase detector then compares the FBCLK, with the REFCLK. Correction pulses are generated based on the phase difference between the two signals. The loop filter then slightly alters the DC voltage on the internal filter capacitor, based on the width and direction of the correction pulse.

The user must select the range of the REFCLK frequency and the range of the VCOCLK frequency to ensure that the correct IPLL loop bandwidth is set.

The lock detector compares the frequencies of the FBCLK, and the REFCLK. Therefore, the speed of the lock detector is directly proportional to the reference clock frequency. The circuit determines the lock condition based on this comparison.

If IPLL LOCK interrupt requests are enabled, the software can wait for an interrupt request and then check the LOCK bit. If interrupt requests are disabled, software can poll the LOCK bit continuously (during IPLL start-up, usually) or at periodic intervals. In either case, only when the LOCK bit is set, the PLLCLK can be selected as the source for the system and core clocks. If the IPLL is selected as the source for the system and core clocks and the LOCK bit is clear, the IPLL has suffered a severe noise hit and the software must take appropriate action, depending on the application.

- The LOCK bit is a read-only indicator of the locked state of the IPLL.
- The LOCK bit is set when the VCO frequency is within a certain tolerance, Δ_{Lock}, and is cleared when the VCO frequency is out of a certain tolerance, Δ_{unl}.
- Interrupt requests can occur if enabled (LOCKIE = 1) when the lock condition changes, toggling the LOCK bit.

1. Read: Anytime
   Write: Anytime when not in initialization mode

NOTE

The CANTIER register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 11-14. CANTIER Register Field Descriptions

Field	Description
2-0 TXEIE[2:0]	**Transmitter Empty Interrupt Enable** 0 No interrupt request is generated from this event. 1 A transmitter empty (transmit buffer available for transmission) event causes a transmitter empty interrupt request.

11.3.2.9 MSCAN Transmitter Message Abort Request Register (CANTARQ)

The CANTARQ register allows abort request of queued messages as described below.

Module Base + 0x0008 Access: User read/write(1)

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	ABTRQ2	ABTRQ1	ABTRQ0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 11-12. MSCAN Transmitter Message Abort Request Register (CANTARQ)

1. Read: Anytime
   Write: Anytime when not in initialization mode

NOTE

The CANTARQ register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 11-15. CANTARQ Register Field Descriptions

Field	Description
2-0 ABTRQ[2:0]	**Abort Request** — The CPU sets the ABTRQx bit to request that a scheduled message buffer (TXEx = 0) be aborted. The MSCAN grants the request if the message has not already started transmission, or if the transmission is not successful (lost arbitration or error). When a message is aborted, the associated TXE (see Section 11.3.2.7, "MSCAN Transmitter Flag Register (CANTFLG)") and abort acknowledge flags (ABTAK, see Section 11.3.2.10, "MSCAN Transmitter Message Abort Acknowledge Register (CANTAAK)") are set and a transmit interrupt occurs if enabled. The CPU cannot reset ABTRQx. ABTRQx is reset whenever the associated TXE flag is set. 0 No abort request 1 Abort request pending

Chapter 12
Periodic Interrupt Timer (S12PIT24B4CV1)

Table 12-1. Revision History

Version Number	Revision Date	Effective Date	Author	Description of Changes
01.00	28-Apr-05	28-Apr-05		Initial Release
01.01	05-Jul-05	05-Jul-05		Added application section, removed table 1-1

12.1 Introduction

The period interrupt timer (PIT) is an array of 24-bit timers that can be used to trigger peripheral modules or raise periodic interrupts. Refer to Figure 12-1 for a simplified block diagram.

12.1.1 Glossary

Acronyms and Abbreviations	
PIT	Periodic Interrupt Timer
ISR	Interrupt Service Routine
CCR	Condition Code Register
SoC	System on Chip
micro time bases	clock periods of the 16-bit timer modulus down-counters, which are generated by the 8-bit modulus down-counters.

12.1.2 Features

The PIT includes these features:

- Four timers implemented as modulus down-counters with independent time-out periods.
- Time-out periods selectable between 1 and 2^{24} bus clock cycles. Time-out equals m*n bus clock cycles with 1 <= m <= 256 and 1 <= n <= 65536.
- Timers that can be enabled individually.
- Four time-out interrupts.
- Four time-out trigger output signals available to trigger peripheral modules.
- Start of timer channels can be aligned to each other.

12.1.3 Modes of Operation

Refer to the SoC guide for a detailed explanation of the chip modes.

15.2.3 $\overline{SS}$ — Slave Select Pin

This pin is used to output the select signal from the SPI module to another peripheral with which a data transfer is to take place when it is configured as a master and it is used as an input to receive the slave select signal when the SPI is configured as slave.

15.2.4 SCK — Serial Clock Pin

In master mode, this is the synchronous output clock. In slave mode, this is the synchronous input clock.

15.3 Memory Map and Register Definition

This section provides a detailed description of address space and registers used by the SPI.

15.3.1 Module Memory Map

The memory map for the SPI is given in Figure 15-2. The address listed for each register is the sum of a base address and an address offset. The base address is defined at the SoC level and the address offset is defined at the module level. Reads from the reserved bits return zeros and writes to the reserved bits have no effect.

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 SPICR1	R	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
	W								
0x0001 SPICR2	R	0	XFRW	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
	W								
0x0002 SPIBR	R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
	W								
0x0003 SPISR	R	SPIF	0	SPTEF	MODF	0	0	0	0
	W								
0x0004 SPIDRH	R	R15	R14	R13	R12	R11	R10	R9	R8
	W	T15	T14	T13	T12	T11	T10	T9	T8
0x0005 SPIDRL	R	R7	R6	R5	R4	R3	R2	R1	R0
	W	T7	T6	T5	T4	T3	T2	T1	T0
0x0006 Reserved	R								
	W								
0x0007 Reserved	R								
	W								

= Unimplemented or Reserved

Figure 15-2. SPI Register Summary

The CPOL clock polarity control bit specifies an active high or low clock and has no significant effect on the transmission format.

The CPHA clock phase control bit selects one of two fundamentally different transmission formats.

Clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

15.4.3.2 CPHA = 0 Transfer Format

The first edge on the SCK line is used to clock the first data bit of the slave into the master and the first data bit of the master into the slave. In some peripherals, the first bit of the slave's data is available at the slave's data out pin as soon as the slave is selected. In this format, the first SCK edge is issued a half cycle after $\overline{SS}$ has become low.

A half SCK cycle later, the second edge appears on the SCK line. When this second edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the shift register, depending on LSBFE bit.

After this second edge, the next bit of the SPI master data is transmitted out of the serial data output pin of the master to the serial input pin on the slave. This process continues for a total of 16 edges on the SCK line, with data being latched on odd numbered edges and shifted on even numbered edges.

Data reception is double buffered. Data is shifted serially into the SPI shift register during the transfer and is transferred to the parallel SPI data register after the last bit is shifted in.

After $2n$[1] (last) SCK edges:

- Data that was previously in the master SPI data register should now be in the slave data register and the data that was in the slave data register should be in the master.
- The SPIF flag in the SPI status register is set, indicating that the transfer is complete.

Figure 15-12 is a timing diagram of an SPI transfer where CPHA = 0. SCK waveforms are shown for CPOL = 0 and CPOL = 1. The diagram may be interpreted as a master or slave timing diagram because the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave and the MOSI signal is the output from the master. The $\overline{SS}$ pin of the master must be either high or reconfigured as a general-purpose output not affecting the SPI.

1. n depends on the selected transfer width, please refer to Section 15.3.2.2, "SPI Control Register 2 (SPICR2)

16.3.2.11 Timer System Control Register 2 (TSCR2)

Module Base + 0x000D

	7	6	5	4	3	2	1	0
R	TOI	0	0	0	TCRE	PR2	PR1	PR0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 16-19. Timer System Control Register 2 (TSCR2)

Read: Anytime

Write: Anytime.

Table 16-14. TSCR2 Field Descriptions

Field	Description
7 TOI	**Timer Overflow Interrupt Enable** 0 Interrupt inhibited. 1 Hardware interrupt requested when TOF flag set.
3 TCRE	**Timer Counter Reset Enable** — This bit allows the timer counter to be reset by a successful output compare 7 event. This mode of operation is similar to an up-counting modulus counter. 0 Counter reset inhibited and counter free runs. 1 Counter reset by a successful output compare 7. **Note:** If TC7 = 0x0000 and TCRE = 1, TCNT will stay at 0x0000 continuously. If TC7 = 0xFFFF and TCRE = 1, TOF will never be set when TCNT is reset from 0xFFFF to 0x0000. **Note:** TCRE=1 and TC7!=0, the TCNT cycle period will be TC7 x "prescaler counter width" + "1 Bus Clock", for a more detail explanation please refer to Section 16.4.3, “Output Compare
2 PR[2:0]	**Timer Prescaler Select** — These three bits select the frequency of the timer prescaler clock derived from the Bus Clock as shown in Table 16-15.

Table 16-15. Timer Clock Selection

PR2	PR1	PR0	Timer Clock
0	0	0	Bus Clock / 1
0	0	1	Bus Clock / 2
0	1	0	Bus Clock / 4
0	1	1	Bus Clock / 8
1	0	0	Bus Clock / 16
1	0	1	Bus Clock / 32
1	1	0	Bus Clock / 64
1	1	1	Bus Clock / 128

NOTE

The pulse accumulator counter can operate in event counter mode even when the timer enable bit, TEN, is clear.

16.4.6 Gated Time Accumulation Mode

Setting the PAMOD bit configures the pulse accumulator for gated time accumulation operation. An active level on the PACNT input pin enables a divided-by-64 clock to drive the pulse accumulator. The PEDGE bit selects low levels or high levels to enable the divided-by-64 clock.

The trailing edge of the active level at the IOC7 pin sets the PAIF. The PAI bit enables the PAIF flag to generate interrupt requests.

The pulse accumulator counter register reflect the number of pulses from the divided-by-64 clock since the last reset.

NOTE

The timer prescaler generates the divided-by-64 clock. If the timer is not active, there is no divided-by-64 clock.

16.5 Resets

The reset state of each individual bit is listed within Section 16.3, “Memory Map and Register Definition” which details the registers and their bit fields.

16.6 Interrupts

This section describes interrupts originated by the TIM16B8CV2 block. Table 16-25 lists the interrupts generated by the TIM16B8CV2 to communicate with the MCU.

Table 16-25. TIM16B8CV1 Interrupts

Interrupt	Offset[1]	Vector[1]	Priority[1]	Source	Description
C[7:0]F	—	—	—	Timer Channel 7–0	Active high timer channel interrupts 7–0
PAOVI	—	—	—	Pulse Accumulator Input	Active high pulse accumulator input interrupt
PAOVF	—	—	—	Pulse Accumulator Overflow	Pulse accumulator overflow interrupt
TOF	—	—	—	Timer Overflow	Timer Overflow interrupt

1 Chip Dependent.

The TIM16B8CV2 uses a total of 11 interrupt vectors. The interrupt vector offsets and interrupt numbers are chip dependent.

[1] Preferred SEC state to set MCU to secured state.

The security function in the Flash module is described in Section 18.5.

18.3.2.3 Flash CCOB Index Register (FCCOBIX)

The FCCOBIX register is used to index the FCCOB register for Flash memory operations.

Offset Module Base + 0x0002

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

= Unimplemented or Reserved

Figure 18-7. FCCOB Index Register (FCCOBIX)

CCOBIX bits are readable and writable while remaining bits read 0 and are not writable.

Table 18-11. FCCOBIX Field Descriptions

Field	Description
2–0 CCOBIX[1:0]	**Common Command Register Index**— The CCOBIX bits are used to select which word of the FCCOB register array is being read or written to. See Section 18.3.2.11, “Flash Common Command Object Register (FCCOB),” for more details.

18.3.2.4 Flash ECCR Index Register (FECCRIX)

The FECCRIX register is used to index the FECCR register for ECC fault reporting.

Offset Module Base + 0x0003

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

= Unimplemented or Reserved

Figure 18-8. FECCR Index Register (FECCRIX)

ECCRIX bits are readable and writable while remaining bits read 0 and are not writable.

Table 18-12. FECCRIX Field Descriptions

Field	Description
2-0 ECCRIX[2:0]	**ECC Error Register Index**— The ECCRIX bits are used to select which word of the FECCR register array is being read. See Section 18.3.2.14, “Flash ECC Error Results Register (FECCR),” for more details.

19.4.1.2 Command Write Sequence

The Memory Controller will launch all valid Flash commands entered using a command write sequence.

Before launching a command, the ACCERR and FPVIOL bits in the FSTAT register must be clear (see Section 19.3.2.7) and the CCIF flag should be tested to determine the status of the current command write sequence. If CCIF is 0, the previous command write sequence is still active, a new command write sequence cannot be started, and all writes to the FCCOB register are ignored.

CAUTION

Writes to any Flash register must be avoided while a Flash command is active (CCIF=0) to prevent corruption of Flash register contents and Memory Controller behavior.

19.4.1.2.1 Define FCCOB Contents

The FCCOB parameter fields must be loaded with all required parameters for the Flash command being executed. Access to the FCCOB parameter fields is controlled via the CCOBIX bits in the FCCOBIX register (see Section 19.3.2.3).

The contents of the FCCOB parameter fields are transferred to the Memory Controller when the user clears the CCIF command completion flag in the FSTAT register (writing 1 clears the CCIF to 0). The CCIF flag will remain clear until the Flash command has completed. Upon completion, the Memory Controller will return CCIF to 1 and the FCCOB register will be used to communicate any results. The flow for a generic command write sequence is shown in Figure 19-26.

- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and phrase program operation
- Flexible protection scheme to prevent accidental program or erase of P-Flash memory

20.1.2.2 D-Flash Features

- 4 Kbytes of D-Flash memory composed of one 4 Kbyte Flash block divided into 16 sectors of 256 bytes
- Single bit fault correction and double bit fault detection within a word during read operations
- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and word program operation
- Protection scheme to prevent accidental program or erase of D-Flash memory
- Ability to program up to four words in a burst sequence

20.1.2.3 Other Flash Module Features

- No external high-voltage power supply required for Flash memory program and erase operations
- Interrupt generation on Flash command completion and Flash error detection
- Security mechanism to prevent unauthorized access to the Flash memory

20.3.2.17 Flash Reserved3 Register (FRSV3)

This Flash register is reserved for factory testing.

Offset Module Base + 0x0012

	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 20-24. Flash Reserved3 Register (FRSV3)

All bits in the FRSV3 register read 0 and are not writable.

20.3.2.18 Flash Reserved4 Register (FRSV4)

This Flash register is reserved for factory testing.

Offset Module Base + 0x0013

	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 20-25. Flash Reserved4 Register (FRSV4)

All bits in the FRSV4 register read 0 and are not writable.

20.4 Functional Description

20.4.1 Flash Command Operations

Flash command operations are used to modify Flash memory contents.

The next sections describe:

- How to write the FCLKDIV register that is used to generate a time base (FCLK) derived from OSCCLK for Flash program and erase command operations
- The command write sequence used to set Flash command parameters and launch execution
- Valid Flash commands available for execution