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	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	CANbus, EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	60
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc9s12c32mfue25

1.5.3.3 Wait

This mode is entered by executing the CPU WAI instruction. In this mode the CPU will not execute instructions. The internal CPU signals (address and data bus) will be fully static. All peripherals stay active. For further power consumption reduction the peripherals can individually turn off their local clocks.

1.5.3.4 Run

Although this is not a low-power mode, unused peripheral modules should not be enabled in order to save power.

1.6 Resets and Interrupts

Consult the Exception Processing section of the CPU12 Reference Manual for information.

1.6.1 Vectors

Table 1-9 lists interrupt sources and vectors in default order of priority.

Table 1-9. Interrupt Vector Locations

Vector Address	Interrupt Source	CCR Mask	Local Enable	HPRIO Value to Elevate
0xFFFE, 0xFFFF	External reset, power on reset, or low voltage reset (see CRG flags register to determine reset source)	None	None	—
0xFFFC, 0xFFFD	Clock monitor fail reset	None	COPCTL (CME, FCME)	—
0xFFFA, 0xFFFB	COP failure reset	None	COP rate select	—
0xFFF8, 0xFFF9	Unimplemented instruction trap	None	None	—
0xFFF6, 0xFFF7	SWI	None	None	—
0xFFF4, 0xFFF5	XIRQ	X-Bit	None	—
0xFFF2, 0xFFF3	IRQ	I bit	INTCR (IRQEN)	0x00F2
0xFFF0, 0xFFF1	Real time Interrupt	I bit	CRGINT (RTIE)	0x00F0
0xFFEE, 0xFFEF	Standard timer channel 0	I bit	TIE (C0I)	0x00EE
0xFFEC, 0xFFED	Standard timer channel 1	I bit	TIE (C1I)	0x00EC
$FFEE, $FFEF	Reserved			
$FFEC, $FFED	Reserved			
0xFFEA, 0xFFEB	Standard timer channel 2	I bit	TIE (C2I)	0x00EA
0xFFE8, 0xFFE9	Standard timer channel 3	I bit	TIE (C3I)	0x00E8
0xFFE6, 0xFFE7	Standard timer channel 4	I bit	TIE (C4I)	0x00E6
0xFFE4, 0xFFE5	Standard timer channel 5	I bit	TIE (C5I)	0x00E4
0xFFE2, 0xFFE3	Standard timer channel 6	I bit	TIE (C6I)	0x00E2
0xFFE0, 0xFFE1	Standard timer channel 7	I bit	TIE (C7I)	0x00E0

2.3.2.3 Port M Registers

2.3.2.3.1 Port M I/O Register (PTM)

Module Base + 0x0010

	7	6	5	4	3	2	1	0
R	0	0	PTM5	PTM4	PTM3	PTM2	PTM1	PTM0
W								
MSCAN/ SPI	—	—	SCK	MOSI	$\overline{SS}$	MISO	TXCAN	RXCAN
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 2-17. Port M I/O Register (PTM)

Read: Anytime.

Write: Anytime.

If the data direction bits of the associated I/O pins are set to 1, a read returns the value of the port register, otherwise the value at the pins is read.

The SPI pin configurations (PM[5:2]) is determined by several status bits in the SPI module. *Please refer to the SPI Block User Guide for details*.

2.3.2.3.2 Port M Input Register (PTIM)

Module Base + 0x0011

	7	6	5	4	3	2	1	0
R	0	0	PTIM5	PTIM4	PTIM3	PTIM2	PTIM1	PTIM0
W								
Reset	—	—	—	—	—	—	—	—

= Unimplemented or Reserved

Figure 2-18. Port M Input Register (PTIM)

Read: Anytime.

Write: Never, writes to this register have no effect.

Table 2-16. PTIM Field Descriptions

Field	Description
5–0 PTIM[5:0]	**Port M Input Register** — This register always reads back the status of the associated pins. This also can be used to detect overload or short circuit conditions on output pins.

2.3.2.5.5 Port J Pull Device Enable Register (PERJ)

Module Base + 0x002C

	7	6	5	4	3	2	1	0
R	PERJ7	PERJ6	0	0	0	0	0	0
W								
Reset	0	0	—	—	—	—	—	—

= Unimplemented or Reserved

Figure 2-36. Port J Pull Device Enable Register (PERJ)

Read: Anytime.

Write: Anytime.

Table 2-30. PERJ Field Descriptions

Field	Description
7–6 PERJ[7:6]	**Reduced Drive Port J** — This register configures whether a pull-up or a pull-down device is activated, if the port is used as input or as wired-or output. This bit has no effect if the port is used as push-pull output. 0 Pull-up or pull-down device is disabled. 1 Either a pull-up or pull-down device is enabled.

2.3.2.5.6 Port J Polarity Select Register (PPSJ)

Module Base + 0x002D

	7	6	5	4	3	2	1	0
R	PPSJ7	PPSJ6	0	0	0	0	0	0
W								
Reset	0	0	—	—	—	—	—	—

= Unimplemented or Reserved

Figure 2-37. Port J Polarity Select Register (PPSJ)

Read: Anytime.

Write: Anytime.

Table 2-31. PPSJ Field Descriptions

Field	Description
7–6 PPSJ[7:6]	**Reduced Drive Port J** — This register serves a dual purpose by selecting the polarity of the active interrupt edge as well as selecting a pull-up or pull-down device if enabled. 0 Falling edge on the associated port J pin sets the associated flag bit in the PIFJ register. A pull-up device is connected to the associated port J pin, if enabled by the associated bit in register PERJ and if the port is used as general purpose input. 1 Rising edge on the associated port J pin sets the associated flag bit in the PIFJ register. A pull-down device is connected to the associated port J pin, if enabled by the associated bit in register PERJ and if the port is used as input.

4.3.2.5 Reserved Registers

Module Base + 0x0004
Starting address location affected by INITRG register setting.

	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 4-6. Reserved Register

Module Base + 0x0005
Starting address location affected by INITRG register setting.

	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 4-7. Reserved Register

Module Base + 0x0006
Starting address location affected by INITRG register setting.

	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 4-8. Reserved Register

Module Base + 0x0007
Starting address location affected by INITRG register setting.

	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 4-9. Reserved Register

5.4.1 Low-Power Modes

The INT does not contain any user-controlled options for reducing power consumption. The operation of the INT in low-power modes is discussed in the following subsections.

5.4.1.1 Operation in Run Mode

The INT does not contain any options for reducing power in run mode.

5.4.1.2 Operation in Wait Mode

Clocks to the INT can be shut off during system wait mode and the asynchronous interrupt path will be used to generate the wake-up signal upon recognition of a valid interrupt or any $\overline{\text{XIRQ}}$ request.

5.4.1.3 Operation in Stop Mode

Clocks to the INT can be shut off during system stop mode and the asynchronous interrupt path will be used to generate the wake-up signal upon recognition of a valid interrupt or any $\overline{\text{XIRQ}}$ request.

5.5 Resets

The INT supports three system reset exception request types: normal system reset or power-on-reset request, crystal monitor reset request, and COP watchdog reset request. The type of reset exception request must be decoded by the system and the proper request made to the core. The INT will then provide the service routine address for the type of reset requested.

5.6 Interrupts

As shown in the block diagram in Figure 5-1, the INT contains a register block to provide interrupt status and control, an optional highest priority I interrupt (HPRIO) block, and a priority decoder to evaluate whether pending interrupts are valid and assess their priority.

5.6.1 Interrupt Registers

The INT registers are accessible only in special modes of operation and function as described in Section 5.3.2.1, “Interrupt Test Control Register,” and Section 5.3.2.2, “Interrupt Test Registers,” previously.

5.6.2 Highest Priority I-Bit Maskable Interrupt

When the optional HPRIO block is implemented, the user is allowed to promote a single I-bit maskable interrupt to be the highest priority I interrupt. The HPRIO evaluates all interrupt exception requests and passes the HPRIO vector to the priority decoder if the highest priority I interrupt is active. RTI replaces the promoted interrupt source.

6.4.7 Serial Interface Hardware Handshake Protocol

BDM commands that require CPU execution are ultimately treated at the MCU bus rate. Because the BDM clock source can be asynchronously related to the bus frequency, when CLKSW = 0, it is very helpful to provide a handshake protocol in which the host could determine when an issued command is executed by the CPU. The alternative is to always wait the amount of time equal to the appropriate number of cycles at the slowest possible rate the clock could be running. This sub-section will describe the hardware handshake protocol.

The hardware handshake protocol signals to the host controller when an issued command was successfully executed by the target. This protocol is implemented by a 16 serial clock cycle low pulse followed by a brief speedup pulse in the BKGD pin. This pulse is generated by the target MCU when a command, issued by the host, has been successfully executed (see Figure 6-10). This pulse is referred to as the ACK pulse. After the ACK pulse has finished: the host can start the bit retrieval if the last issued command was a read command, or start a new command if the last command was a write command or a control command (BACKGROUND, GO, GO_UNTIL, or TRACE1). The ACK pulse is not issued earlier than 32 serial clock cycles after the BDM command was issued. The end of the BDM command is assumed to be the 16th tick of the last bit. This minimum delay assures enough time for the host to perceive the ACK pulse. Note also that, there is no upper limit for the delay between the command and the related ACK pulse, because the command execution depends upon the CPU bus frequency, which in some cases could be very slow compared to the serial communication rate. This protocol allows a great flexibility for the POD designers, because it does not rely on any accurate time measurement or short response time to any event in the serial communication.

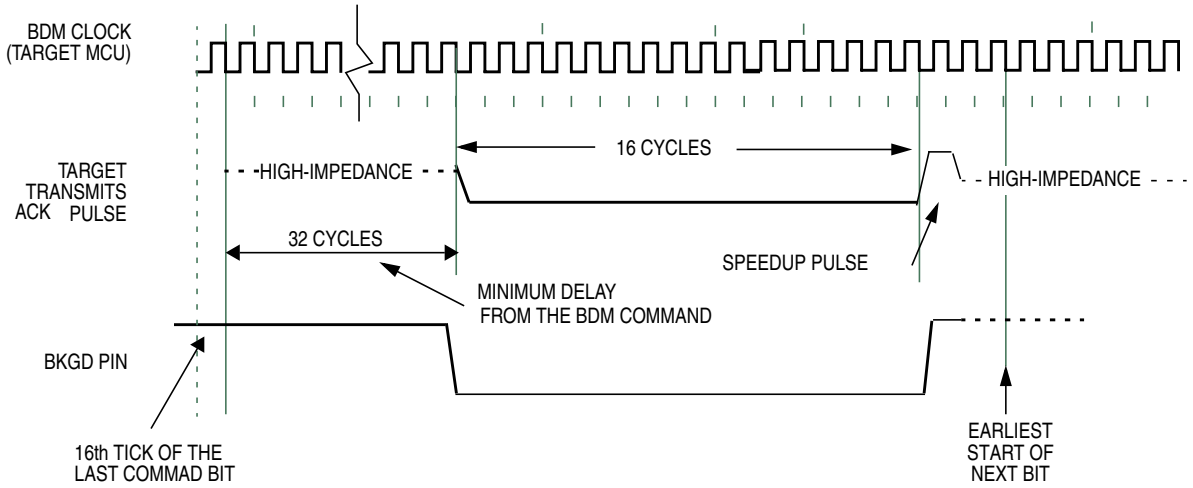

Figure 6-10. Target Acknowledge Pulse (ACK)

NOTE

If the ACK pulse was issued by the target, the host assumes the previous command was executed. If the CPU enters WAIT or STOP prior to executing a hardware command, the ACK pulse will not be issued meaning that the BDM command was not executed. After entering wait or stop mode, the BDM command is no longer pending.

8.3.2.13.1 Left Justified Result Data

Module Base + 0x0010 = ATDDR0H, 0x0012 = ATDDR1H, 0x0014 = ATDDR2H, 0x0016 = ATDDR3H
0x0018 = ATDDR4H, 0x001A = ATDDR5H, 0x001C = ATDDR6H, 0x001E = ATDDR7H

	7	6	5	4	3	2	1	0	
R W	BIT 9 MSB BIT 7 MSB	BIT 8 BIT 6	BIT 7 BIT 5	BIT 6 BIT 4	BIT 5 BIT 3	BIT 4 BIT 2	BIT 3 BIT 1	BIT 2 BIT 0	10-bit data 8-bit data
Reset	0	0	0	0	0	0	0	0	

Figure 8-15. Left Justified, ATD Conversion Result Register, High Byte (ATDDRxH)

Module Base + 0x0011 = ATDDR0L, 0x0013 = ATDDR1L, 0x0015 = ATDDR2L, 0x0017 = ATDDR3L
0x0019 = ATDDR4L, 0x001B = ATDDR5L, 0x001D = ATDDR6L, 0x001F = ATDDR7L

	7	6	5	4	3	2	1	0	
R W	BIT 1 U	BIT 0 U	0 0	0 0	0 0	0 0	0 0	0 0	10-bit data 8-bit data
Reset	0	0	0	0	0	0	0	0	

Figure 8-16. Left Justified, ATD Conversion Result Register, Low Byte (ATDDRxL)

8.3.2.13.2 Right Justified Result Data

Module Base + 0x0010 = ATDDR0H, 0x0012 = ATDDR1H, 0x0014 = ATDDR2H, 0x0016 = ATDDR3H
0x0018 = ATDDR4H, 0x001A = ATDDR5H, 0x001C = ATDDR6H, 0x001E = ATDDR7H

	7	6	5	4	3	2	1	0	
R W	0 0	0 0	0 0	0 0	0 0	0 0	BIT 9 MSB 0	BIT 8 0	10-bit data 8-bit data
Reset	0	0	0	0	0	0	0	0	

Figure 8-17. Right Justified, ATD Conversion Result Register, High Byte (ATDDRxH)

Module Base + 0x0011 = ATDDR0L, 0x0013 = ATDDR1L, 0x0015 = ATDDR2L, 0x0017 = ATDDR3L
0x0019 = ATDDR4L, 0x001B = ATDDR5L, 0x001D = ATDDR6L, 0x001F = ATDDR7L

	7	6	5	4	3	2	1	0	
R W	BIT 7 BIT 7 MSB	BIT 6 BIT 6	BIT 5 BIT 5	BIT 4 BIT 4	BIT 3 BIT 3	BIT 2 BIT 2	BIT 1 BIT 1	BIT 0 BIT 0	10-bit data 8-bit data
Reset	0	0	0	0	0	0	0	0	

Figure 8-18. Right Justified, ATD Conversion Result Register, Low Byte (ATDDRxL)

Table 10-1. CANCTL0 Register Field Descriptions (continued)

Field	Description
1 SLPRQ[5]	**Sleep Mode Request** — This bit requests the MSCAN to enter sleep mode, which is an internal power saving mode (see Section 10.4.5.4, "MSCAN Sleep Mode"). The sleep mode request is serviced when the CAN bus is idle, i.e., the module is not receiving a message and all transmit buffers are empty. The module indicates entry to sleep mode by setting SLPAK = 1 (see Section 10.3.2.2, "MSCAN Control Register 1 (CANCTL1)"). SLPRQ cannot be set while the WUPIF flag is set (see Section 10.3.2.5, "MSCAN Receiver Flag Register (CANRFLG)"). Sleep mode will be active until SLPRQ is cleared by the CPU or, depending on the setting of WUPE, the MSCAN detects activity on the CAN bus and clears SLPRQ itself. 0 Running — The MSCAN functions normally 1 Sleep mode request — The MSCAN enters sleep mode when CAN bus idle
0 INITRQ[6],[7]	**Initialization Mode Request** — When this bit is set by the CPU, the MSCAN skips to initialization mode (see Section 10.4.5.5, "MSCAN Initialization Mode"). Any ongoing transmission or reception is aborted and synchronization to the CAN bus is lost. The module indicates entry to initialization mode by setting INITAK = 1 (Section 10.3.2.2, "MSCAN Control Register 1 (CANCTL1)"). The following registers enter their hard reset state and restore their default values: CANCTL0[8], CANRFLG[9], CANRIER[10], CANTFLG, CANTIER, CANTARQ, CANTAAK, and CANTBSEL. The registers CANCTL1, CANBTR0, CANBTR1, CANIDAC, CANIDAR0-7, and CANIDMR0-7 can only be written by the CPU when the MSCAN is in initialization mode (INITRQ = 1 and INITAK = 1). The values of the error counters are not affected by initialization mode. When this bit is cleared by the CPU, the MSCAN restarts and then tries to synchronize to the CAN bus. If the MSCAN is not in bus-off state, it synchronizes after 11 consecutive recessive bits on the CAN bus; if the MSCAN is in bus-off state, it continues to wait for 128 occurrences of 11 consecutive recessive bits. Writing to other bits in CANCTL0, CANRFLG, CANRIER, CANTFLG, or CANTIER must be done only after initialization mode is exited, which is INITRQ = 0 and INITAK = 0. 0 Normal operation 1 MSCAN in initialization mode

1. The MSCAN must be in normal mode for this bit to become set.
2. See the Bosch CAN 2.0A/B specification for a detailed definition of transmitter and receiver states.
3. In order to protect from accidentally violating the CAN protocol, the TXCAN pin is immediately forced to a recessive state when the CPU enters wait (CSWAI = 1) or stop mode (see Section 10.4.5.2, "Operation in Wait Mode" and Section 10.4.5.3, "Operation in Stop Mode").
4. The CPU has to make sure that the WUPE register and the WUPIE wake-up interrupt enable register (see Section 10.3.2.6, "MSCAN Receiver Interrupt Enable Register (CANRIER)) is enabled, if the recovery mechanism from stop or wait is required.
5. The CPU cannot clear SLPRQ before the MSCAN has entered sleep mode (SLPRQ = 1 and SLPAK = 1).
6. The CPU cannot clear INITRQ before the MSCAN has entered initialization mode (INITRQ = 1 and INITAK = 1).
7. In order to protect from accidentally violating the CAN protocol, the TXCAN pin is immediately forced to a recessive state when the initialization mode is requested by the CPU. Thus, the recommended procedure is to bring the MSCAN into sleep mode (SLPRQ = 1 and SLPAK = 1) before requesting initialization mode.
8. Not including WUPE, INITRQ, and SLPRQ.
9. TSTAT1 and TSTAT0 are not affected by initialization mode.
10. RSTAT1 and RSTAT0 are not affected by initialization mode.

10.3.2.2 MSCAN Control Register 1 (CANCTL1)

The CANCTL1 register provides various control bits and handshake status information of the MSCAN module as described below.

10.3.2.12 MSCAN Identifier Acceptance Control Register (CANIDAC)

The CANIDAC register is used for identifier acceptance control as described below.

Module Base + 0x000B

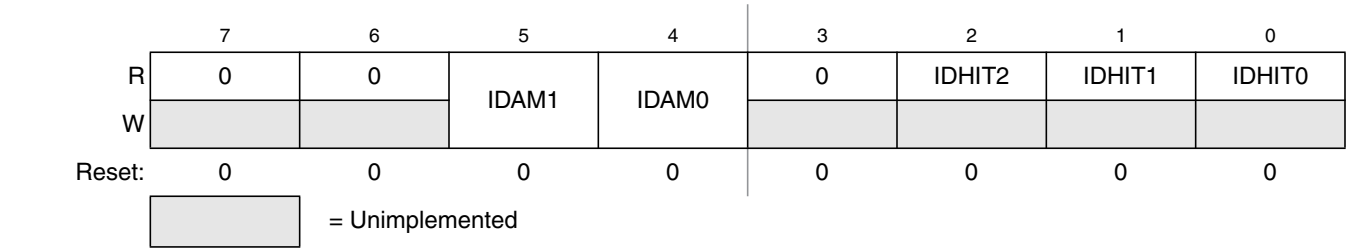

Figure 10-15. MSCAN Identifier Acceptance Control Register (CANIDAC)

Read: Anytime
Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1), except bits IDHITx, which are read-only

Table 10-16. CANIDAC Register Field Descriptions

Field	Description
5:4 IDAM[1:0]	**Identifier Acceptance Mode** — The CPU sets these flags to define the identifier acceptance filter organization (see Section 10.4.3, "Identifier Acceptance Filter"). Table 10-17 summarizes the different settings. In filter closed mode, no message is accepted such that the foreground buffer is never reloaded.
2:0 IDHIT[2:0]	Identifier Acceptance Hit Indicator — The MSCAN sets these flags to indicate an identifier acceptance hit (see Section 10.4.3, "Identifier Acceptance Filter"). Table 10-18 summarizes the different settings.

Table 10-17. Identifier Acceptance Mode Settings

IDAM1	IDAM0	Identifier Acceptance Mode
0	0	Two 32-bit acceptance filters
0	1	Four 16-bit acceptance filters
1	0	Eight 8-bit acceptance filters
1	1	Filter closed

Table 10-18. Identifier Acceptance Hit Indication

IDHIT2	IDHIT1	IDHIT0	Identifier Acceptance Hit
0	0	0	Filter 0 hit
0	0	1	Filter 1 hit
0	1	0	Filter 2 hit
0	1	1	Filter 3 hit
1	0	0	Filter 4 hit
1	0	1	Filter 5 hit
1	1	0	Filter 6 hit
1	1	1	Filter 7 hit

14.3.2.2 SPI Control Register 2 (SPICR2)

Module Base 0x0001

	7	6	5	4	3	2	1	0
R	0	0	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 14-4. SPI Control Register 2 (SPICR2)

Read: anytime

Write: anytime; writes to the reserved bits have no effect

Table 14-4. SPICR2 Field Descriptions

Field	Description
4 MODFEN	**Mode Fault Enable Bit** — This bit allows the MODF failure being detected. If the SPI is in master mode and MODFEN is cleared, then the $\overline{SS}$ port pin is not used by the SPI. In slave mode, the $\overline{SS}$ is available only as an input regardless of the value of MODFEN. For an overview on the impact of the MODFEN bit on the $\overline{SS}$ port pin configuration refer to Table 14-3. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 $\overline{SS}$ port pin is not used by the SPI 1 $\overline{SS}$ port pin with MODF feature
3 BIDIROE	**Output Enable in the Bidirectional Mode of Operation** — This bit controls the MOSI and MISO output buffer of the SPI, when in bidirectional mode of operation (SPC0 is set). In master mode this bit controls the output buffer of the MOSI port, in slave mode it controls the output buffer of the MISO port. In master mode, with SPC0 set, a change of this bit will abort a transmission in progress and force the SPI into idle state. 0 Output buffer disabled 1 Output buffer enabled
1 SPISWAI	**SPI Stop in Wait Mode Bit** — This bit is used for power conservation while in wait mode. 0 SPI clock operates normally in wait mode 1 Stop SPI clock generation when in wait mode
0 SPC0	**Serial Pin Control Bit 0** — This bit enables bidirectional pin configurations as shown in Table 14-5. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state

Table 14-5. Bidirectional Pin Configurations

Pin Mode	SPC0	BIDIROE	MISO	MOSI
Master Mode of Operation				
Normal	0	X	Master In	Master Out
Bidirectional	1	0	MISO not used by SPI	Master In
		1		Master I/O
Slave Mode of Operation				
Normal	0	X	Slave Out	Slave In

Note: in Table 15-11, the IOS7 and IOSx should be set to 1

IOSx is the register TIOS bit x,

OC7Mx is the register OC7M bit x,

TCx is timer Input Capture/Output Compare register,

IOCx is channel x,

OMx/OLx is the register TCTL1/TCTL2,

OC7Dx is the register OC7D bit x.

IOCx = OC7Dx+ OMx/OLx, means that both OC7 event and OCx event will change channel x value.

Table 15-22. PAFLG Field Descriptions

Field	Description
1 PAOVF	**Pulse Accumulator Overflow Flag** — Set when the 16-bit pulse accumulator overflows from 0xFFFF to 0x0000. Clearing this bit requires wirting a one to this bit in the PAFLG register while TEN bit of TSCR1 register is set to one.
0 PAIF	**Pulse Accumulator Input edge Flag** — Set when the selected edge is detected at the IOC7 input pin.In event mode the event edge triggers PAIF and in gated time accumulation mode the trailing edge of the gate signal at the IOC7 input pin triggers PAIF. Clearing this bit requires writing a one to this bit in the PAFLG register while TEN bit of TSCR1 register is set to one. Any access to the PACNT register will clear all the flags in this register when TFFCA bit in register TSCR(0x0006) is set.

15.4.2 Input Capture

Clearing the I/O (input/output) select bit, IOSx, configures channel x as an input capture channel. The input capture function captures the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the timer transfers the value in the timer counter into the timer channel registers, TCx.

The minimum pulse width for the input capture input is greater than two bus clocks.

An input capture on channel x sets the CxF flag. The CxI bit enables the CxF flag to generate interrupt requests. Timer module must stay enabled (TEN bit of TSCR1 must be set to one) while clearing CxF (writing one to CxF).

15.4.3 Output Compare

Setting the I/O select bit, IOSx, configures channel x as an output compare channel. The output compare function can generate a periodic pulse with a programmable polarity, duration, and frequency. When the timer counter reaches the value in the channel registers of an output compare channel, the timer can set, clear, or toggle the channel pin. An output compare on channel x sets the CxF flag. The CxI bit enables the CxF flag to generate interrupt requests. Timer module must stay enabled (TEN bit of TSCR1 register must be set to one) while clearing CxF (writing one to CxF).

The output mode and level bits, OMx and OLx, select set, clear, toggle on output compare. Clearing both OMx and OLx disconnects the pin from the output logic.

Setting a force output compare bit, FOCx, causes an output compare on channel x. A forced output compare does not set the channel flag.

A successful output compare on channel 7 overrides output compares on all other output compare channels. The output compare 7 mask register masks the bits in the output compare 7 data register. The timer counter reset enable bit, TCRE, enables channel 7 output compares to reset the timer counter. A channel 7 output compare can reset the timer counter even if the IOC7 pin is being used as the pulse accumulator input.

Writing to the timer port bit of an output compare pin does not affect the pin state. The value written is stored in an internal latch. When the pin becomes available for general-purpose output, the last value written to the bit appears at the pin.

When TCRE is set and TC7 is not equal to 0, then TCNT will cycle from 0 to TC7. When TCNT reaches TC7 value, it will last only one bus cycle then reset to 0.

17.3.2.4 Flash Configuration Register (FCNFG)

The FCNFG register enables the Flash interrupts and gates the security backdoor key writes.

Module Base + 0x0003

	7	6	5	4	3	2	1	0
R	CBEIE	CCIE	KEYACC	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 17-7. Flash Configuration Register (FCNFG)

CBEIE, CCIE, and KEYACC are readable and writable while remaining bits read 0 and are not writable. KEYACC is only writable if the KEYEN bit in the FSEC register is set to the enabled state (see Section 17.3.2.2).

Table 17-7. FCNFG Field Descriptions

Field	Description
7 CBEIE	**Command Buffer Empty Interrupt Enable** — The CBEIE bit enables the interrupts in case of an empty command buffer in the Flash module. 0 Command Buffer Empty interrupts disabled 1 An interrupt will be requested whenever the CBEIF flag is set (see Section 17.3.2.6)
6 CCIE	**Command Complete Interrupt Enable** — The CCIE bit enables the interrupts in case of all commands being completed in the Flash module. 0 Command Complete interrupts disabled 1 An interrupt will be requested whenever the CCIF flag is set (see Section 17.3.2.6)
5 KEYACC	**Enable Security Key Writing**. 0 Flash writes are interpreted as the start of a command write sequence 1 Writes to the Flash array are interpreted as a backdoor key while reads of the Flash array return invalid data

17.3.2.5 Flash Protection Register (FPROT)

The FPROT register defines which Flash sectors are protected against program or erase.

Module Base + 0x0004

	7	6	5	4	3	2	1	0
R	FPOPEN	NV6	FPHDIS	FPHS1	FPHS0	NV2	NV1	NV0
W								
Reset	F	F	F	F	F	F	F	F

Figure 17-8. Flash Protection Register (FPROT)

The FPROT register is readable in normal and special modes. FPOPEN can only be written from a 1 to a 0. FPHS[1:0] can be written anytime until FPHDIS is cleared. The FPROT register is loaded from Flash address 0xFF0D during the reset sequence, indicated by F in Figure 17-8.

Figure 17-9 illustrates all possible protection scenarios. Although the protection scheme is loaded from the Flash array after reset, it is allowed to change in normal modes. This protection scheme can be used by applications requiring re-programming in single chip mode while providing as much protection as possible if no re-programming is required.

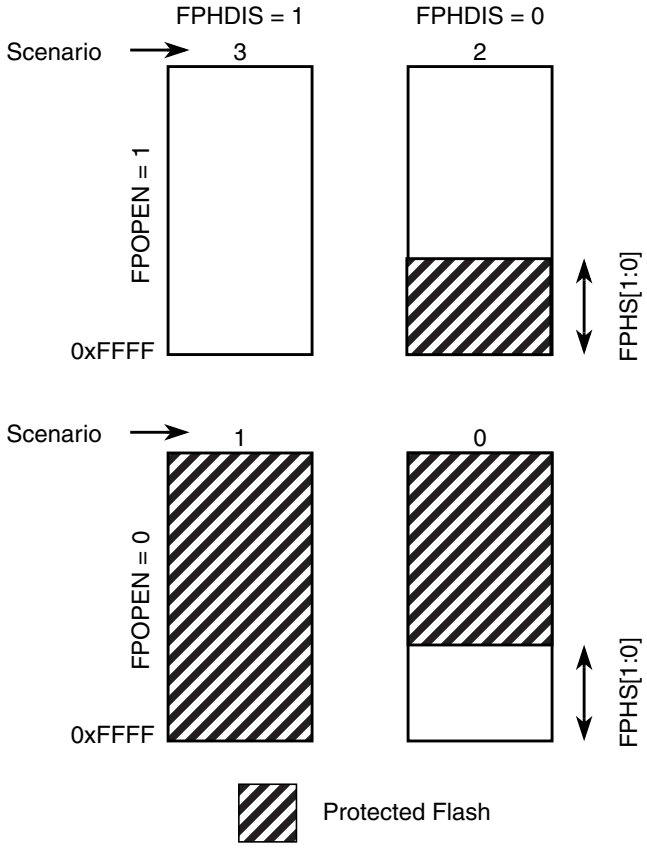

Figure 17-9. Flash Protection Scenarios

17.3.2.5.1 Flash Protection Restrictions

The general guideline is that protection can only be added, not removed. All valid transitions between Flash protection scenarios are specified in Table 17-11. Any attempt to write an invalid scenario to the FPROT register will be ignored and the FPROT register will remain unchanged. The contents of the FPROT register reflect the active protection scenario.

Table 17-11. Flash Protection Scenario Transitions

From Protection Scenario	To Protection Scenario[1]			
	0	1	2	3
0	X	X		
1		X		
2		X	X	

17.4.1.4 Illegal Flash Operations

17.4.1.4.1 Access Error

The ACCERR flag in the FSTAT register will be set during the command write sequence if any of the following illegal Flash operations are performed causing the command write sequence to immediately abort:

1. Writing to the Flash address space before initializing the FCLKDIV register
2. Writing a misaligned word or a byte to the valid Flash address space
3. Writing to the Flash address space while CBEIF is not set
4. Writing a second word to the Flash address space before executing a program or erase command on the previously written word
5. Writing to any Flash register other than FCMD after writing a word to the Flash address space
6. Writing a second command to the FCMD register before executing the previously written command
7. Writing an invalid command to the FCMD register
8. Writing to any Flash register other than FSTAT (to clear CBEIF) after writing to the FCMD register
9. The part enters stop mode and a program or erase command is in progress. The command is aborted and any pending command is killed
10. When security is enabled, a command other than mass erase originating from a non-secure memory or from the background debug mode is written to the FCMD register
11. A 0 is written to the CBEIF bit in the FSTAT register to abort a command write sequence.

The ACCERR flag will not be set if any Flash register is read during the command write sequence. If the Flash array is read during execution of an algorithm (CCIF=0), the Flash module will return invalid data and the ACCERR flag will not be set. If an ACCERR flag is set in the FSTAT register, the Flash command controller is locked. It is not possible to launch another command until the ACCERR flag is cleared.

17.4.1.4.2 Protection Violation

The PVIOL flag in the FSTAT register will be set during the command write sequence after the word write to the Flash address space if any of the following illegal Flash operations are performed, causing the command write sequence to immediately abort:

1. Writing a Flash address to program in a protected area of the Flash array (see Section 17.3.2.5).
2. Writing a Flash address to erase in a protected area of the Flash array.
3. Writing the mass erase command to the FCMD register while any protection is enabled.

If the PVIOL flag is set, the Flash command controller is locked. It is not possible to launch another command until the PVIOL flag is cleared.

FPHDIS is cleared. The FPROT register is loaded from Flash address 0xFF0D during the reset sequence, indicated by F in Figure 19-10.

To change the Flash protection that will be loaded on reset, the upper sector of the Flash array must be unprotected, then the Flash protection byte located at Flash address 0xFF0D must be written to.

A protected Flash sector is disabled by FPHDIS and FPLDIS while the size of the protected sector is defined by FPHS[1:0] and FPLS[1:0] in the FPROT register.

Trying to alter any of the protected areas will result in a protect violation error and the PVIOL flag will be set in the FSTAT register (see Section 19.3.2.6). A mass erase of the whole Flash array is only possible when protection is fully disabled by setting the FPOPEN, FPLDIS, and FPHDIS bits. An attempt to mass erase a Flash array while protection is enabled will set the PVIOL flag in the FSTAT register.

Table 19-9. FPROT Field Descriptions

Field	Description
7 FPOPEN	**Protection Function for Program or Erase** — It is possible using the FPOPEN bit to either select address ranges to be protected using FPHDIS, FPLDIS, FPHS[1:0] and FPLS[1:0] or to select the same ranges to be unprotected. When FPOPEN is set, FPxDIS enables the ranges to be protected, whereby clearing FPxDIS enables protection for the range specified by the corresponding FPxS[1:0] bits. When FPOPEN is cleared, FPxDIS defines unprotected ranges as specified by the corresponding FPxS[1:0] bits. In this case, setting FPxDIS enables protection. Thus the effective polarity of the FPxDIS bits is swapped by the FPOPEN bit as shown in Table 19-10. This function allows the main part of the Flash array to be protected while a small range can remain unprotected for EEPROM emulation. 0 The FPHDIS and FPLDIS bits define Flash address ranges to be unprotected 1 The FPHDIS and FPLDIS bits define Flash address ranges to be protected
6 NV6	**Nonvolatile Flag Bit** — The NV6 bit should remain in the erased state for future enhancements.
5 FPHDIS	**Flash Protection Higher Address Range Disable** — The FPHDIS bit determines whether there is a protected/unprotected area in the higher space of the Flash address map. 0 Protection/unprotection enabled 1 Protection/unprotection disabled
4–3 FPHS[1:0]	**Flash Protection Higher Address Size** — The FPHS[1:0] bits determine the size of the protected/unprotected sector as shown in Table 19-11. The FPHS[1:0] bits can only be written to while the FPHDIS bit is set.
2 FPLDIS	**Flash Protection Lower Address Range Disable** — The FPLDIS bit determines whether there is a protected/unprotected sector in the lower space of the Flash address map. 0 Protection/unprotection enabled 1 Protection/unprotection disabled
1–0 FPLS[1:0]	**Flash Protection Lower Address Size** — The FPLS[1:0] bits determine the size of the protected/unprotected sector as shown in Table 19-12. The FPLS[1:0] bits can only be written to while the FPLDIS bit is set.

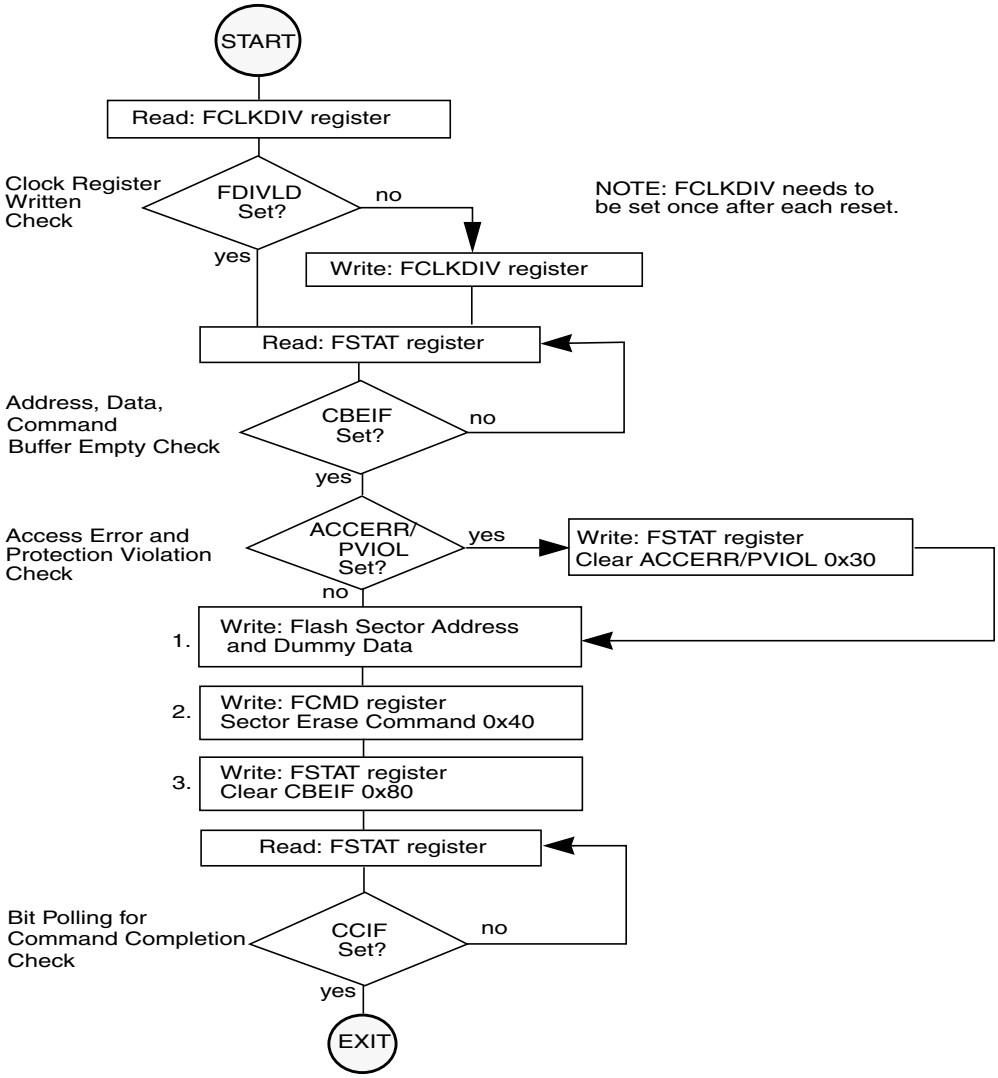

Figure 19-27. Example Sector Erase Command Flow

Module Base + 0x0009

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

Figure 20-16. Flash Address Low Register (FADDRLO)

In normal modes, all FABHI and FABLO bits read 0 and are not writable. In special modes, the FABHI and FABLO bits are readable and writable. For sector erase, the MCU address bits [9:0] are ignored. For mass erase, any address within the Flash array is valid to start the command.

20.3.2.10 Flash Data Register (FDATA)

FDATAHI and FDATALO are the Flash data registers.

Module Base + 0x000A

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

Figure 20-17. Flash Data High Register (FDATAHI)

Module Base + 0x000B

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

Figure 20-18. Flash Data Low Register (FDATALO)

In normal modes, all FDATAHI and FDATALO bits read 0 and are not writable. In special modes, all FDATAHI and FDATALO bits are readable and writable when writing to an address within the Flash address range.

20.3.2.11 RESERVED3

This register is reserved for factory testing and is not accessible to the user.

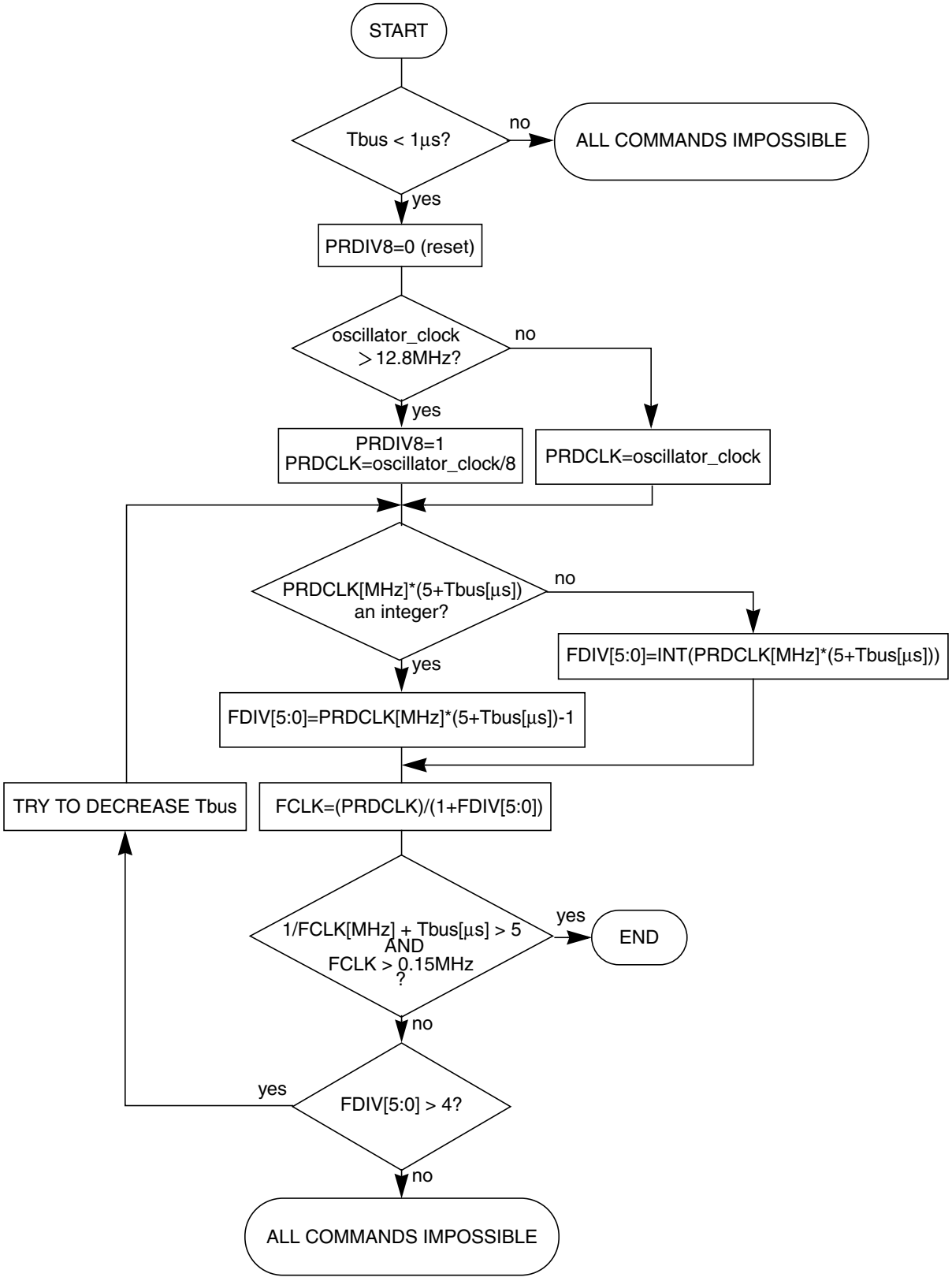

Figure 21-21. PRDIV8 and FDIV Bits Determination Procedure