



Welcome to [E-XFL.COM](http://E-XFL.COM)

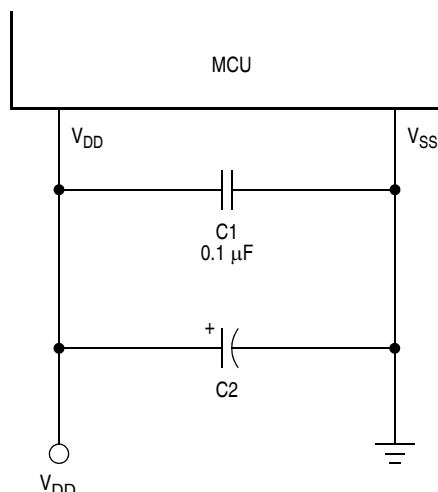
### What is "[Embedded - Microcontrollers](#)"?

"[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	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	CANbus, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	21
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908gz48mfje">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908gz48mfje</a>



Note: Component values shown represent typical applications.

**Figure 1-5. Power Supply Bypassing**

### 1.5.2 Oscillator Pins (OSC1 and OSC2)

OSC1 and OSC2 are the connections for an external crystal, resonator, or clock circuit. See Chapter 4 Clock Generator Module (CGM).

### 1.5.3 External Reset Pin ( $\overline{\text{RST}}$ )

A low on the  $\overline{\text{RST}}$  pin forces the MCU to a known startup state.  $\overline{\text{RST}}$  is bidirectional, allowing a reset of the entire system. It is driven low when any internal reset source is asserted. This pin contains an internal pullup resistor. See Chapter 15 System Integration Module (SIM).

### 1.5.4 External Interrupt Pin ( $\overline{\text{IRQ}}$ )

$\overline{\text{IRQ}}$  is an asynchronous external interrupt pin. This pin contains an internal pullup resistor. See Chapter 8 External Interrupt (IRQ).

### 1.5.5 CGM Power Supply Pins ( $V_{\text{DDA}}$ and $V_{\text{SSA}}$ )

$V_{\text{DDA}}$  and  $V_{\text{SSA}}$  are the power supply pins for the analog portion of the clock generator module (CGM). Decoupling of these pins should be as per the digital supply. See Chapter 4 Clock Generator Module (CGM).

### 1.5.6 External Filter Capacitor Pin (CGMXFC)

CGMXFC is an external filter capacitor connection for the CGM. See Chapter 4 Clock Generator Module (CGM).

### 1.5.7 ADC Power Supply/Reference Pins ( $V_{\text{DDAD}}/V_{\text{REFH}}$ and $V_{\text{SSAD}}/V_{\text{REFL}}$ )

$V_{\text{DDAD}}$  and  $V_{\text{SSAD}}$  are the power supply pins to the analog-to-digital converter (ADC).  $V_{\text{REFH}}$  and  $V_{\text{REFL}}$  are the reference voltage pins for the ADC.  $V_{\text{REFH}}$  is the high reference supply for the ADC, and by default the  $V_{\text{DDAD}}/V_{\text{REFH}}$  pin should be externally filtered and connected to the same voltage potential as  $V_{\text{DD}}$ .

## Memory

- E. The time between each FLASH address change (step 7 to step 7), or the time between the last FLASH address programmed to clearing the PGM bit (step 7 to step 10) must not exceed the maximum programming time,  $t_{\text{PROG}}$  maximum.*
- F. Be cautious when programming the FLASH-2 array to ensure that non-FLASH locations are not used as the address that is written to when selecting either the desired row address range in step 3 of the algorithm or the byte to be programmed in step 7 of the algorithm.*

### 2.7.7 Low-Power Modes

The WAIT and STOP instructions will place the MCU in low power-consumption standby modes.

#### 2.7.7.1 Wait Mode

Putting the MCU into wait mode while the FLASH is in read mode does not affect the operation of the FLASH memory directly; however, no memory activity will take place since the CPU is inactive.

The WAIT instruction should not be executed while performing a program or erase operation on the FLASH. Wait mode will suspend any FLASH program/erase operations and leave the memory in a standby mode.

#### 2.7.7.2 Stop Mode

Putting the MCU into stop mode while the FLASH is in read mode does not affect the operation of the FLASH memory directly; however, no memory activity will take place since the CPU is inactive.

The STOP instruction should not be executed while performing a program or erase operation on the FLASH. Stop mode will suspend any FLASH program/erase operations and leave the memory in a standby mode.

#### **NOTE**

*Standby mode is the power saving mode of the FLASH module, in which all internal control signals to the FLASH are inactive and the current consumption of the FLASH is minimum.*

### 3.8.2 ADC Data Register High and Data Register Low

#### 3.8.2.1 Left Justified Mode

In left justified mode, the ADRH register holds the eight MSBs of the 10-bit result. The ADRL register holds the two LSBs of the 10-bit result. All other bits read as 0. ADRH and ADRL are updated each time an ADC single channel conversion completes. Reading ADRH latches the contents of ADRL until ADRL is read. All subsequent results will be lost until the ADRH and ADRL reads are completed.

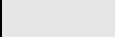
Address:	\$003D							ADRH
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
Write:								
Reset:	Unaffected by reset							
Address:	\$003E							ADRL
Read:	AD1	AD0	0	0	0	0	0	0
Write:								
Reset:	Unaffected by reset							
	 = Unimplemented							

Figure 3-5. ADC Data Register High (ADRH) and Low (ADRL)

#### 3.8.2.2 Right Justified Mode

In right justified mode, the ADRH register holds the two MSBs of the 10-bit result. All other bits read as 0. The ADRL register holds the eight LSBs of the 10-bit result. ADRH and ADRL are updated each time an ADC single channel conversion completes. Reading ADRH latches the contents of ADRL until ADRL is read. All subsequent results will be lost until the ADRH and ADRL reads are completed.


Address:	\$003D							ADRH
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	AD9	AD8
Write:								
Reset:	Unaffected by reset							
Address:	\$003E							ADRL
Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Write:								
Reset:	Unaffected by reset							
	 = Unimplemented							

Figure 3-6. ADC Data Register High (ADRH) and Low (ADRL)

Table 7-2. Opcode Map

	Bit Manipulation		Branch	Read-Modify-Write						Control		Register/Memory							
	DIR	DIR	REL	DIR	INH	INH	IX1	SP1	IX	INH	INH	IMM	DIR	EXT	IX2	SP2	IX1	SP1	IX
MSB LSB	0	1	2	3	4	5	6	9E6	7	8	9	A	B	C	D	9ED	E	9EE	F
0	BRSET0 3 DIR	BSET0 2 DIR	BRA 2 REL	NEG 2 DIR	NEGA 1 INH	NEGX 1 INH	NEG 2 IX1	NEG 3 SP1	NEG 1 IX	RTI 1 INH	BGE 2 REL	SUB 2 IMM	SUB 2 DIR	SUB 3 EXT	SUB 3 IX2	SUB 4 SP2	SUB 2 IX1	SUB 3 SP1	SUB 1 IX
1	BRCLR0 3 DIR	BCLR0 2 DIR	BRN 2 REL	CBEQ 3 DIR	CBEQA 3 IMM	CBEQX 3 IMM	CBEQ 3 IX1+	CBEQ 4 SP1	CBEQ 2 IX+	RTS 1 INH	BLT 2 REL	CMP 2 IMM	CMP 2 DIR	CMP 3 EXT	CMP 3 IX2	CMP 4 SP2	CMP 2 IX1	CMP 3 SP1	CMP 1 IX
2	BRSET1 3 DIR	BSET1 2 DIR	BHI 2 REL		MUL 1 INH	DIV 1 INH	NSA 1 INH		DAA 1 INH		BGT 2 REL	SBC 2 IMM	SBC 2 DIR	SBC 3 EXT	SBC 3 IX2	SBC 4 SP2	SBC 2 IX1	SBC 3 SP1	SBC 1 IX
3	BRCLR1 3 DIR	BCLR1 2 DIR	BLS 2 REL	COM 2 DIR	COMA 1 INH	COMX 1 INH	COM 2 IX1	COM 3 SP1	COM 1 IX	SWI 1 INH	BLE 2 REL	CPX 2 IMM	CPX 2 DIR	CPX 3 EXT	CPX 3 IX2	CPX 4 SP2	CPX 2 IX1	CPX 3 SP1	CPX 1 IX
4	BRSET2 3 DIR	BSET2 2 DIR	BCC 2 REL	LSR 2 DIR	LSRA 1 INH	LSRX 1 INH	LSR 2 IX1	LSR 3 SP1	LSR 1 IX	TAP 1 INH	TXS 1 INH	AND 2 IMM	AND 2 DIR	AND 3 EXT	AND 3 IX2	AND 4 SP2	AND 2 IX1	AND 3 SP1	AND 1 IX
5	BRCLR2 3 DIR	BCLR2 2 DIR	BCS 2 REL	STHX 2 DIR	LDHX 3 IMM	LDHX 2 DIR	CPHX 3 IMM		CPHX 2 DIR	TPA 1 INH	TSX 1 INH	BIT 2 IMM	BIT 2 DIR	BIT 3 EXT	BIT 3 IX2	BIT 4 SP2	BIT 2 IX1	BIT 3 SP1	BIT 1 IX
6	BRSET3 3 DIR	BSET3 2 DIR	BNE 2 REL	ROR 2 DIR	RORA 1 INH	RORX 1 INH	ROR 2 IX1	ROR 3 SP1	ROR 1 IX	PULA 1 INH		LDA 2 IMM	LDA 2 DIR	LDA 3 EXT	LDA 3 IX2	LDA 4 SP2	LDA 2 IX1	LDA 3 SP1	LDA 1 IX
7	BRCLR3 3 DIR	BCLR3 2 DIR	BEQ 2 REL	ASR 2 DIR	ASRA 1 INH	ASRX 1 INH	ASR 2 IX1	ASR 3 SP1	ASR 1 IX	PSHA 1 INH	TAX 1 INH	AIS 2 IMM	STA 2 DIR	STA 3 EXT	STA 3 IX2	STA 4 SP2	STA 2 IX1	STA 3 SP1	STA 1 IX
8	BRSET4 3 DIR	BSET4 2 DIR	BHCC 2 REL	LSL 2 DIR	LSLA 1 INH	LSLX 1 INH	LSL 2 IX1	LSL 3 SP1	LSL 1 IX	PULX 1 INH	CLC 1 INH	EOR 2 IMM	EOR 2 DIR	EOR 3 EXT	EOR 3 IX2	EOR 4 SP2	EOR 2 IX1	EOR 3 SP1	EOR 1 IX
9	BRCLR4 3 DIR	BCLR4 2 DIR	BHCS 2 REL	ROL 2 DIR	ROLA 1 INH	ROLX 1 INH	ROL 2 IX1	ROL 3 SP1	ROL 1 IX	PSHX 1 INH	SEC 1 INH	ADC 2 IMM	ADC 2 DIR	ADC 3 EXT	ADC 3 IX2	ADC 4 SP2	ADC 2 IX1	ADC 3 SP1	ADC 1 IX
A	BRSET5 3 DIR	BSET5 2 DIR	BPL 2 REL	DEC 2 DIR	DECA 1 INH	DECX 1 INH	DEC 2 IX1	DEC 3 SP1	DEC 1 IX	PULH 1 INH	CLI 1 INH	ORA 2 IMM	ORA 2 DIR	ORA 3 EXT	ORA 3 IX2	ORA 4 SP2	ORA 2 IX1	ORA 3 SP1	ORA 1 IX
B	BRCLR5 3 DIR	BCLR5 2 DIR	BMI 2 REL	DBNZ 3 DIR	DBNZA 2 INH	DBNZX 2 INH	DBNZ 3 IX1	DBNZ 4 SP1	DBNZ 2 IX	PSHH 1 INH	SEI 1 INH	ADD 2 IMM	ADD 2 DIR	ADD 3 EXT	ADD 3 IX2	ADD 4 SP2	ADD 2 IX1	ADD 3 SP1	ADD 1 IX
C	BRSET6 3 DIR	BSET6 2 DIR	BMC 2 REL	INC 2 DIR	INCA 1 INH	INCX 1 INH	INC 2 IX1	INC 3 SP1	INC 1 IX	CLRH 1 INH	RSP 1 INH		JMP 2 DIR	JMP 3 EXT	JMP 3 IX2		JMP 2 IX1		JMP 1 IX
D	BRCLR6 3 DIR	BCLR6 2 DIR	BMS 2 REL	TST 2 DIR	TSTA 1 INH	TSTX 1 INH	TST 2 IX1	TST 3 SP1	TST 1 IX		NOP 1 INH	BSR 2 REL	JSR 2 DIR	JSR 3 EXT	JSR 3 IX2		JSR 2 IX1		JSR 1 IX
E	BRSET7 3 DIR	BSET7 2 DIR	BIL 2 REL		MOV 3 DD	MOV 2 DIX+	MOV 3 IMD		MOV 2 IX+D	STOP 1 INH	*	LDX 2 IMM	LDX 2 DIR	LDX 3 EXT	LDX 3 IX2	LDX 4 SP2	LDX 2 IX1	LDX 3 SP1	LDX 1 IX
F	BRCLR7 3 DIR	BCLR7 2 DIR	BIH 2 REL	CLR 2 DIR	CLRA 1 INH	CLRAX 1 INH	CLR 2 IX1	CLR 3 SP1	CLR 1 IX	WAIT 1 INH	TXA 1 INH	AIX 2 IMM	STX 2 DIR	STX 3 EXT	STX 3 IX2	STX 4 SP2	STX 2 IX1	STX 3 SP1	STX 1 IX

INH Inherent  
IMM Immediate  
DIR Direct  
EXT Extended  
DD Direct-Direct  
IX+D Indexed-Direct  
REL Relative  
IX Indexed, No Offset  
IX1 Indexed, 8-Bit Offset  
IX2 Indexed, 16-Bit Offset  
IMD Immediate-Direct  
DIX+ Direct-Indexed

SP1 Stack Pointer, 8-Bit Offset  
SP2 Stack Pointer, 16-Bit Offset  
IX+ Indexed, No Offset with Post Increment  
IX1+ Indexed, 1-Byte Offset with Post Increment

Low Byte of Opcode in Hexadecimal

MSB LSB	0	High Byte of Opcode in Hexadecimal
0	5 BRSET0 3 DIR	Cycles Opcode Mnemonic Number of Bytes / Addressing Mode

## Chapter 9

# Keyboard Interrupt Module (KBI)

### 9.1 Introduction

The keyboard interrupt module (KBI) provides eight independently maskable external interrupts which are accessible via PTA0–PTA7. When a port pin is enabled for keyboard interrupt function, an internal pullup/pulldown device is also enabled on the pin.

### 9.2 Features

Features include:

- Eight keyboard interrupt pins with separate keyboard interrupt enable bits and one keyboard interrupt mask
- Hysteresis buffers
- Programmable edge-only or edge- and level- interrupt sensitivity
- Edge detect programmable for rising or falling edges
- Level detect programmable for high or low levels
- Exit from low-power modes
- Pullup/pulldown device automatically configured based on polarity of edge/level selection

### 9.3 Functional Description

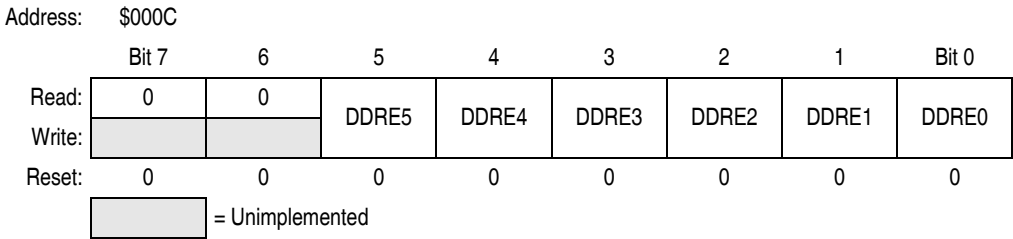
Writing to the KBIE7–KBIE0 bits in the keyboard interrupt enable register independently enables or disables each port A pin as a keyboard interrupt pin. Enabling a keyboard interrupt pin also enables its internal pullup/pulldown device. On falling edge or low level selection a pullup device is configured. On rising edge or high level selection a pulldown device is configured.

- A falling edge is detected when an enabled keyboard input signal is seen as a 1 (the deasserted level) during one bus cycle and then a 0 (the asserted level) during the next cycle.
- A rising edge is detected when the input signal is seen as a 0 during one bus cycle and then a 1 during the next cycle.

A keyboard interrupt is latched when one or more keyboard pins are asserted. The MODEK bit in the keyboard status and control register controls the triggering mode of the keyboard interrupt.

## 13.7.2 Data Direction Register E

Data direction register E (DDRE) determines whether each port E pin is an input or an output. Writing a 1 to a DDRE bit enables the output buffer for the corresponding port E pin; a 0 disables the output buffer.



**Figure 13-18. Data Direction Register E (DDRE)**

### DDRE5–DDRE0 — Data Direction Register E Bits

These read/write bits control port E data direction. Reset clears DDRE5–DDRE0, configuring all port E pins as inputs.

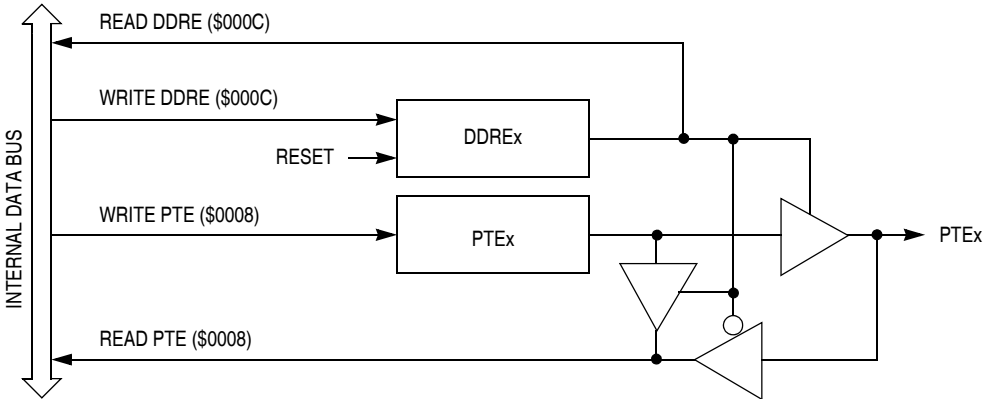
- 1 = Corresponding port E pin configured as output
- 0 = Corresponding port E pin configured as input

#### NOTE

*Avoid glitches on port E pins by writing to the port E data register before changing data direction register E bits from 0 to 1.*

Figure 13-19 shows the port E I/O logic.

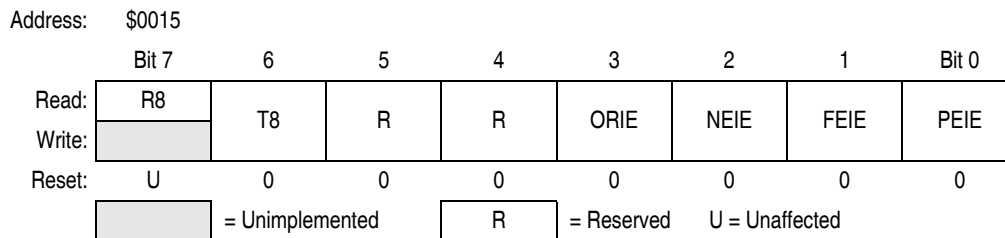
When bit DDREx is a 1, reading address \$0008 reads the PTE<sub>x</sub> data latch. When bit DDREx is a 0, reading address \$0008 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 13-6 summarizes the operation of the port E pins.



**Figure 13-19. Port E I/O Circuit**  
**Table 13-6. Port E Pin Functions**

DDRE Bit	PTE Bit	I/O Pin Mode	Accesses to DDRE	Accesses to PTE	
			Read/Write	Read	Write
0	X <sup>(1)</sup>	Input, Hi-Z <sup>(2)</sup>	DDRE5–DDRE0	Pin	PTE5–PTE0 <sup>(3)</sup>
1	X	Output	DDRE5–DDRE0	PTE5–PTE0	PTE5–PTE0

1. X = Don't care
2. Hi-Z = High impedance
3. Writing affects data register, but does not affect input.



**Figure 14-12. ESCI Control Register 3 (SCC3)**

## R8 — Received Bit 8

When the ESCI is receiving 9-bit characters, R8 is the read-only ninth bit (bit 8) of the received character. R8 is received at the same time that the SCDR receives the other 8 bits.

When the ESCI is receiving 8-bit characters, R8 is a copy of the eighth bit (bit 7). Reset has no effect on the R8 bit.

## T8 — Transmitted Bit 8

When the ESCI is transmitting 9-bit characters, T8 is the read/write ninth bit (bit 8) of the transmitted character. T8 is loaded into the transmit shift register at the same time that the SCDR is loaded into the transmit shift register. Reset clears the T8 bit.

## ORIE — Receiver Overrun Interrupt Enable Bit

This read/write bit enables ESCI error CPU interrupt requests generated by the receiver overrun bit, OR. Reset clears ORIE.

- 1 = ESCI error CPU interrupt requests from OR bit enabled
- 0 = ESCI error CPU interrupt requests from OR bit disabled

## NEIE — Receiver Noise Error Interrupt Enable Bit

This read/write bit enables ESCI error CPU interrupt requests generated by the noise error bit, NE. Reset clears NEIE.

- 1 = ESCI error CPU interrupt requests from NE bit enabled
- 0 = ESCI error CPU interrupt requests from NE bit disabled

## FEIE — Receiver Framing Error Interrupt Enable Bit

This read/write bit enables ESCI error CPU interrupt requests generated by the framing error bit, FE. Reset clears FEIE.

- 1 = ESCI error CPU interrupt requests from FE bit enabled
- 0 = ESCI error CPU interrupt requests from FE bit disabled

## PEIE — Receiver Parity Error Interrupt Enable Bit

This read/write bit enables ESCI receiver CPU interrupt requests generated by the parity error bit, PE. Reset clears PEIE.

- 1 = ESCI error CPU interrupt requests from PE bit enabled
- 0 = ESCI error CPU interrupt requests from PE bit disabled





# Chapter 15

## System Integration Module (SIM)

### 15.1 Introduction

This section describes the system integration module (SIM). Together with the central processor unit (CPU), the SIM controls all microcontroller unit (MCU) activities. A block diagram of the SIM is shown in Figure 15-1. Table 15-1 is a summary of the SIM input/output (I/O) registers. The SIM is a system state controller that coordinates CPU and exception timing.

The SIM is responsible for:

- Bus clock generation and control for CPU and peripherals:
  - Stop/wait/reset/break entry and recovery
  - Internal clock control
- Master reset control, including power-on reset (POR) and computer operating properly (COP) timeout
- Interrupt arbitration

Table 15-1 shows the internal signal names used in this section.

**Table 15-1. Signal Name Conventions**

Signal Name	Description
CGMXCLK	Buffered version of OSC1 from clock generator module (CGM)
CGMVCLK	PLL output
CGMOUT	PLL-based or OSC1-based clock output from CGM module (Bus clock = CGMOUT divided by two)
IAB	Internal address bus
IDB	Internal data bus
PORRST	Signal from the power-on reset module to the SIM
IRST	Internal reset signal
R/ $\overline{W}$	Read/write signal

## 15.4 SIM Counter

The SIM counter is used by the power-on reset module (POR) and in stop mode recovery to allow the oscillator time to stabilize before enabling the internal bus clocks. The SIM counter also serves as a prescaler for the computer operating properly (COP) module. The SIM counter overflow supplies the clock for the COP module. The SIM counter is 12 bits long.

### 15.4.1 SIM Counter During Power-On Reset

The power-on reset module (POR) detects power applied to the MCU. At power-on, the POR circuit asserts the signal PORRST. Once the SIM is initialized, it enables the clock generation module (CGM) to drive the bus clock state machine.

### 15.4.2 SIM Counter During Stop Mode Recovery

The SIM counter also is used for stop mode recovery. The STOP instruction clears the SIM counter. After an interrupt, break, or reset, the SIM senses the state of the short stop recovery bit, SSREC, in the CONFIG1 register. If the SSREC bit is a 1, then the stop recovery is reduced from the normal delay of 4096 CGMXCLK cycles down to 32 CGMXCLK cycles. This is ideal for applications using crystals with the OSCENINSTOP bit set. External crystal applications should use the full stop recovery time, SSREC cleared, with the OSCENINSTOP bit cleared. See 5.2 Functional Description.

### 15.4.3 SIM Counter and Reset States

External reset has no effect on the SIM counter. See 15.6.2 Stop Mode for details. The SIM counter is free-running after all reset states. See 15.3.2 Active Resets from Internal Sources for counter control and internal reset recovery sequences.

## 15.5 Exception Control

Normal, sequential program execution can be changed in three different ways:

- Interrupts:
  - Maskable hardware CPU interrupts
  - Non-maskable software interrupt instruction (SWI)
- Reset
- Break interrupts

### 15.5.1 Interrupts

At the beginning of an interrupt, the CPU saves the CPU register contents on the stack and sets the interrupt mask (I bit) to prevent additional interrupts. At the end of an interrupt, the RTI instruction recovers the CPU register contents from the stack so that normal processing can resume. Figure 15-8 shows interrupt entry timing. Figure 15-9 shows interrupt recovery timing.

Interrupts are latched, and arbitration is performed in the SIM at the start of interrupt processing. The arbitration result is a constant that the CPU uses to determine which vector to fetch. Once an interrupt is latched by the SIM, no other interrupt can take precedence, regardless of priority, until the latched interrupt is serviced (or the I bit is cleared). See Figure 15-10.

## Serial Peripheral Interface (SPI) Module

input ( $\overline{SS}$ ) is low, so that only the selected slave drives to the master. The  $\overline{SS}$  pin of the master is not shown but is assumed to be inactive. The  $\overline{SS}$  pin of the master must be high or must be reconfigured as general-purpose I/O not affecting the SPI. (See 16.6.2 Mode Fault Error.) When  $CPHA = 0$ , the first SPSCCK edge is the MSB capture strobe. Therefore, the slave must begin driving its data before the first SPSCCK edge, and a falling edge on the  $\overline{SS}$  pin is used to start the slave data transmission. The slave's  $\overline{SS}$  pin must be toggled back to high and then low again between each byte transmitted as shown in Figure 16-6.

When  $CPHA = 0$  for a slave, the falling edge of  $\overline{SS}$  indicates the beginning of the transmission. This causes the SPI to leave its idle state and begin driving the MISO pin with the MSB of its data. Once the transmission begins, no new data is allowed into the shift register from the transmit data register. Therefore, the SPI data register of the slave must be loaded with transmit data before the falling edge of  $\overline{SS}$ . Any data written after the falling edge is stored in the transmit data register and transferred to the shift register after the current transmission.

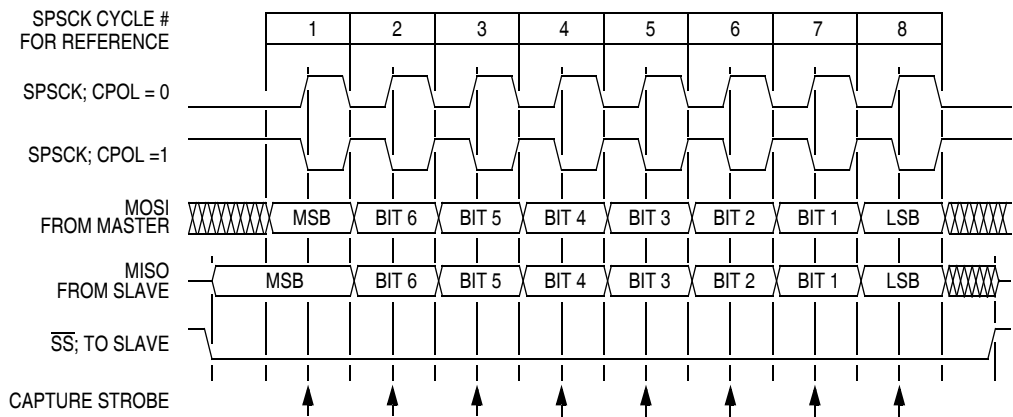


Figure 16-5. Transmission Format ( $CPHA = 0$ )

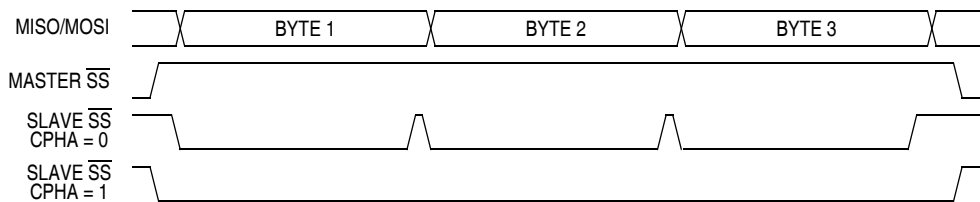


Figure 16-6.  $CPHA/\overline{SS}$  Timing

### 16.4.3 Transmission Format When $CPHA = 1$

Figure 16-7 shows an SPI transmission in which  $CPHA = 1$ . The figure should not be used as a replacement for data sheet parametric information. Two waveforms are shown for SPSCCK: one for  $CPOL = 0$  and another for  $CPOL = 1$ . The diagram may be interpreted as a master or slave timing diagram since the serial clock (SPSCCK), master in/slave out (MISO), and master out/slave in (MOSI) pins are directly connected 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  $\overline{SS}$  line is the slave select input to the slave. The slave SPI drives its MISO output only when its slave select input ( $\overline{SS}$ ) is low, so that only the selected slave drives to the master. The  $\overline{SS}$  pin of the master is not shown but is assumed to be inactive. The  $\overline{SS}$

**Table 16-2. SPI Configuration**

SPE	SPMSTR	MODFEN	SPI Configuration	Function of $\overline{SS}$ Pin
0	X <sup>(1)</sup>	X	Not enabled	General-purpose I/O; $\overline{SS}$ ignored by SPI
1	0	X	Slave	Input-only to SPI
1	1	0	Master without MODF	General-purpose I/O; $\overline{SS}$ ignored by SPI
1	1	1	Master with MODF	Input-only to SPI

1. X = Don't care

## 16.12 I/O Registers

Three registers control and monitor SPI operation:

- SPI control register (SPCR)
- SPI status and control register (SPSCR)
- SPI data register (SPDR)

### 16.12.1 SPI Control Register

The SPI control register:

- Enables SPI module interrupt requests
- Configures the SPI module as master or slave
- Selects serial clock polarity and phase
- Configures the SPSCCK, MOSI, and MISO pins as open-drain outputs
- Enables the SPI module

Address: \$0010

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SPRIE	R	SPMSTR	CPOL	CPHA	SPWOM	SPE	SPTIE
Write:								
Reset:	0	0	1	0	1	0	0	0
	R		= Reserved					

**Figure 16-14. SPI Control Register (SPCR)**

#### SPRIE — SPI Receiver Interrupt Enable Bit

This read/write bit enables CPU interrupt requests generated by the SPRF bit. The SPRF bit is set when a byte transfers from the shift register to the receive data register. Reset clears the SPRIE bit.

- 1 = SPRF CPU interrupt requests enabled
- 0 = SPRF CPU interrupt requests disabled

#### SPMSTR — SPI Master Bit

This read/write bit selects master mode operation or slave mode operation. Reset sets the SPMSTR bit.

- 1 = Master mode
- 0 = Slave mode

### SPRF — SPI Receiver Full Bit

This clearable, read-only flag is set each time a byte transfers from the shift register to the receive data register. SPRF generates a CPU interrupt request if the SPRIE bit in the SPI control register is set also.

During an SPRF CPU interrupt, the CPU clears SPRF by reading the SPI status and control register with SPRF set and then reading the SPI data register.

Reset clears the SPRF bit.

- 1 = Receive data register full
- 0 = Receive data register not full

### ERRIE — Error Interrupt Enable Bit

This read/write bit enables the MODF and OVRF bits to generate CPU interrupt requests. Reset clears the ERRIE bit.

- 1 = MODF and OVRF can generate CPU interrupt requests
- 0 = MODF and OVRF cannot generate CPU interrupt requests

### OVRF — Overflow Bit

This clearable, read-only flag is set if software does not read the byte in the receive data register before the next full byte enters the shift register. In an overflow condition, the byte already in the receive data register is unaffected, and the byte that shifted in last is lost. Clear the OVRF bit by reading the SPI status and control register with OVRF set and then reading the receive data register. Reset clears the OVRF bit.

- 1 = Overflow
- 0 = No overflow

### MODF — Mode Fault Bit

This clearable, read-only flag is set in a slave SPI if the  $\overline{SS}$  pin goes high during a transmission with MODFEN set. In a master SPI, the MODF flag is set if the  $\overline{SS}$  pin goes low at any time with the MODFEN bit set. Clear MODF by reading the SPI status and control register (SPSCR) with MODF set and then writing to the SPI control register (SPCR). Reset clears the MODF bit.

- 1 =  $\overline{SS}$  pin at inappropriate logic level
- 0 =  $\overline{SS}$  pin at appropriate logic level

### SPTE — SPI Transmitter Empty Bit

This clearable, read-only flag is set each time the transmit data register transfers a byte into the shift register. SPTE generates an SPTE CPU interrupt request if SPTIE in the SPI control register is set also.

#### NOTE

*Do not write to the SPI data register unless SPTE is high.*

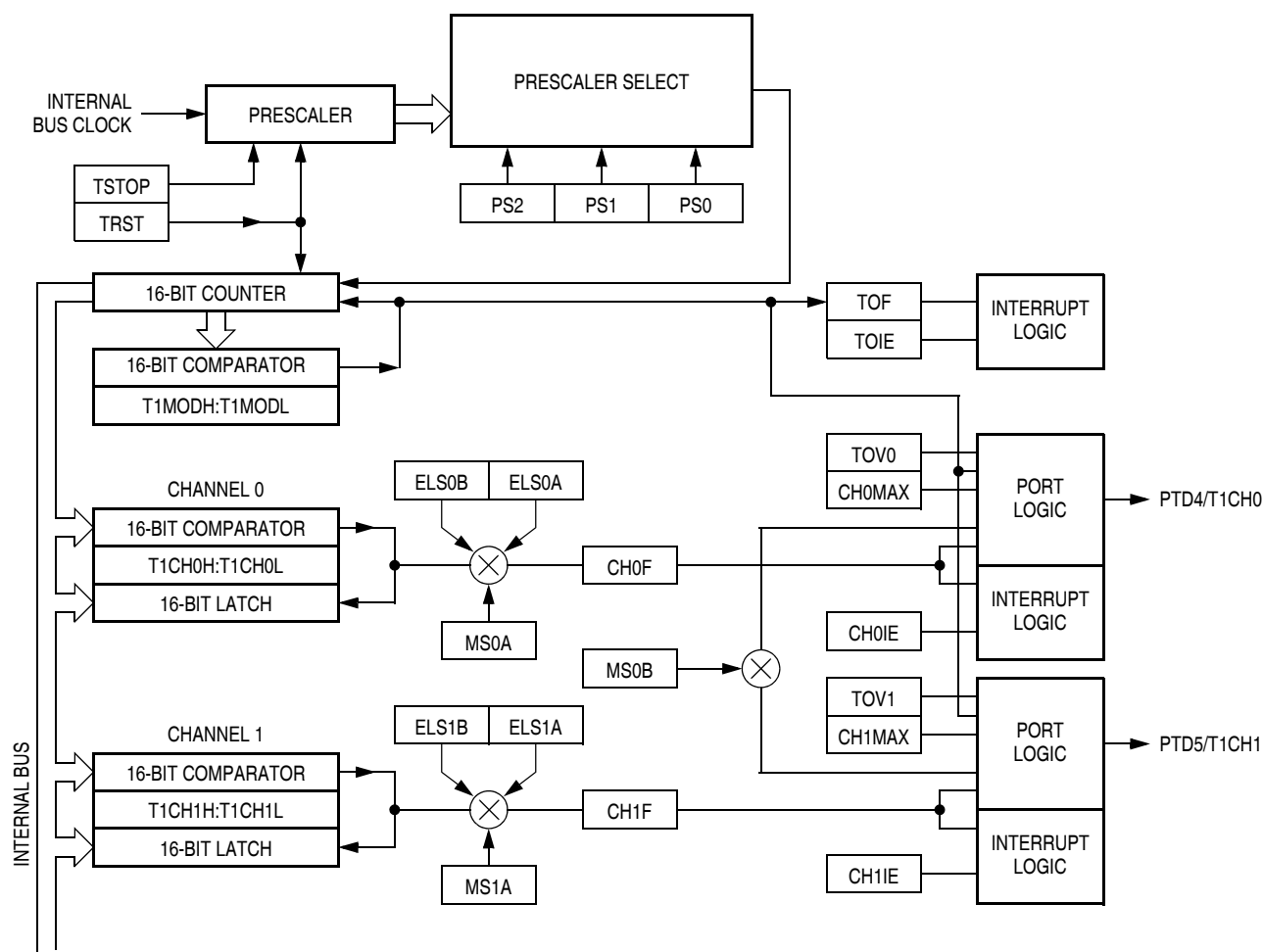
During an SPTE CPU interrupt, the CPU clears SPTE by writing to the transmit data register.

Reset sets the SPTE bit.

- 1 = Transmit data register empty
- 0 = Transmit data register not empty

### MODFEN — Mode Fault Enable Bit

This read/write bit, when set, allows the MODF flag to be set. If the MODF flag is set, clearing MODFEN does not clear the MODF flag. If the SPI is enabled as a master and the MODFEN bit is 0, then the  $\overline{SS}$  pin is available as a general-purpose I/O.



### Figure 18-2. TIM1 Block Diagram

### 18.3.3 Output Compare

With the output compare function, the TIM1 can generate a periodic pulse with a programmable polarity, duration, and frequency. When the counter reaches the value in the registers of an output compare channel, the TIM1 can set, clear, or toggle the channel pin. Output compares can generate TIM1 CPU interrupt requests.

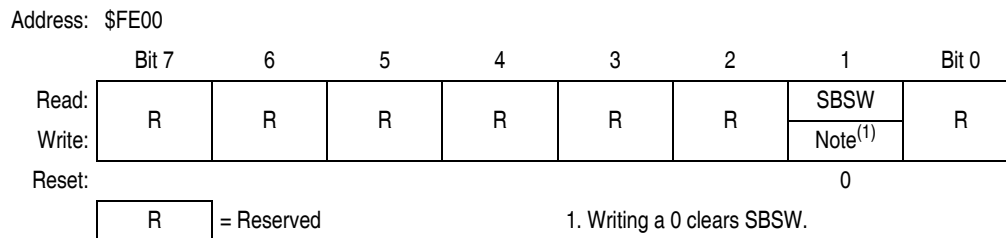
### 18.3.3.1 Unbuffered Output Compare

Any output compare channel can generate unbuffered output compare pulses as described in 18.3.3 Output Compare. The pulses are unbuffered because changing the output compare value requires writing the new value over the old value currently in the TIM1 channel registers.

An unsynchronized write to the TIM1 channel registers to change an output compare value could cause incorrect operation for up to two counter overflow periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that counter overflow period. Also, using a TIM1 overflow interrupt routine to write a new, smaller output compare value may cause the compare to be missed. The TIM1 may pass the new value before it is written.

### 20.2.2.3 Break Status Register

The break status register (BSR) contains a flag to indicate that a break caused an exit from wait mode. This register is only used in emulation mode.



**Figure 20-7. Break Status Register (BSR)**

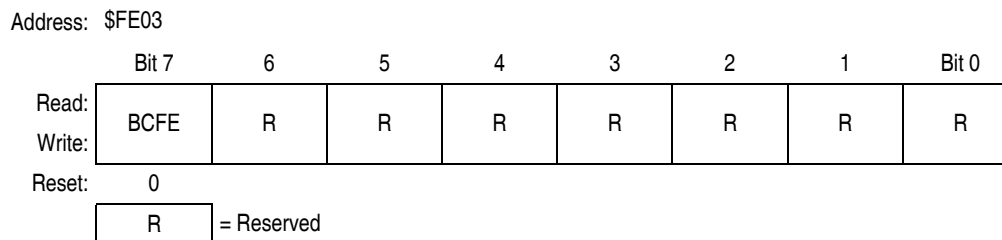
#### SBSW — SIM Break Stop/Wait

SBSW can be read within the break state SWI routine. The user can modify the return address on the stack by subtracting one from it.

- 1 = Wait mode was exited by break interrupt
- 0 = Wait mode was not exited by break interrupt

### 20.2.2.4 Break Flag Control Register

The break control register (BFCR) contains a bit that enables software to clear status bits while the MCU is in a break state.



**Figure 20-8. Break Flag Control Register (BFCR)**

#### BCFE — Break Clear Flag Enable Bit

This read/write bit enables software to clear status bits by accessing status registers while the MCU is in a break state. To clear status bits during the break state, the BCFE bit must be set.

- 1 = Status bits clearable during break
- 0 = Status bits not clearable during break

### 20.2.3 Low-Power Modes

The WAIT and STOP instructions put the MCU in low power- consumption standby modes. If enabled, the break module will remain enabled in wait and stop modes. However, since the internal address bus does not increment in these modes, a break interrupt will never be triggered.



## 20.3 Monitor Module (MON)

The monitor module allows debugging and programming of the microcontroller unit (MCU) through a single-wire interface with a host computer. Monitor mode entry can be achieved without use of the higher test voltage,  $V_{TST}$ , as long as vector addresses \$FFFE and \$FFFF are blank, thus reducing the hardware requirements for in-circuit programming.

Features of the monitor module include:

- Normal user-mode pin functionality
- One pin dedicated to serial communication between MCU and host computer
- Standard non-return-to-zero (NRZ) communication with host computer
- Standard communication baud rate (7200 @ 2-MHz bus frequency)
- Execution of code in random-access memory (RAM) or FLASH
- FLASH memory security feature<sup>(1)</sup>
- FLASH memory programming interface
- Monitor mode entry without high voltage,  $V_{TST}$ , if reset vector is blank (\$FFFE and \$FFFF contain \$FF)
- Normal monitor mode entry if  $V_{TST}$  is applied to  $\overline{IRQ}$

### 20.3.1 Functional Description

Figure 20-9 shows a simplified diagram of the monitor mode.

The monitor module receives and executes commands from a host computer.

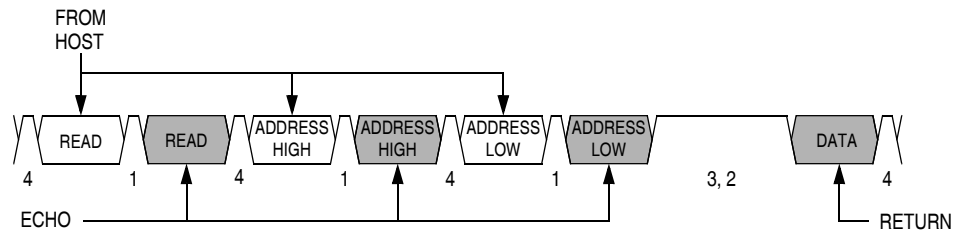
Figure 20-10 and Figure 20-11 show example circuits used to enter monitor mode and communicate with a host computer via a standard RS-232 interface.

Simple monitor commands can access any memory address. In monitor mode, the MCU can execute code downloaded into RAM by a host computer while most MCU pins retain normal operating mode functions. All communication between the host computer and the MCU is through the PTA0 pin. A level-shifting and multiplexing interface is required between PTA0 and the host computer. PTA0 is used in a wired-OR configuration and requires a pullup resistor.

Table 20-1 shows the pin conditions for entering monitor mode. As specified in the table, monitor mode may be entered after a power-on reset (POR) and will allow communication at 7200 baud provided one of the following sets of conditions is met:

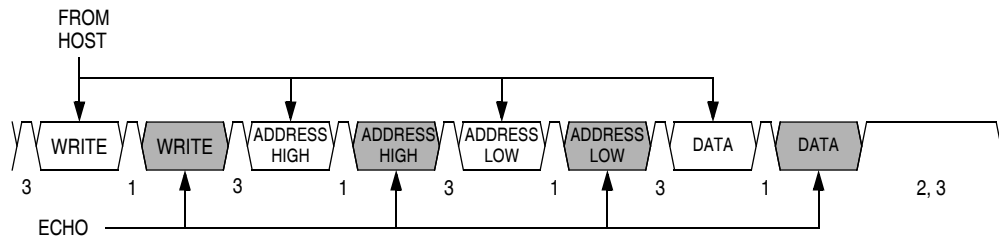
- If \$FFFE and \$FFFF does not contain \$FF (programmed state):
  - The external clock is 4.0 MHz (7200 baud)
  - PTB4 = low
  - $\overline{IRQ} = V_{TST}$
- If \$FFFE and \$FFFF do not contain \$FF (programmed state):
  - The external clock is 8.0 MHz (7200 baud)
  - PTB4 = high
  - $IRQ = V_{TST}$
- If \$FFFE and \$FFFF contain \$FF (erased state):
  - The external clock is 8.0 MHz (7200 baud)
  - $\overline{IRQ} = V_{DD}$  (this can be implemented through the internal  $\overline{IRQ}$  pullup) or  $V_{SS}$

1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the FLASH difficult for unauthorized users.



Notes:  
 1 = Echo delay, approximately 2 bit times  
 2 = Data return delay, approximately 2 bit times  
 3 = Cancel command delay, 11 bit times  
 4 = Wait 1 bit time before sending next byte.

**Figure 20-14. Read Transaction**



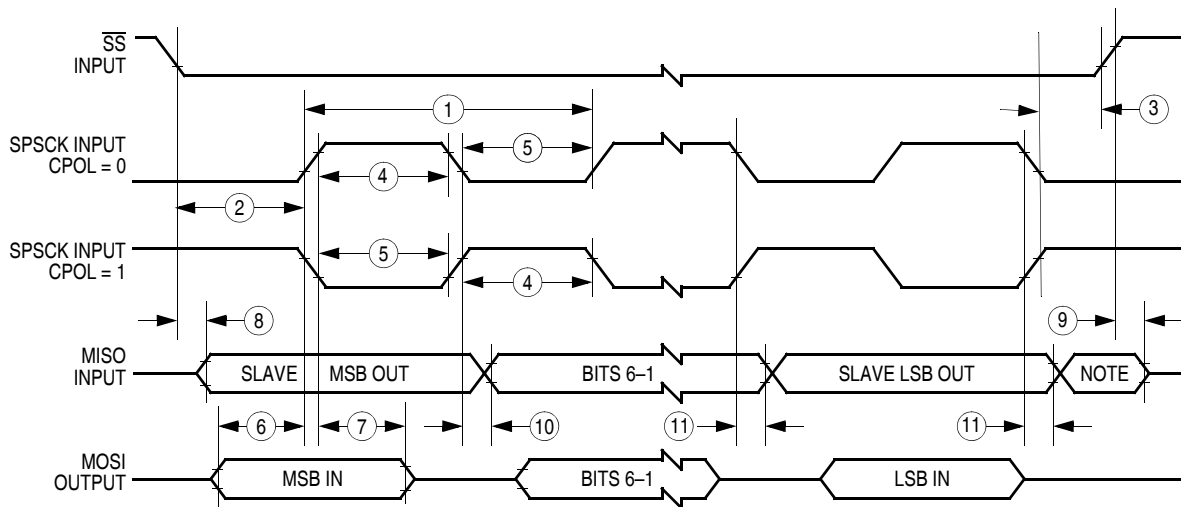
Notes:  
 1 = Echo delay, approximately 2 bit times  
 2 = Cancel command delay, 11 bit times  
 3 = Wait 1 bit time before sending next byte.

**Figure 20-15. Write Transaction**

A brief description of each monitor mode command is given in Table 20-3 through Table 20-8.

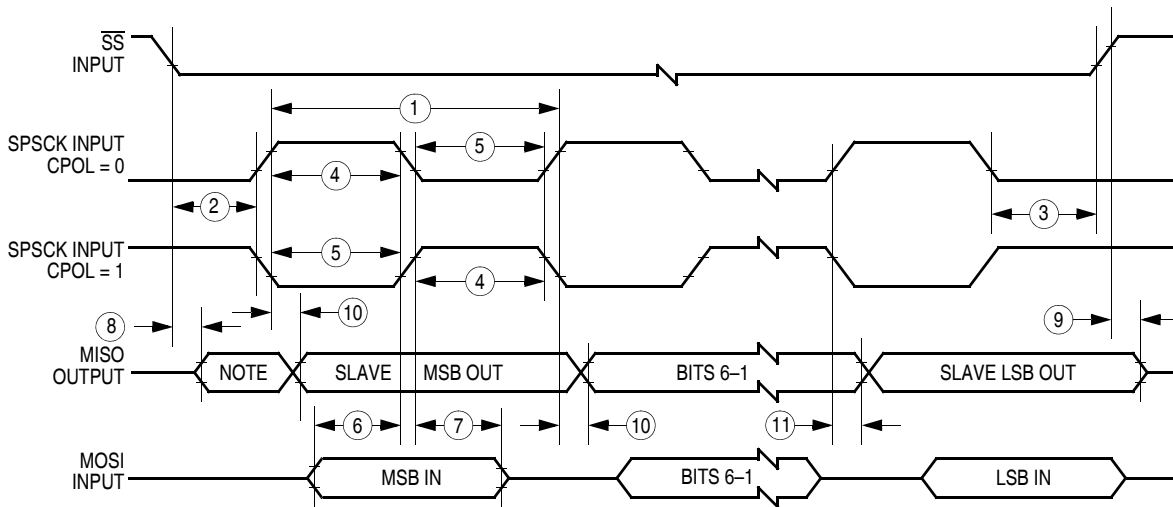
**Table 20-3. READ (Read Memory) Command**

Description	Read byte from memory
Operand	2-byte address in high-byte:low-byte order
Data Returned	Returns contents of specified address
Opcode	\$4A
<p style="text-align: center;"><b>Command Sequence</b></p>	



Note: Not defined but normally MSB of character just received

#### a) SPI Slave Timing (CPHA = 0)



Note: Not defined but normally LSB of character previously transmitted

#### b) SPI Slave Timing (CPHA = 1)

**Figure 21-3. SPI Slave Timing**

# 21.14 Timer Interface Module Characteristics

Characteristic	Symbol	Min	Max	Unit
Timer input capture pulse width	$t_{TH}, t_{TL}$	2	—	$t_{cyc}$
Timer input capture period	$t_{TLTL}$	Note <sup>(1)</sup>	—	$t_{cyc}$
Timer input clock pulse width	$t_{TCL}, t_{TCH}$	$t_{cyc} + 5$	—	ns

1. The minimum period is the number of cycles it takes to execute the interrupt service routine plus 1  $t_{cyc}$ .

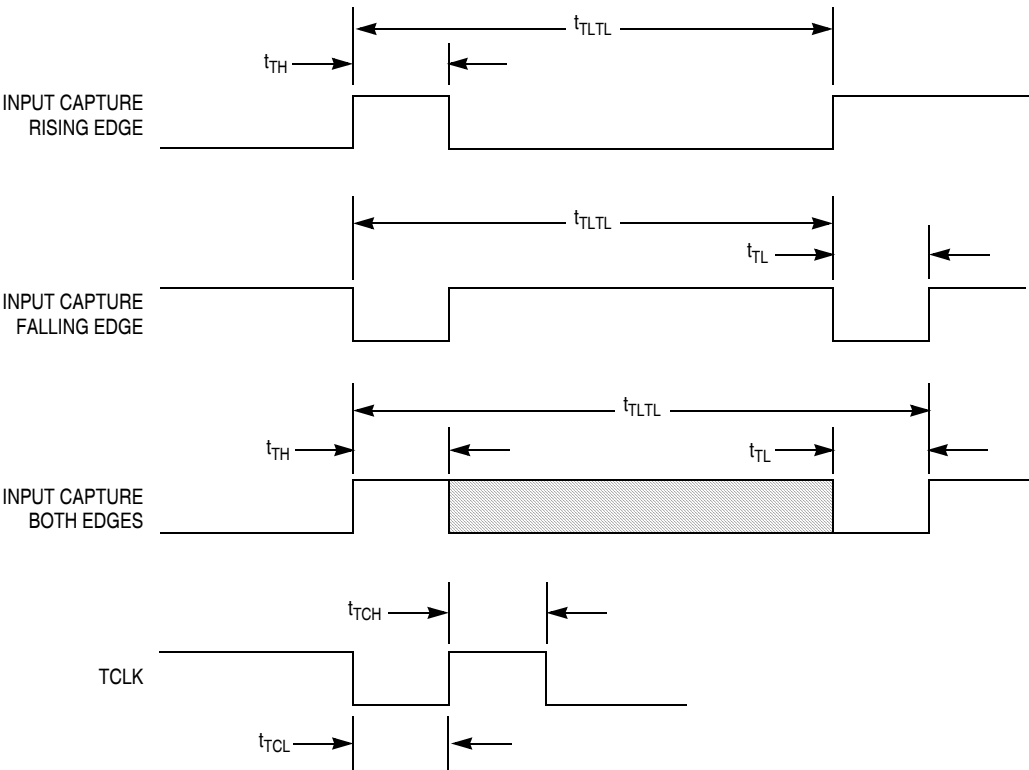


Figure 21-4. Timer Input Timing

# Chapter 22

## Ordering Information and Mechanical Specifications

### 22.1 Introduction

This section contains ordering numbers for the MC68HC908GZ60 and gives the dimensions for:

- 32-pin low-profile quad flat pack (case 873A)
- 48-pin low-profile quad flat pack (case 932-03)
- 64-pin quad flat pack (case 840B)

The following figures show the latest package drawings at the time of this publication. To make sure that you have the latest package specifications, contact your local Freescale Sales Office.

### 22.2 MC Order Numbers

Table 22-1. MC Order Numbers

MC Order Number	Operating Temperature Range	Package
MC908GZ60CFJ	−40°C to +85°C	32-pin low-profile quad flat package (LQFP)
MC908GZ60VFJ	−40°C to +105°C	
MC908GZ60MFJ	−40°C to +125°C	
MC908GZ60CFA	−40°C to +85°C	48-pin low-profile quad flat package (LQFP)
MC908GZ60VFA	−40°C to +105°C	
MC908GZ60MFA	−40°C to +125°C	
MC908GZ60CFU	−40°C to +85°C	64-pin quad flat package (QFP)
MC908GZ60VFU	−40°C to +105°C	
MC908GZ60MFU	−40°C to +125°C	

Temperature designators:  
C = −40°C to +85°C  
V = −40°C to +105°C  
M = −40°C to +125°C

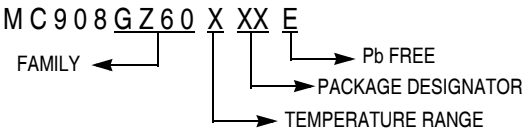


Figure 22-1. Device Numbering System

### 22.3 Package Dimensions

Refer to the following pages for detailed package dimensions.