

Welcome to [E·XFL.COM](https://www.e-fl.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	-
Peripherals	LVD, POR, PWM
Number of I/O	13
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 4x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	16-DIP (0.300", 7.62mm)
Supplier Device Package	16-PDIP
Purchase URL	<a href="https://www.e-fl.com/product-detail/nxp-semiconductors/mc908qy8cpe">https://www.e-fl.com/product-detail/nxp-semiconductors/mc908qy8cpe</a>

14.1	Introduction	139
14.2	RST and IRQ Pins Initialization	139
14.3	SIM Bus Clock Control and Generation	140
14.3.1	Bus Timing	141
14.3.2	Clock Start-Up from POR	141
14.3.3	Clocks in Stop Mode and Wait Mode	141
14.4	Reset and System Initialization	141
14.4.1	External Pin Reset	141
14.4.2	Active Resets from Internal Sources	142
14.4.2.1	Power-On Reset	143
14.4.2.2	Computer Operating Properly (COP) Reset	143
14.4.2.3	Illegal Opcode Reset	143
14.4.2.4	Illegal Address Reset	144
14.4.2.5	Low-Voltage Inhibit (LVI) Reset	144
14.5	SIM Counter	144
14.5.1	SIM Counter During Power-On Reset	144
14.5.2	SIM Counter During Stop Mode Recovery	144
14.5.3	SIM Counter and Reset States	144
14.6	Exception Control	145
14.6.1	Interrupts	145
14.6.1.1	Hardware Interrupts	145
14.6.1.2	SWI Instruction	148
14.6.2	Interrupt Status Registers	148
14.6.2.1	Interrupt Status Register 1	149
14.6.2.2	Interrupt Status Register 2	149
14.6.2.3	Interrupt Status Register 3	149
14.6.3	Reset	150
14.6.4	Break Interrupts	150
14.6.5	Status Flag Protection in Break Mode	150
14.7	Low-Power Modes	150
14.7.1	Wait Mode	150
14.7.2	Stop Mode	151
14.8	SIM Registers	152
14.8.1	SIM Reset Status Register	152
14.8.2	Break Flag Control Register	153

## Chapter 15

### Serial Peripheral Interface (SPI) Module

15.1	Introduction	155
15.2	Features	155
15.3	Functional Description	157
15.3.1	Master Mode	158
15.3.2	Slave Mode	158
15.3.3	Transmission Formats	159
15.3.3.1	Clock Phase and Polarity Controls	159
15.3.3.2	Transmission Format When CPHA = 0	159

# Chapter 1

## General Description

### 1.1 Introduction

The MC68HC908QB8 is a member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs). All MCUs in the family use the enhanced M68HC08 central processor unit (CPU08) and are available with a variety of modules, memory sizes and types, and package types.

**Table 1-1. Summary of Device Variations**

Device	FLASH/RAM Memory Size	ADC	16-Bit Timer Channels	ESCI	SPI	Pin Count
MC68HC908QB8	8K/256 bytes	10 channel, 10 bit	4	Yes	Yes	16 pins
MC68HC908QB4	4K/128 bytes	10 channel, 10 bit	4	Yes	Yes	16 pins
MC68HC908QY8	8K/256 bytes	4 channel, 10 bit	2	No	No	16 pins

### 1.2 Features

Features include:

- High-performance M68HC08 CPU core
- Fully upward-compatible object code with M68HC05 Family
- 5-V and 3-V operating voltages ( $V_{DD}$ )
- 8-MHz internal bus operation at 5 V, 4-MHz at 3 V
- Trimmable internal oscillator
  - Software selectable 1 MHz, 2 MHz, or 3.2 MHz internal bus operation
  - 8-bit trim capability
  - $\pm 25\%$  untrimmed
  - Trimmable to approximately  $0.4\%$ <sup>(1)</sup>
- Software selectable crystal oscillator range, 32–100 kHz, 1–8 MHz, and 8–32 MHz
- Software configurable input clock from either internal or external source
- Auto wakeup from STOP capability using dedicated internal 32-kHz RC or bus clock source
- On-chip in-application programmable FLASH memory
  - Internal program/erase voltage generation
  - Monitor ROM containing user callable program/erase routines
  - FLASH security<sup>(2)</sup>

1. See [18.11 Oscillator Characteristics](#) for internal oscillator specifications

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

## Memory

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$0014	ESCI Status Register 2 (SCS2) <a href="#">See page 129.</a>	Read:	0	0	0	0	0	0	BKF	RPF
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0015	ESCI Data Register (SCDR) <a href="#">See page 129.</a>	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by reset							
\$0016	ESCI Baud Rate Register (SCBR) <a href="#">See page 130.</a>	Read:	LINT	LINR	SCP1	SCP0	R	SCR2	SCR1	SCR0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0017	ESCI Prescaler Register (SCPSC) <a href="#">See page 131.</a>	Read:	PDS2	PDS1	PDS0	PSSB4	PSSB3	PSSB2	PSSB1	PSSB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0018	ESCI Arbiter Control Register (SCIACTL) <a href="#">See page 135.</a>	Read:	AM1	AHOST	AM0	ACLK	AFIN	ARUN	AROVFL	ARD8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0019	ESCI Arbiter Data Register (SCIA DAT) <a href="#">See page 136.</a>	Read:	ARD7	ARD6	ARD5	ARD4	ARD3	ARD2	ARD1	ARD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$001A	Keyboard Status and Control Register (KBSCR) <a href="#">See page 87.</a>	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
\$001B	Keyboard Interrupt Enable Register (KBIER) <a href="#">See page 88.</a>	Read:	0	AWUIE	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$001C	Keyboard Interrupt Polarity Register (KBIPR) <a href="#">See page 88.</a>	Read:	0	0	KBIP5	KBIP4	KBIP3	KBIP2	KBIP1	KBIP0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$001D	IRQ Status and Control Register (INTSCR) <a href="#">See page 81.</a>	Read:	0	0	0	0	IRQF	0	IMASK	MODE
		Write:						ACK		
		Reset:	0	0	0	0	0	0	0	0
\$001E	Configuration Register 2 (CONFIG2) <sup>(1)</sup> <a href="#">See page 57.</a>	Read:	IRQPUD	IRQEN	R	R	R	ESCIBDSRC	OSCENIN-STOP	RSTEN
		Write:								
		Reset:	0	0	0	0	0	0	0	0 <sup>(2)</sup>
1. One-time writable register after each reset. 2. RSTEN reset to 0 by a power-on reset (POR) only.										
\$001F	Configuration Register 1 (CONFIG1) <sup>(1)</sup> <a href="#">See page 58.</a>	Read:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVITRIP	SSREC	STOP	COPD
		Write:								
		Reset:	0	0	0	0	0 <sup>(2)</sup>	0	0	0
1. One-time writable register after each reset. 2. LVITRIP reset to 0 by a power-on reset (POR) only.										
\$0020	TIM Status and Control Register (TSC) <a href="#">See page 183.</a>	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST				
		Reset:	0	0	1	0	0	0	0	0
\$0021	TIM Counter Register High (TCNTH) <a href="#">See page 185.</a>	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
= Unimplemented      R = Reserved      U = Unaffected										

**Figure 2-2. Control, Status, and Data Registers (Sheet 2 of 5)**

## Chapter 3

# Analog-to-Digital Converter (ADC10) Module

### 3.1 Introduction

This section describes the 10-bit successive approximation analog-to-digital converter (ADC10).

The ADC10 module shares its pins with general-purpose input/output (I/O) port pins. See [Figure 3-1](#) for port location of these shared pins. The ADC10 on this MCU uses  $V_{DD}$  and  $V_{SS}$  as its supply and reference pins. This MCU uses BUSCLKX4 as its alternate clock source for the ADC. This MCU does not have a hardware conversion trigger.

### 3.2 Features

Features of the ADC10 module include:

- Linear successive approximation algorithm with 10-bit resolution
- Output formatted in 10- or 8-bit right-justified format
- Single or continuous conversion (automatic power-down in single conversion mode)
- Configurable sample time and conversion speed (to save power)
- Conversion complete flag and interrupt
- Input clock selectable from up to three sources
- Operation in wait and stop modes for lower noise operation
- Selectable asynchronous hardware conversion trigger

### 3.3 Functional Description

The ADC10 uses successive approximation to convert the input sample taken from ADVIN to a digital representation. The approximation is taken and then rounded to the nearest 10- or 8-bit value to provide greater accuracy and to provide a more robust mechanism for achieving the ideal code-transition voltage.

[Figure 3-2](#) shows a block diagram of the ADC10

For proper conversion, the voltage on ADVIN must fall between  $V_{REFH}$  and  $V_{REFL}$ . If ADVIN is equal to or exceeds  $V_{REFH}$ , the converter circuit converts the signal to \$3FF for a 10-bit representation or \$FF for a 8-bit representation. If ADVIN is equal to or less than  $V_{REFL}$ , the converter circuit converts it to \$000. Input voltages between  $V_{REFH}$  and  $V_{REFL}$  are straight-line linear conversions.

#### **NOTE**

*Input voltage must not exceed the analog supply voltages.*

### AWUIE — Auto Wakeup Interrupt Enable Bit

This read/write bit enables the auto wakeup interrupt input to latch interrupt requests. Reset clears AWUIE.

1 = Auto wakeup enabled as interrupt input

0 = Auto wakeup not enabled as interrupt input

#### NOTE

*KBIE5–KBIE0 bits are not used in conjunction with the auto wakeup feature. To see a description of these bits, see [9.8.2 Keyboard Interrupt Enable Register \(KBIER\)](#).*

## 4.6.4 Configuration Register 2

The configuration register 2 (CONFIG2), is used to allow the bus clock source to run in STOP. In this case, the clock, BUSCLKX2 will be used to drive the AWU request generator.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	IRQPUD	IRQEN	R	R	R	ESCI-BDSRC	OSCENINSTOP	RSTEN
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 4-5. Configuration Register 2 (CONFIG2)

### OSCENINSTOP — Oscillator Enable in Stop Mode Bit

OSCENINSTOP, when set, will allow the bus clock source (BUSCLKX2) to generate clocks for the AWU in stop mode. See [11.8.1 Oscillator Status and Control Register](#) for information on enabling the external clock sources.

1 = Oscillator enabled to operate during stop mode

0 = Oscillator disabled during stop mode

#### NOTE

*IRQPUD, IRQEN, ESCIBDSRC, and RSTEN bits are not used in conjunction with the auto wakeup feature. To see a description of these bits, see [Chapter 5 Configuration Register \(CONFIG\)](#).*

## 4.6.5 Configuration Register 1

The configuration register 1 (CONFIG1), is used to select the period for the AWU. The timeout will be based on the COPRS bit along with the clock source for the AWU.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	COPRS	LVISTOP	LVIRSTD	LVIPWRD	LVITRIP	SSREC	STOP	COPD
Write:								
Reset: POR:	0	0	0	0	U	0	0	0
	0	0	0	0	0	0	0	0

U = Unaffected

Figure 4-6. Configuration Register 1 (CONFIG1)

Table 7-1. Instruction Set Summary (Sheet 3 of 6)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
CLR <i>opr</i> CLRA CLR <sub>X</sub> CLR <sub>H</sub> CLR <i>opr</i> , <sub>X</sub> CLR <sub>X</sub> CLR <i>opr</i> ,SP	Clear	M ← \$00 A ← \$00 X ← \$00 H ← \$00 M ← \$00 M ← \$00 M ← \$00	0	–	–	0	1	–	DIR INH INH INH IX1 IX SP1	3F 4F 5F 8C 6F 7F 9E6F	dd  ff ff	3 1 1 1 3 2 4
CMP # <i>opr</i> CMP <i>opr</i> CMP <i>opr</i> CMP <i>opr</i> , <sub>X</sub> CMP <i>opr</i> , <sub>X</sub> CMP <sub>X</sub> CMP <i>opr</i> ,SP CMP <i>opr</i> ,SP	Compare A with M	(A) – (M)	†	–	–	†	†	†	IMM DIR EXT IX2 IX1 IX SP1 SP2	A1 B1 C1 D1 E1 F1 9EE1 9ED1	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
COM <i>opr</i> COMA COM <sub>X</sub> COM <i>opr</i> , <sub>X</sub> COM <sub>X</sub> COM <i>opr</i> ,SP	Complement (One's Complement)	M ← (M̄) = \$FF – (M) A ← (Ā) = \$FF – (M) X ← (X̄) = \$FF – (M) M ← (M̄) = \$FF – (M) M ← (M̄) = \$FF – (M) M ← (M̄) = \$FF – (M)	0	–	–	†	†	1	DIR INH INH IX1 IX SP1	33 43 53 63 73 9E63	dd  ff ff ff	4 1 1 4 3 5
CPHX # <i>opr</i> CPHX <i>opr</i>	Compare H:X with M	(H:X) – (M:M + 1)	†	–	–	†	†	†	IMM DIR	65 75	ii ii+1 dd	3 4
CPX # <i>opr</i> CPX <i>opr</i> CPX <i>opr</i> CPX <sub>X</sub> CPX <i>opr</i> , <sub>X</sub> CPX <i>opr</i> , <sub>X</sub> CPX <i>opr</i> ,SP CPX <i>opr</i> ,SP	Compare X with M	(X) – (M)	†	–	–	†	†	†	IMM DIR EXT IX2 IX1 IX SP1 SP2	A3 B3 C3 D3 E3 F3 9EE3 9ED3	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
DAA	Decimal Adjust A	(A) <sub>10</sub>	U	–	–	†	†	†	INH	72		2
DBNZ <i>opr</i> , <i>rel</i> DBNZ <sub>A</sub> <i>rel</i> DBNZ <sub>X</sub> <i>rel</i> DBNZ <i>opr</i> , <sub>X</sub> , <i>rel</i> DBNZ <sub>X</sub> , <i>rel</i> DBNZ <i>opr</i> ,SP, <i>rel</i>	Decrement and Branch if Not Zero	A ← (A) – 1 or M ← (M) – 1 or X ← (X) – 1 PC ← (PC) + 3 + <i>rel</i> ? (result) ≠ 0 PC ← (PC) + 2 + <i>rel</i> ? (result) ≠ 0 PC ← (PC) + 2 + <i>rel</i> ? (result) ≠ 0 PC ← (PC) + 3 + <i>rel</i> ? (result) ≠ 0 PC ← (PC) + 2 + <i>rel</i> ? (result) ≠ 0 PC ← (PC) + 4 + <i>rel</i> ? (result) ≠ 0	–	–	–	–	–	–	DIR INH INH IX1 IX SP1	3B 4B 5B 6B 7B 9E6B	dd rr rr rr ff rr rr ff rr	5 3 3 5 4 6
DEC <i>opr</i> DECA DEC <sub>X</sub> DEC <i>opr</i> , <sub>X</sub> DEC <sub>X</sub> DEC <i>opr</i> ,SP	Decrement	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1 M ← (M) – 1	†	–	–	†	†	–	DIR INH INH IX1 IX SP1	3A 4A 5A 6A 7A 9E6A	dd  ff ff	4 1 1 4 3 5
DIV	Divide	A ← (H:A)/(X) H ← Remainder	–	–	–	–	†	†	INH	52		7
EOR # <i>opr</i> EOR <i>opr</i> EOR <i>opr</i> EOR <i>opr</i> , <sub>X</sub> EOR <i>opr</i> , <sub>X</sub> EOR <sub>X</sub> EOR <i>opr</i> ,SP EOR <i>opr</i> ,SP	Exclusive OR M with A	A ← (A ⊕ M)	0	–	–	†	†	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A8 B8 C8 D8 E8 F8 9EE8 9ED8	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
INC <i>opr</i> INCA INC <sub>X</sub> INC <i>opr</i> , <sub>X</sub> INC <sub>X</sub> INC <i>opr</i> ,SP	Increment	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1 M ← (M) + 1	†	–	–	†	†	–	DIR INH INH IX1 IX SP1	3C 4C 5C 6C 7C 9E6C	dd  ff ff	4 1 1 4 3 5

## 9.7 I/O Signals

The KBI module can share its pins with the general-purpose I/O pins. See [Figure 9-1](#) for the port pins that are shared.

### 9.7.1 KBI Input Pins (KBIX:KBI0)

Each KBI pin is independently programmable as an external interrupt source. KBI pin polarity can be controlled independently. Each KBI pin when enabled will automatically configure the appropriate pullup/pulldown device based on polarity.

## 9.8 Registers

The following registers control and monitor operation of the KBI module:


- KBSCR (keyboard interrupt status and control register)
- KBIER (keyboard interrupt enable register)
- KBIPR (keyboard interrupt polarity register)

### 9.8.1 Keyboard Status and Control Register (KBSCR)

Features of the KBSCR:

- Flags keyboard interrupt requests
- Acknowledges keyboard interrupt requests
- Masks keyboard interrupt requests
- Controls keyboard interrupt triggering sensitivity

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
Write:						ACKK		
Reset:	0	0	0	0	0	0	0	0

 = Unimplemented

**Figure 9-3. Keyboard Status and Control Register (KBSCR)**

#### Bits 7–4 — Not used

#### KEYF — Keyboard Flag Bit

This read-only bit is set when a keyboard interrupt is pending.

1 = Keyboard interrupt pending

0 = No keyboard interrupt pending

#### ACKK — Keyboard Acknowledge Bit

Writing a 1 to this write-only bit clears the KBI request. ACKK always reads 0.

#### IMASKK — Keyboard Interrupt Mask Bit

Writing a 1 to this read/write bit prevents the output of the KBI latch from generating interrupt requests.

1 = Keyboard interrupt requests disabled

0 = Keyboard interrupt requests enabled

#### MODEK — Keyboard Triggering Sensitivity Bit

This read/write bit controls the triggering sensitivity of the keyboard interrupt pins.

1 = Keyboard interrupt requests on edge and level

0 = Keyboard interrupt requests on edge only



### 11.3.4 XTAL Oscillator

The XTAL oscillator circuit is designed for use with an external crystal or ceramic resonator to provide an accurate clock source. In this configuration, the OSC2 pin is dedicated to the external crystal circuit. The OSC2EN bit has no effect when this clock mode is selected.

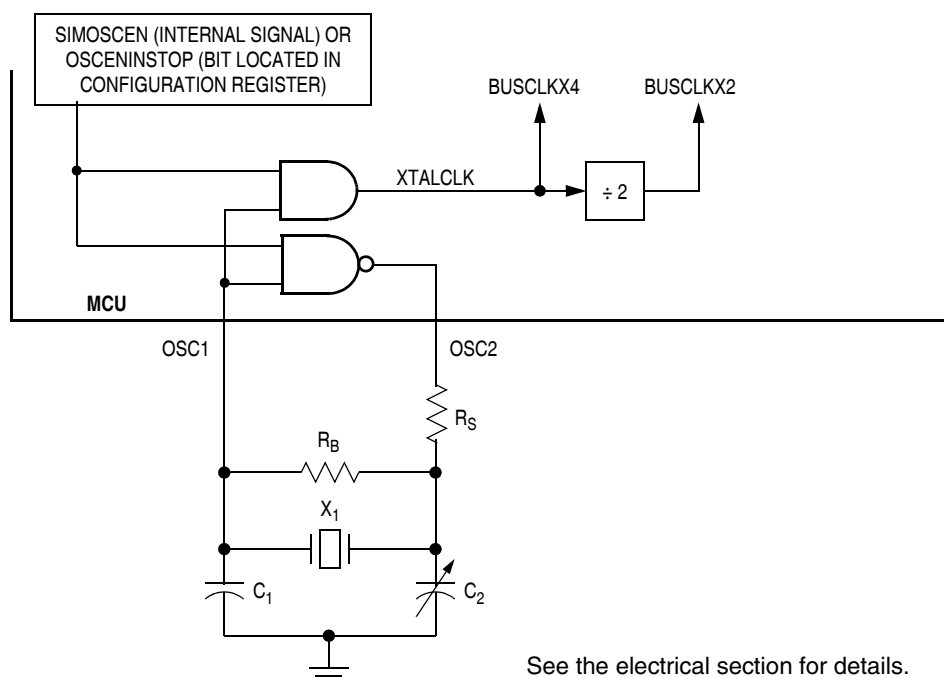
In its typical configuration, the XTAL oscillator is connected in a Pierce oscillator configuration, as shown in Figure 11-2. This figure shows only the logical representation of the internal components and may not represent actual circuitry.

The oscillator configuration uses five components:

- Crystal,  $X_1$
- Fixed capacitor,  $C_1$
- Tuning capacitor,  $C_2$  (can also be a fixed capacitor)
- Feedback resistor,  $R_B$
- Series resistor,  $R_S$  (optional)

#### NOTE

*The series resistor ( $R_S$ ) is included in the diagram to follow strict Pierce oscillator guidelines and may not be required for all ranges of operation, especially with high frequency crystals. Refer to the oscillator characteristics table in the Electricals section for more information.*



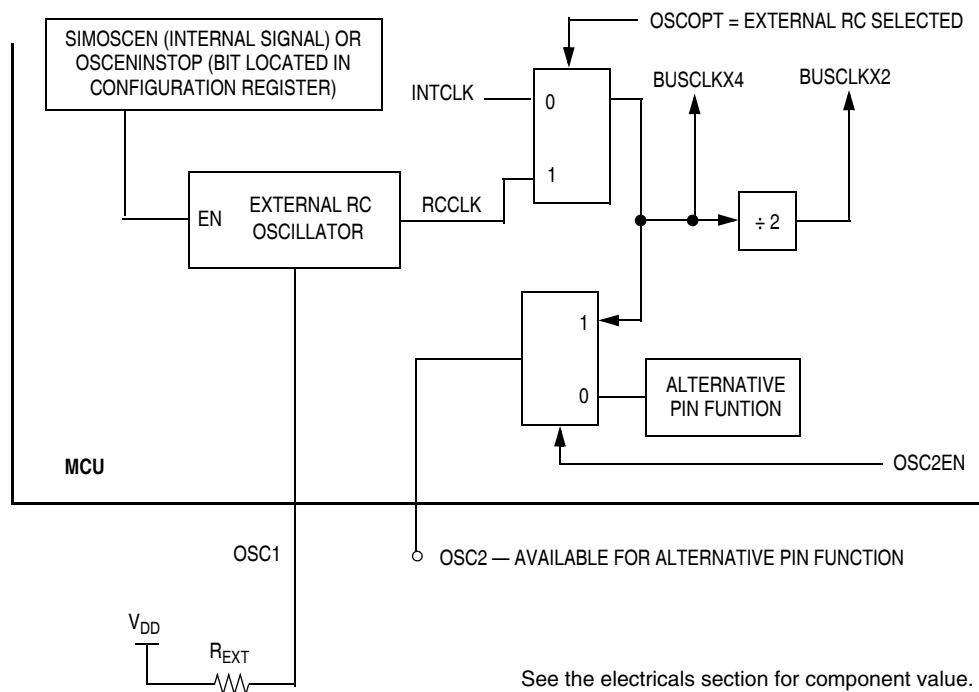
**Figure 11-2. XTAL Oscillator External Connections**

## 11.3.5 RC Oscillator

The RC oscillator circuit is designed for use with an external resistor ( $R_{EXT}$ ) to provide a clock source with a tolerance within 25% of the expected frequency. See [Figure 11-3](#).

The capacitor (C) for the RC oscillator is internal to the MCU. The  $R_{EXT}$  value must have a tolerance of 1% or less to minimize its effect on the frequency.

In this configuration, the OSC2 pin can be used as general-purpose input/output (I/O) port pins or other alternative pin function. The OSC2EN bit can be set to enable the OSC2 output function on the pin. Enabling the OSC2 output can affect the external RC oscillator frequency,  $f_{RCCLK}$ .



See the electricals section for component value.

**Figure 11-3. RC Oscillator External Connections**

## 11.4 Interrupts

There are no interrupts associated with the OSC module.

## 11.5 Low-Power Modes

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

### 11.5.1 Wait Mode

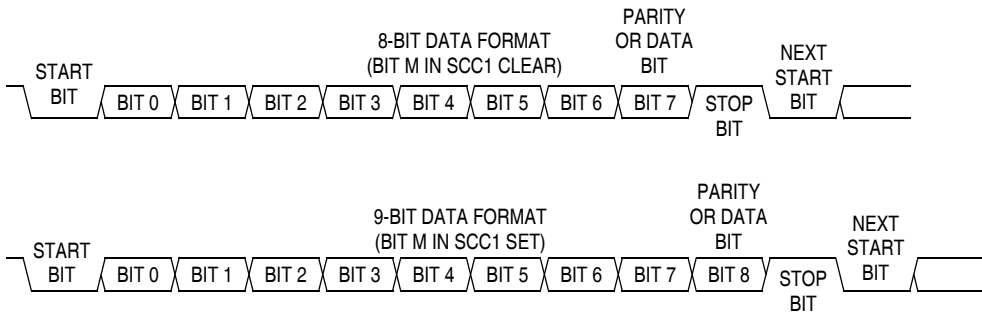
The OSC module remains active in wait mode.

### 11.5.2 Stop Mode

The OSC module can be configured to remain active in stop mode by setting OSCENINSTOP located in a configuration register.

## 13.3.1 Data Format

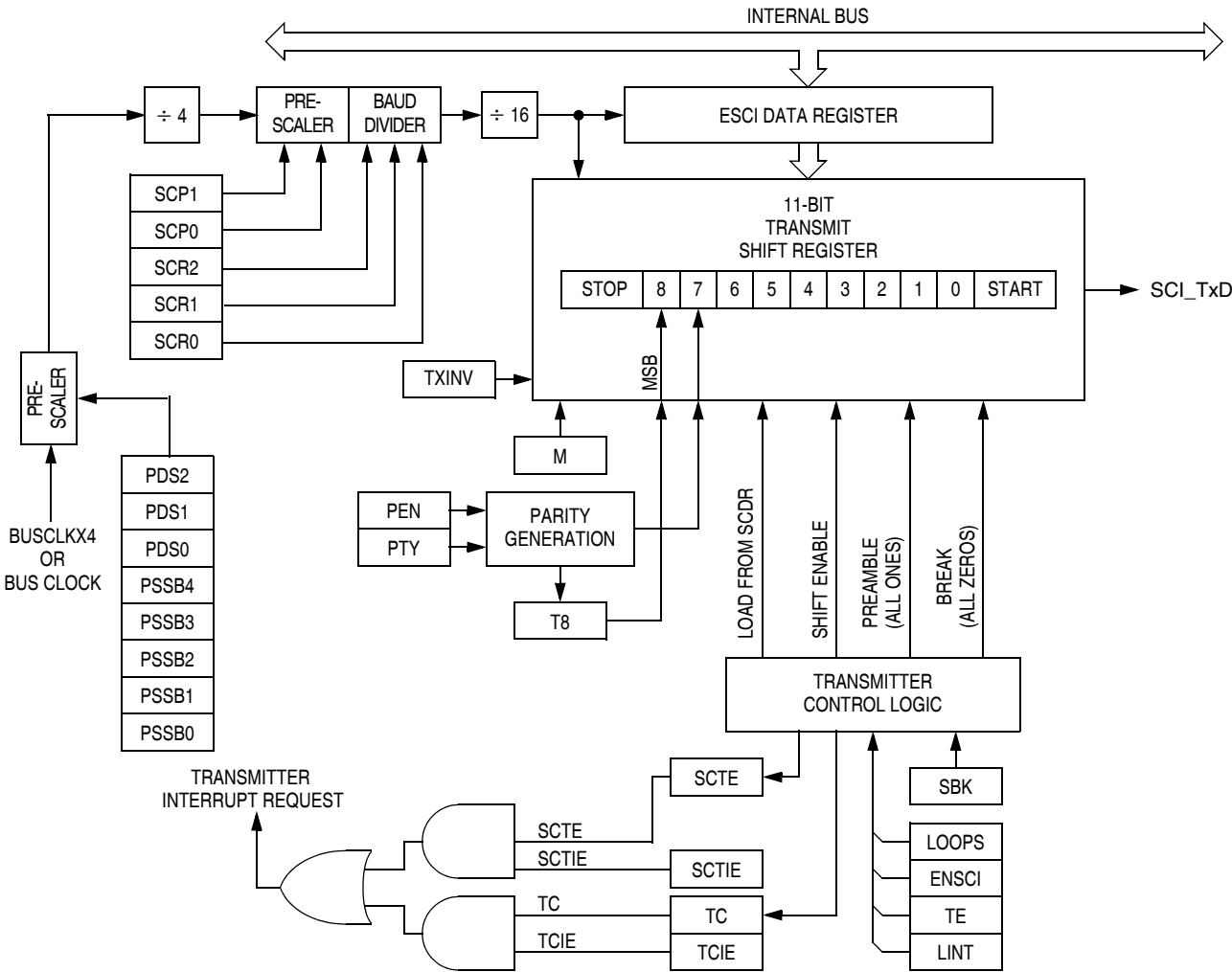
The SCI uses the standard mark/space non-return-to-zero (NRZ) format illustrated in [Figure 13-3](#).



**Figure 13-3. SCI Data Formats**

## 13.3.2 Transmitter

[Figure 13-4](#) shows the structure of the SCI transmitter.



**Figure 13-4. ESCI Transmitter**

**NOTE**

Writing to the RE bit is not allowed when the enable ESCI bit (ENSCI) is clear. ENSCI is in ESCI control register 1.

**RWU — Receiver Wakeup Bit**

This read/write bit puts the receiver in a standby state during which receiver interrupts are disabled. The WAKE bit in SCC1 determines whether an idle input or an address mark brings the receiver out of the standby state and clears the RWU bit.

- 1 = Standby state
- 0 = Normal operation

**SBK — Send Break Bit**

Setting and then clearing this read/write bit transmits a break character followed by a 1. The 1 after the break character guarantees recognition of a valid start bit. If SBK remains set, the transmitter continuously transmits break characters with no 1s between them.

- 1 = Transmit break characters
- 0 = No break characters being transmitted

**NOTE**

Do not toggle the SBK bit immediately after setting the SCTE bit. Toggling SBK before the preamble begins causes the ESCI to send a break character instead of a preamble.

### 13.8.3 ESCI Control Register 3

ESCI control register 3 (SCC3):

- Stores the ninth ESCI data bit received and the ninth ESCI data bit to be transmitted.
- Enables these interrupts:
  - Receiver overrun
  - Noise error
  - Framing error
  - Parity error

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R8	T8	R	R	ORIE	NEIE	FEIE	PEIE
Write:								
Reset:	U	0	0	0	0	0	0	0

  = Unimplemented
 R = Reserved
 U = Unaffected

**Figure 13-11. 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).

**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.

## PDS2–PDS0 — Prescaler Divisor Select Bits

These read/write bits select the prescaler divisor as shown in [Table 13-8](#).

### NOTE

*The setting of '000' will bypass not only this prescaler but also the prescaler divisor fine adjust (PDFA). It is not recommended to bypass the prescaler while ENSCI is set, because the switching is not glitch free.*

**Table 13-8. ESCI Prescaler Division Ratio**

PDS[2:1:0]	Prescaler Divisor (PD)
0 0 0	Bypass this prescaler
0 0 1	2
0 1 0	3
0 1 1	4
1 0 0	5
1 0 1	6
1 1 0	7
1 1 1	8

## PSSB4–PSSB0 — Clock Insertion Select Bits

These read/write bits select the number of clocks inserted in each 32 output cycle frame to achieve more timing resolution on the **average** prescaler frequency as shown in [Table 13-9](#).

**Table 13-9. ESCI Prescaler Divisor Fine Adjust**

PSSB[4:3:2:1:0]	Prescaler Divisor Fine Adjust (PDFA)
0 0 0 0 0	0/32 = 0
0 0 0 0 1	1/32 = 0.03125
0 0 0 1 0	2/32 = 0.0625
0 0 0 1 1	3/32 = 0.09375
0 0 1 0 0	4/32 = 0.125
0 0 1 0 1	5/32 = 0.15625
0 0 1 1 0	6/32 = 0.1875
0 0 1 1 1	7/32 = 0.21875
0 1 0 0 0	8/32 = 0.25
0 1 0 0 1	9/32 = 0.28125
0 1 0 1 0	10/32 = 0.3125
0 1 0 1 1	11/32 = 0.34375
0 1 1 0 0	12/32 = 0.375
0 1 1 0 1	13/32 = 0.40625
0 1 1 1 0	14/32 = 0.4375

Continued on next page

### 14.4.2 Active Resets from Internal Sources

The  $\overline{\text{RST}}$  pin is initially setup as a general-purpose input after a POR. Setting the RSTEN bit in the CONFIG2 register enables the pin for the reset function. This section assumes the RSTEN bit is set when describing activity on the  $\overline{\text{RST}}$  pin.

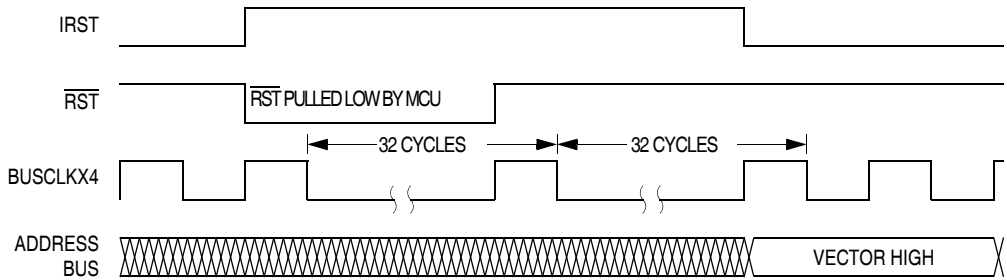
**NOTE**

*For POR and LVI resets, the SIM cycles through 4096 BUSCLKX4 cycles. The internal reset signal then follows the sequence from the falling edge of  $\overline{\text{RST}}$  shown in Figure 14-4.*

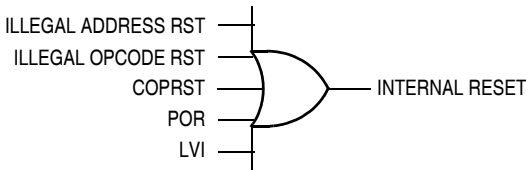
*The COP reset is asynchronous to the bus clock.*

The active reset feature allows the part to issue a reset to peripherals and other chips within a system built around the MCU.

All internal reset sources actively pull the  $\overline{\text{RST}}$  pin low for 32 BUSCLKX4 cycles to allow resetting of external peripherals. The internal reset signal IRST continues to be asserted for an additional 32 cycles (see Figure 14-4). An internal reset can be caused by an illegal address, illegal opcode, COP time out, LVI, or POR (see Figure 14-5).



**Figure 14-4. Internal Reset Timing**



**Figure 14-5. Sources of Internal Reset**

**Table 14-2. Reset Recovery Timing**

Reset Recovery Type	Actual Number of Cycles
POR/LVI	4163 (4096 + 64 + 3)
All others	67 (64 + 3)

controls the speed of the SPSCCK generated by an SPI configured as a master. Therefore, the frequency of the SPSCCK for an SPI configured as a slave can be any frequency less than or equal to the bus speed.

When the master SPI starts a transmission, the data in the slave shift register begins shifting out on the MISO pin. The slave can load its shift register with a new byte for the next transmission by writing to its transmit data register. The slave must write to its transmit data register at least one bus cycle before the master starts the next transmission. Otherwise, the byte already in the slave shift register shifts out on the MISO pin. Data written to the slave shift register during a transmission remains in a buffer until the end of the transmission.

When the clock phase bit (CPHA) is set, the first edge of SPSCCK starts a transmission. When CPHA is clear, the falling edge of  $\overline{SS}$  starts a transmission. See [15.3.3 Transmission Formats](#).

#### NOTE

*SPSCCK must be in the proper idle state before the slave is enabled to prevent SPSCCK from appearing as a clock edge.*

### 15.3.3 Transmission Formats

During an SPI transmission, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). A serial clock synchronizes shifting and sampling on the two serial data lines. A slave select line allows selection of an individual slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. On a master SPI device, the slave select line can optionally be used to indicate multiple-master bus contention.

#### 15.3.3.1 Clock Phase and Polarity Controls

Software can select any of four combinations of serial clock (SPSCCK) phase and polarity using two bits in the SPI control register (SPCR). The clock polarity is specified by the CPOL control bit, which selects an active high or low clock and has no significant effect on the transmission format.

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

#### NOTE

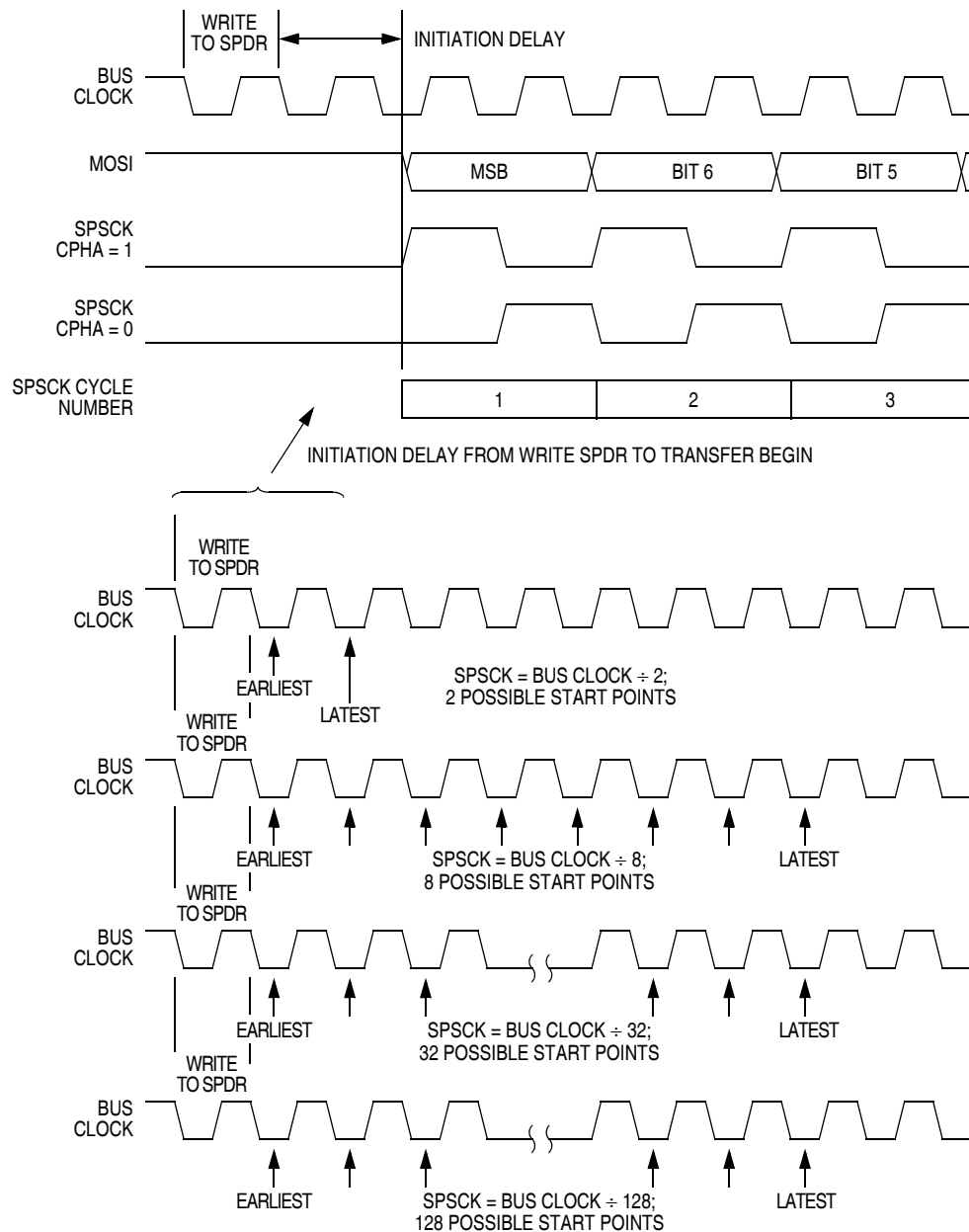
*Before writing to the CPOL bit or the CPHA bit, disable the SPI by clearing the SPI enable bit (SPE).*

#### 15.3.3.2 Transmission Format When CPHA = 0

[Figure 15-4](#) shows an SPI transmission in which CPHA = 0. The figure should not be used as a replacement for data sheet parametric information.

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. After 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.

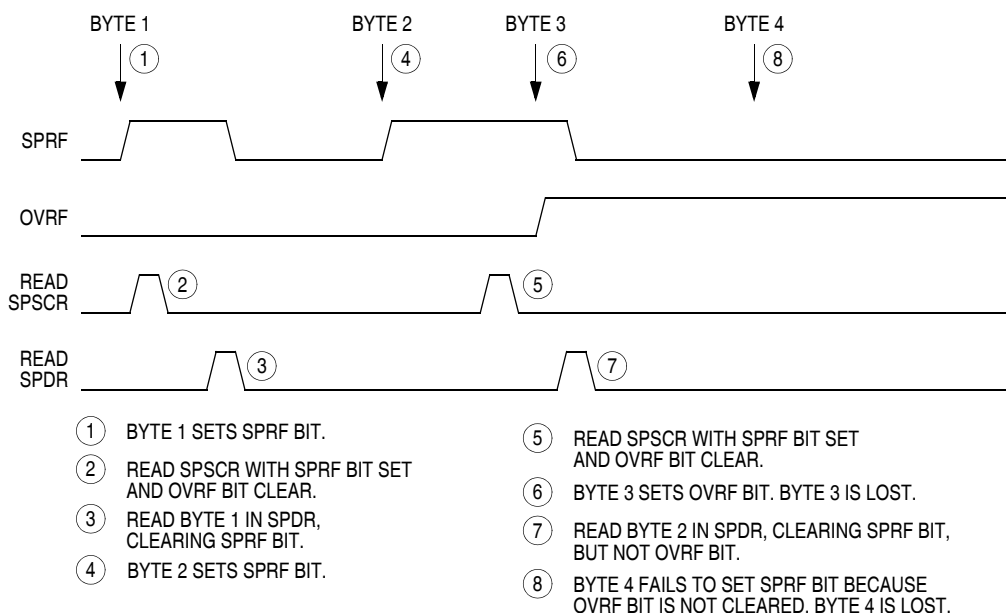
# Serial Peripheral Interface (SPI) Module



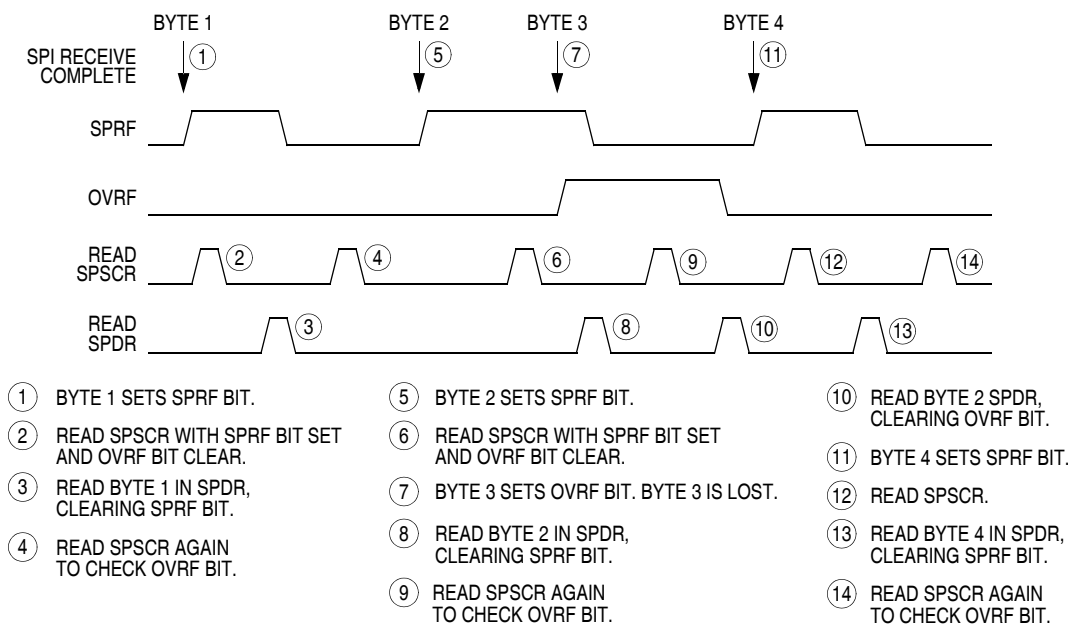
**Figure 15-7. Transmission Start Delay (Master)**



In this case, an overflow can be missed easily. Because no more SPRF interrupts can be generated until this OVRF is serviced, it is not obvious that bytes are being lost as more transmissions are completed. To prevent this, either enable the OVRF interrupt or do another read of the SPSCR following the read of the SPDR. This ensures that the OVRF was not set before the SPRF was cleared and that future transmissions can set the SPRF bit. Figure 15-10 illustrates this process. Generally, to avoid this second SPSCR read, enable OVRF by setting the ERRIE bit.



**Figure 15-9. Missed Read of Overflow Condition**



**Figure 15-10. Clearing SPRF When OVRF Interrupt Is Not Enabled**

## 15.7 I/O Signals

The SPI module can share its pins with the general-purpose I/O pins. See [Figure 15-1](#) for the port pins that are shared.

The SPI module has four I/O pins:

- MISO — Master input/slave output
- MOSI — Master output/slave input
- SPSCCK — Serial clock
- $\overline{SS}$  — Slave select

### 15.7.1 MISO (Master In/Slave Out)

MISO is one of the two SPI module pins that transmits serial data. In full duplex operation, the MISO pin of the master SPI module is connected to the MISO pin of the slave SPI module. The master SPI simultaneously receives data on its MISO pin and transmits data from its MOSI pin.

Slave output data on the MISO pin is enabled only when the SPI is configured as a slave. The SPI is configured as a slave when its SPMSTR bit is 0 and its  $\overline{SS}$  pin is low. To support a multiple-slave system, a high on the  $\overline{SS}$  pin puts the MISO pin in a high-impedance state.

When enabled, the SPI controls data direction of the MISO pin regardless of the state of the data direction register of the shared I/O port.

### 15.7.2 MOSI (Master Out/Slave In)

MOSI is one of the two SPI module pins that transmits serial data. In full-duplex operation, the MOSI pin of the master SPI module is connected to the MOSI pin of the slave SPI module. The master SPI simultaneously transmits data from its MOSI pin and receives data on its MISO pin.

When enabled, the SPI controls data direction of the MOSI pin regardless of the state of the data direction register of the shared I/O port.

### 15.7.3 SPSCCK (Serial Clock)

The serial clock synchronizes data transmission between master and slave devices. In a master MCU, the SPSCCK pin is the clock output. In a slave MCU, the SPSCCK pin is the clock input. In full-duplex operation, the master and slave MCUs exchange a byte of data in eight serial clock cycles.

When enabled, the SPI controls data direction of the SPSCCK pin regardless of the state of the data direction register of the shared I/O port.

### 15.7.4 $\overline{SS}$ (Slave Select)

The  $\overline{SS}$  pin has various functions depending on the current state of the SPI. For an SPI configured as a slave,  $\overline{SS}$  is used to select a slave. For CPHA = 0, the  $\overline{SS}$  is used to define the start of a transmission. (See [15.3.3 Transmission Formats](#).) Because it is used to indicate the start of a transmission,  $\overline{SS}$  must be toggled high and low between each byte transmitted for the CPHA = 0 format. However, it can remain low between transmissions for the CPHA = 1 format. See [Figure 15-12](#).

## 16.4 Interrupts

The following TIM sources can generate interrupt requests:

- TIM overflow flag (TOF) — The TOF bit is set when the counter reaches the modulo value programmed in the TIM counter modulo registers. The TIM overflow interrupt enable bit, TOIE, enables TIM overflow interrupt requests. TOF and TOIE are in the TSC register.
- TIM channel flags (CH3F:CH0F) — The CHxF bit is set when an input capture or output compare occurs on channel x. Channel x TIM interrupt requests are controlled by the channel x interrupt enable bit, CHxIE. Channel x TIM interrupt requests are enabled when CHxIE = 1. CHxF and CHxIE are in the TSCx register.

## 16.5 Low-Power Modes

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

### 16.5.1 Wait Mode

The TIM remains active after the execution of a WAIT instruction. In wait mode the TIM registers are not accessible by the CPU. Any enabled interrupt request from the TIM can bring the MCU out of wait mode.

If TIM functions are not required during wait mode, reduce power consumption by stopping the TIM before executing the WAIT instruction.

### 16.5.2 Stop Mode

The TIM module is inactive after the execution of a STOP instruction. The STOP instruction does not affect register conditions. TIM operation resumes after an external interrupt. If stop mode is exited by reset, the TIM is reset.

## 16.6 TIM During Break Interrupts

A break interrupt stops the counter.

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear status bits during the break state. See BFCR in the SIM section of this data sheet.

To allow software to clear status bits during a break interrupt, write a 1 to BCFE. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect status bits during the break state, write a 0 to BCFE. With BCFE cleared (its default state), software can read and write registers during the break state without affecting status bits. Some status bits have a two-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is cleared. After the break, doing the second step clears the status bit.

# Chapter 18

## Electrical Specifications

### 18.1 Introduction

This section contains electrical and timing specifications.

### 18.2 Absolute Maximum Ratings

Maximum ratings are the extreme limits to which the microcontroller unit (MCU) can be exposed without permanently damaging it.

#### NOTE

*This device is not guaranteed to operate properly at the maximum ratings. Refer to [18.5 5-V DC Electrical Characteristics](#) and [18.8 3-V DC Electrical Characteristics](#) for guaranteed operating conditions.*

Characteristic <sup>(1)</sup>	Symbol	Value	Unit
Supply voltage	$V_{DD}$	-0.3 to +6.0	V
Input voltage	$V_{IN}$	$V_{SS} - 0.3$ to $V_{DD} + 0.3$	V
Mode entry voltage, $\overline{IRQ}$ pin	$V_{TST}$	$V_{SS} - 0.3$ to +9.1	V
Maximum current per pin excluding PTA0–PTA5, $V_{DD}$ , and $V_{SS}$	I	±15	mA
Maximum current for pins PTA0–PTA5	$I_{PTA0} - I_{PTA5}$	±25	mA
Storage temperature	$T_{STG}$	-55 to +150	°C
Maximum current out of $V_{SS}$	$I_{MVSS}$	100	mA
Maximum current into $V_{DD}$	$I_{MVDD}$	100	mA

1. Voltages references to  $V_{SS}$ .

#### NOTE

*This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that  $V_{IN}$  and  $V_{OUT}$  be constrained to the range  $V_{SS} \leq (V_{IN} \text{ or } V_{OUT}) \leq V_{DD}$ . Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (for example, either  $V_{SS}$  or  $V_{DD}$ .)*

## 18.3 Functional Operating Range

Characteristic	Symbol	Value	Unit	Temperature Code
Operating temperature range	$T_A$ ( $T_L$ to $T_H$ )	-40 to +125 -40 to +105 -40 to +85	°C	M V C
Operating voltage range	$V_{DD}$	2.7 to 5.5	V	—

## 18.4 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal resistance 16-pin PDIP 16-pin SOIC 16-pin TSSOP	$\theta_{JA}$	76 90 133	°C/W
I/O pin power dissipation	$P_{I/O}$	User determined	W
Power dissipation <sup>(1)</sup>	$P_D$	$P_D = (I_{DD} \times V_{DD})$ $+ P_{I/O} = K / (T_J + 273^\circ\text{C})$	W
Constant <sup>(2)</sup>	K	$P_D \times (T_A + 273^\circ\text{C})$ $+ P_D^2 \times \theta_{JA}$	W/°C
Average junction temperature	$T_J$	$T_A + (P_D \times \theta_{JA})$	°C
Maximum junction temperature	$T_{JM}$	150	°C

1. Power dissipation is a function of temperature.

2. K constant unique to the device. K can be determined for a known  $T_A$  and measured  $P_D$ . With this value of K,  $P_D$  and  $T_J$  can be determined for any value of  $T_A$ .