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	HC05
Core Size	8-Bit
Speed	2.1MHz
Connectivity	SCI, SPI
Peripherals	POR, WDT
Number of I/O	24
Program Memory Size	8KB (8K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	304 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Through Hole
Package / Case	42-SDIP (0.600", 15.24mm)
Supplier Device Package	42-PDIP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc705c8abe

MC68HC705C8A
MC68HSC705C8A

Technical Data

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

http://www.freescale.com

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

3.3.3 Stack Pointer

The stack pointer (SP) shown in **Figure 3-4** is a 13-bit register that contains the address of the next free location on the stack. During a reset or after the reset stack pointer (RSP) instruction, the stack pointer initializes to $00FF. The address in the stack pointer decrements as data is pushed onto the stack and increments as data is pulled from the stack.

The seven most significant bits of the stack pointer are fixed permanently at 0000011, so the stack pointer produces addresses from $00C0 to $00FF. If subroutines and interrupts use more than 64 stack locations, the stack pointer wraps around to address $00FF and begins writing over the previously stored data. A subroutine uses two stack locations. An interrupt uses five locations.

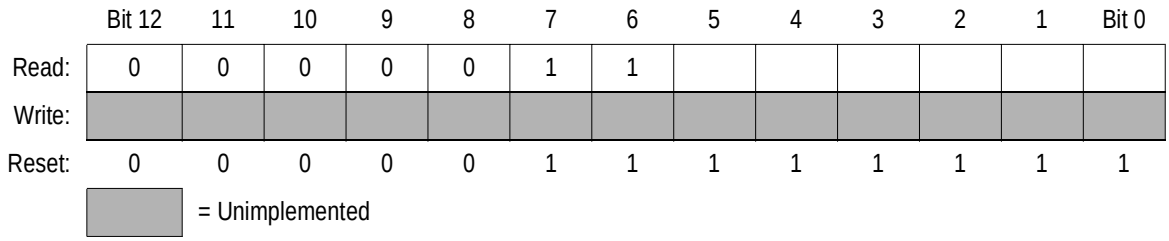

Figure 3-4. Stack Pointer (SP)

3.3.4 Program Counter

The program counter (PC) shown in **Figure 3-5** is a 13-bit register that contains the address of the next instruction or operand to be fetched.

Normally, the address in the program counter automatically increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, and interrupt operations load the program counter with an address other than that of the next sequential location.

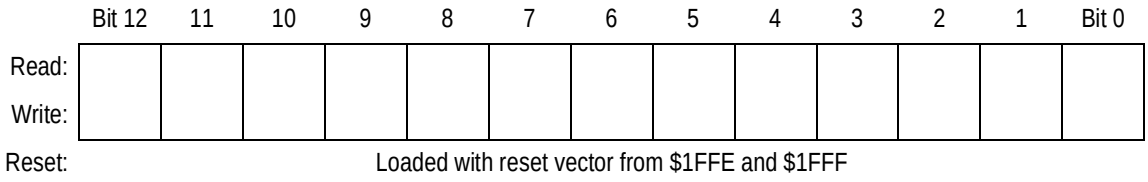

Figure 3-5. Program Counter (PC)

6.3.1 SCI During Stop Mode

When the MCU enters stop mode, the baud rate generator stops, halting all SCI activity. If the STOP instruction is executed during a transmitter transfer, that transfer is halted. If a low input to the $\overline{\text{IRQ}}$ pin is used to exit stop mode, the transfer resumes.

If the SCI receiver is receiving data and stop mode is entered, received data sampling stops because the baud rate generator stops, and all subsequent data is lost. Therefore, all SCI transfers should be in the idle state when the STOP instruction is executed.

6.3.2 SPI During Stop Mode

When the MCU enters stop mode, the baud rate generator stops, terminating all master mode SPI operations. If the STOP instruction is executed during an SPI transfer, that transfer halts until the MCU exits stop mode by a low signal on the $\overline{\text{IRQ}}$ pin. If reset is used to exit stop mode, the SPI control and status bits are cleared, and the SPI is disabled.

If the MCU is in slave mode when the STOP instruction is executed, the slave SPI continues to operate and can still accept data and clock information in addition to transmitting its own data back to a master device. At the end of a possible transmission with a slave SPI in stop mode, no flags are set until a low on the $\overline{\text{IRQ}}$ pin wakes up the MCU.

NOTE: *Although a slave SPI in stop mode can exchange data with a master SPI, the status bits of a slave SPI are inactive in stop mode.*

6.3.3 Programmable COP Watchdog in Stop Mode

The STOP instruction turns off the internal oscillator and suspends the computer operating properly (COP) watchdog counter. If the $\overline{\text{RESET}}$ pin brings the MCU out of stop mode, the reset function clears and disables the COP watchdog.

If the $\overline{\text{IRQ}}$ pin brings the MCU out of stop mode, the COP counter resumes counting from its suspended value after the 4064-t_{CYC} clock stabilization delay. See **Figure 6-2**.

NOTE: *If the clock monitor is enabled (CME = 1), the STOP instruction causes the clock monitor to time out and reset the MCU.*

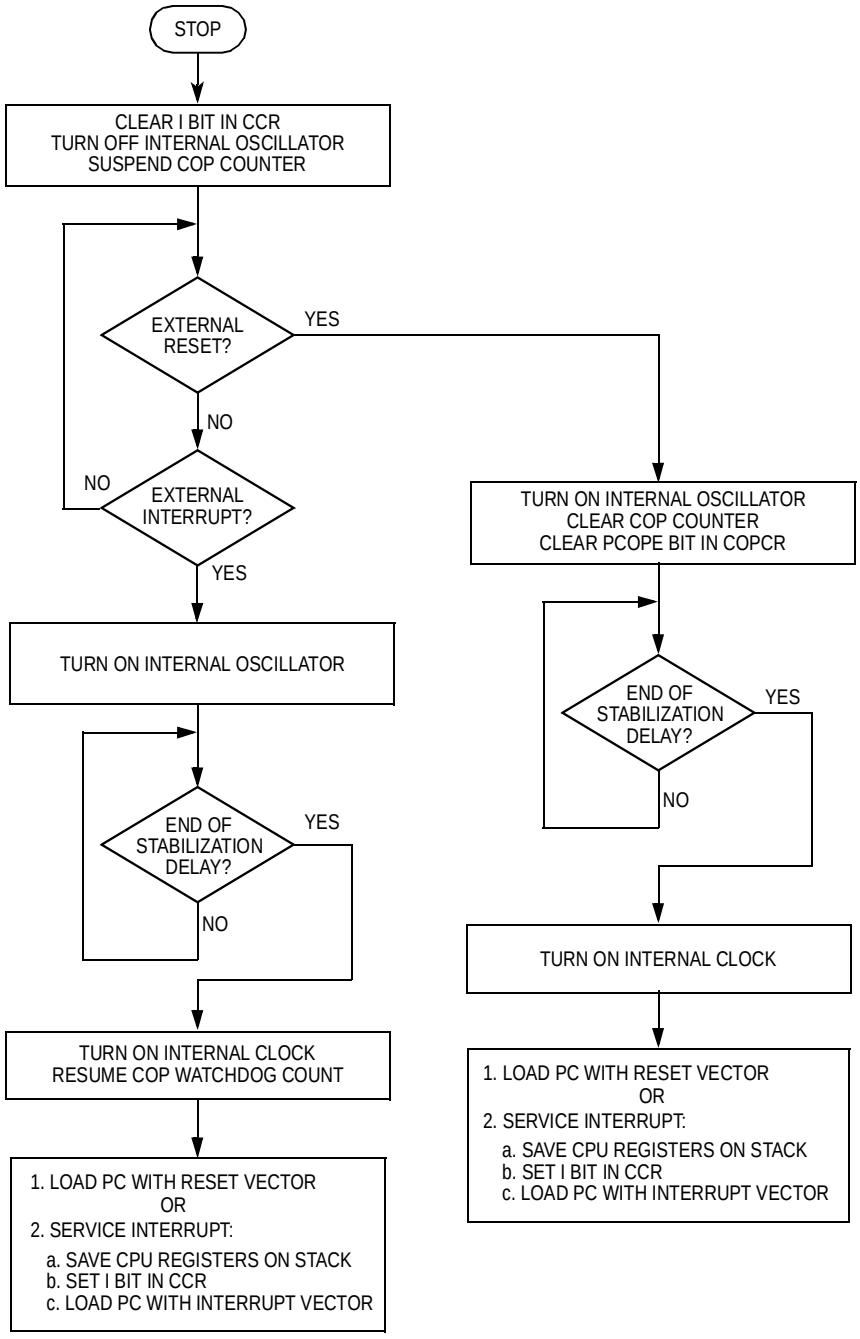

Figure 6-2. Programmable COP Watchdog in Stop Mode (PCOPE = 1) Flowchart

7.4 Port B

Port B is an 8-bit, general-purpose, bidirectional I/O port. Port B pins can also be configured to function as external interrupts. The port B pullup devices are enabled in mask option register 1 (MOR1). See **9.5.2 Mask Option Register 1** and **4.3.3 Port B Interrupts**.

7.4.1 Port B Data Register

The port B data register (PORTB) shown in **Figure 7-4** contains a data latch for each of the eight port B pins.

Address: $0001

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Reset:	Unaffected by reset							

Figure 7-4. Port B Data Register (PORTB)

PB7–PB0 — Port B Data Bits

These read/write bits are software programmable. Data direction of each bit is under the control of the corresponding bit in data direction register B. Reset has no effect on port B data.

7.4.3 Port B Logic

Figure 7-6 shows the port B I/O logic.

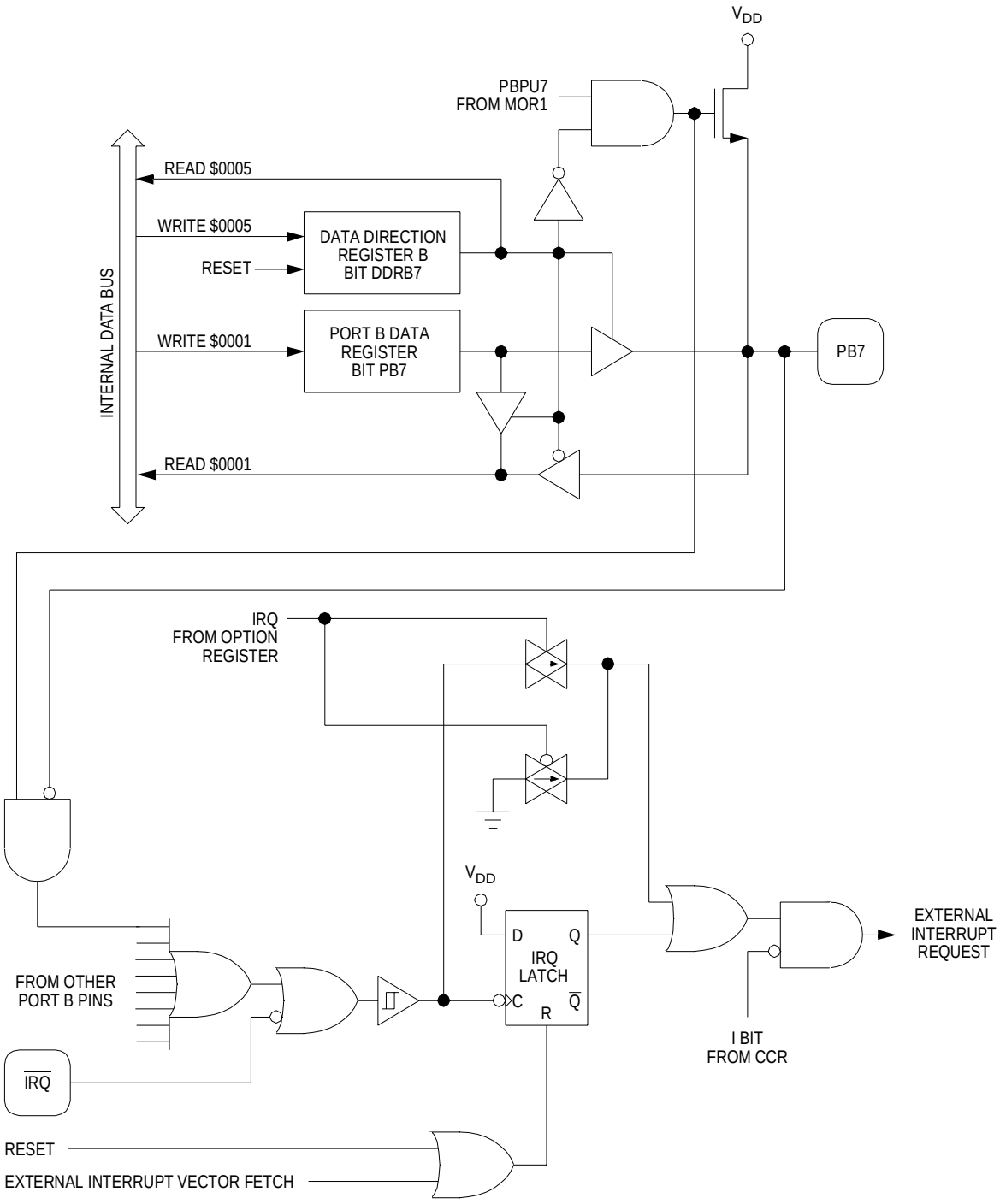

Figure 7-6. Port B I/O Logic

7.5 Port C

Port C is an 8-bit, general-purpose, bidirectional I/O port. PC7 has a high current sink and source capability.

7.5.1 Port C Data Register

The port C data register (PORTC) shown in **Figure 7-7** contains a data latch for each of the eight port C pins. When a port C pin is programmed to be an output, the state of its data register bit determines the state of the output pin. When a port C pin is programmed to be an input, reading the port C data register returns the logic state of the pin.

Address: $0002

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Reset:	Unaffected by reset							

Figure 7-7. Port C Data Register (PORTC)

PC7–PC0 — Port C Data Bits

These read/write bits are software programmable. Data direction of each bit is under the control of the corresponding bit in data direction register C. PC7 has a high current sink and source capability. Reset has no effect on port C data.

7.5.3 Port C Logic

Figure 7-9 shows port C I/O logic.

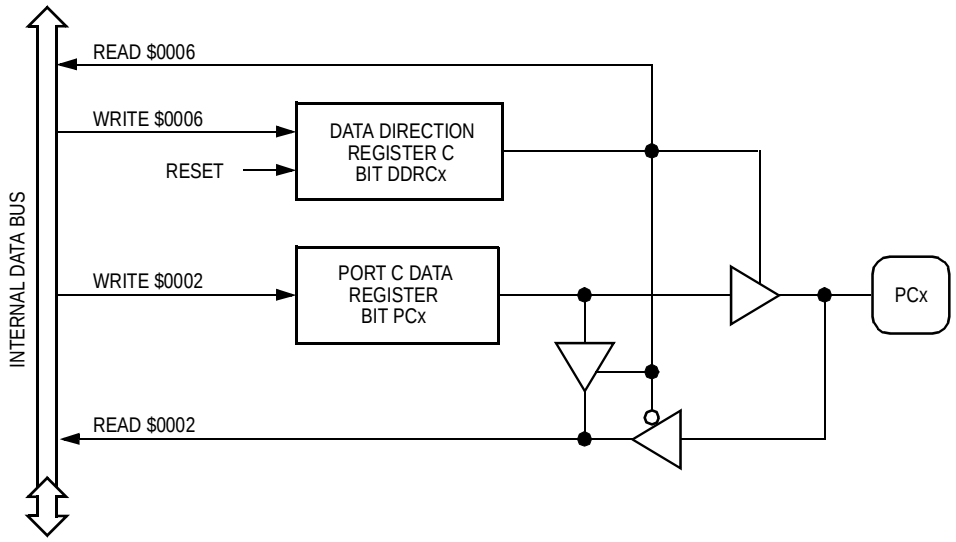

Figure 7-9. Port C I/O Logic

When a port C pin is programmed as an output, reading the port bit reads the value of the data latch and not the voltage on the pin. When a port C pin is programmed as an input, reading the port bit reads the voltage level on the pin. The data latch can always be written, regardless of the state of its DDRC bit. Table 7-3 summarizes the operation of the port C pins.

Table 7-3. Port C Pin Functions

DDRC Bit	I/O Pin Mode	Accesses to DDRC	Accesses to PORTC	
		Read/Write	Read	Write
0	Input, Hi-Z(1)	DDRC7–DDRC0	Pin	PC7–PC0(2)
1	Output	DDRC7–DDRC0	PC7–PC0	PC7–PC0

1. Hi-Z = high impedance
2. Writing affects data register but does not affect input.

NOTE: *To avoid excessive current draw, tie all unused input pins to V_{DD} or V_{SS} or change I/O pins to outputs by writing to DDRC in user code as early as possible.*

ICIE — Input Capture Interrupt Enable Bit

This read/write bit enables interrupts caused by an active signal on the TCAP pin. Reset clears the ICIE bit.

1 = Input capture interrupts enabled
0 = Input capture interrupts disabled

OCIE — Output Compare Interrupt Enable Bit

This read/write bit enables interrupts caused by an active signal on the TCMP pin. Reset clears the OCIE bit.

1 = Output compare interrupts enabled
0 = Output compare interrupts disabled

TOIE — Timer Overflow Interrupt Enable Bit

This read/write bit enables interrupts caused by a timer overflow. Reset clears the TOIE bit.

1 = Timer overflow interrupts enabled
0 = Timer overflow interrupts disabled

IEDG — Input Edge Bit

The state of this read/write bit determines whether a positive or negative transition on the TCAP pin triggers a transfer of the contents of the timer register to the input capture registers. Reset has no effect on the IEDG bit.

1 = Positive edge (low-to-high transition) triggers input capture
0 = Negative edge (high-to-low transition) triggers input capture

OLVL — Output Level Bit

The state of this read/write bit determines whether a logic 1 or a logic 0 appears on the TCMP pin when a successful output compare occurs. Reset clears the OLVL bit.

1 = TCMP goes high on output compare
0 = TCMP goes low on output compare

Bits 4–2 — Not used; these bits always read 0

Section 9. EPROM/OTPROM (PROM)

9.1 Contents

9.2 Introduction

This section describes erasable, programmable read-only memory/one-time programmable read-only memory (EPROM/OTPROM (PROM)) programming.

9.6 EPROM Erasing

The erased state of an EPROM or OTPROM byte is $00. EPROM devices can be erased by exposure to a high intensity ultraviolet (UV) light with a wave length of 2537 Å. The recommended erasure dosage (UV intensity on a given surface area x exposure time) is 15 Ws/cm^2. UV lamps should be used without short-wave filters, and the EPROM device should be positioned about one inch from the UV source.

OTPROM devices are shipped in an erased state. Once programmed, they cannot be erased. Electrical erasing procedures cannot be performed on either EPROM or OTPROM devices.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000D	Baud Rate Register (Baud) See page 136.	Read:			SCP1	SCP0		SCR2	SCR1	SCR0
		Write:								
		Reset:	U	U	0	0	U	U	U	U
$000E	SCI Control Register 1 (SCCR1) See page 130.	Read:	R8	T8		M	WAKE			
		Write:								
		Reset:	U	U		U	U			
$000F	SCI Control Register 2 (SCCR2) See page 131.	Read:	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0010	SCI Status Register (SCSR) See page 133.	Read:	TDRE	TC	RDRF	IDLE	OR	NF	FE	
		Write:								
		Reset:	1	1	0	0	0	0	0	U
$0011	SCI Data Register (SCDR) See page 129.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Unaffected by reset							

= Unimplemented U = Unaffected

Figure 10-3. SCI Transmitter I/O Register Summary

Writing a logic 1 to the TE bit in SCI control register 2 (SCCR2) and then writing data to the SCDR begins the transmission. At the start of a transmission, transmitter control logic automatically loads the transmit shift register with a preamble of logic 1s. After the preamble shifts out, the control logic transfers the SCDR data into the shift register. A logic 0 start bit automatically goes into the least significant bit (LSB) position of the shift register, and a logic 1 stop bit goes into the most significant bit (MSB) position.

When the data in the SCDR transfers to the transmit shift register, the transmit data register empty (TDRE) flag in the SCI status register (SCSR) becomes set. The TDRE flag indicates that the SCDR can accept new data from the internal data bus.

When the shift register is not transmitting a character, the PD1/TDO pin goes to the idle condition, logic 1. If software clears the TE bit during the idle condition, and while TDRE is set, the transmitter relinquishes control of the PD1/TDO pin.

- Break Characters — Writing a logic 1 to the SBK bit in SCCR2 loads the shift register with a break character. A break character contains all logic 0s and has no start and stop bits. Break character length depends on the M bit in SCCR1. As long as SBK is at logic 1, transmitter logic continuously loads break characters into the shift register. After software clears the SBK bit, the shift register finishes transmitting the last break character and then transmits at least one logic 1. The automatic logic 1 at the end of a break character is to guarantee the recognition of the start bit of the next character.
- Idle Characters — An idle character contains all logic 1s and has no start or stop bits. Idle character length depends on the M bit in SCCR1. The preamble is a synchronizing idle character that begins every transmission.

  Clearing the TE bit during a transmission relinquishes the PD1/TDO pin after the last character to be transmitted is shifted out. The last character may already be in the shift register, or waiting in the SCDR, or it may be a break character generated by writing to the SBK bit. Toggling TE from logic 0 to logic 1 while the last character is in transmission generates an idle character (a preamble) that allows the receiver to maintain control of the PD1/TDO pin.
- Transmitter Interrupts — These sources can generate SCI transmitter interrupt requests:
  - Transmit Data Register Empty (TDRE) — The TDRE bit in the SCSR indicates that the SCDR has transferred a character to the transmit shift register. TDRE is a source of SCI interrupt requests. The transmission complete interrupt enable bit (TCIE) in SCCR2 is the local mask for TDRE interrupts.
  - Transmission Complete (TC) — The TC bit in the SCSR indicates that both the transmit shift register and the SCDR are empty and that no break or idle character has been generated. TC is a source of SCI interrupt requests. The transmission complete interrupt enable bit (TCIE) in SCCR2 is the local mask for TC interrupts.

10.6.2 SCI Control Register 1

SCI control register 1 (SCCR1) shown in **Figure 10-6** has these functions:

- Stores ninth SCI data bit received and ninth SCI data bit transmitted
- Controls SCI character length
- Controls SCI wakeup method

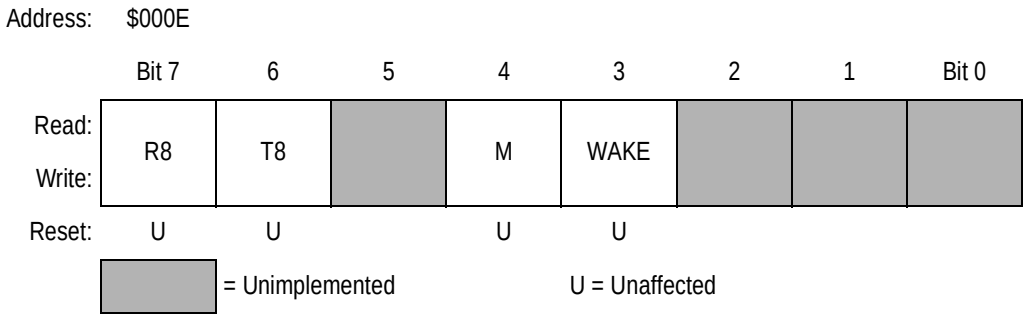

Figure 10-6. SCI Control Register 1 (SCCR1)

R8 — Bit 8 (Received)

When the SCI is receiving 9-bit characters, R8 is the ninth bit of the received character. R8 receives the ninth bit at the same time that the SCDR receives the other eight bits. Reset has no effect on the R8 bit.

T8 — Bit 8 (Transmitted)

When the SCI is transmitting 9-bit characters, T8 is the ninth bit of the transmitted character. T8 is loaded into the transmit shift register at the same time that SCDR is loaded into the transmit shift register. Reset has no effect on the T8 bit.

M — Character Length Bit

This read/write bit determines whether SCI characters are eight or nine bits long. The ninth bit can be used as an extra stop bit, as a receiver wakeup signal, or as a mark or space parity bit. Reset has no effect on the M bit.

1 = 9-bit SCI characters
0 = 8-bit SCI characters

SPI — SPI Enable Bit

This read/write bit enables the SPI. Reset clears the SPE bit.

1 = SPI enabled
0 = SPI disabled

MSTR — Master Bit

This read/write bit selects master mode operation or slave mode operation. Reset clears the MSTR bit.

1 = Master mode
0 = Slave mode

CPOL — Clock Polarity Bit

This read/write bit determines the logic state of the PD4/SCK pin between transmissions. To transmit data between SPIs, the SPIs must have identical CPOL bits. Reset has no effect on the CPOL bit.

1 = PD4/SCK pin at logic 1 between transmissions
0 = PD4/SCK pin at logic 0 between transmissions

CPHA — Clock Phase Bit

This read/write bit controls the timing relationship between the serial clock and SPI data. To transmit data between SPIs, the SPIs must have identical CPHA bits. When CPHA = 0, the PD5/$\overline{SS}$ pin of the slave SPI must be set to logic 1 between bytes. Reset has no effect on the CPHA bit.

1 = Edge following first active edge on PD4/SCK latches data
0 = First active edge on PD4/SCK latches data

SPR1 and SPR0 — SPI Clock Rate Bits

These read/write bits select the master mode serial clock rate, as shown in **Table 11-1**. The SPR1 and SPR0 bits of a slave SPI have no effect on the serial clock. Reset has no effect on SPR1 and SPR0.

Table 11-1. SPI Clock Rate Selection

SPR[1:0]	SPI Clock Rate
00	Internal Clock ÷ 2
01	Internal Clock ÷ 4
10	Internal Clock ÷ 16
11	Internal Clock ÷ 32

Section 12. Instruction Set

12.1 Contents

12.4.5 Control Instructions

These instructions act on CPU registers and control CPU operation during program execution.

Table 12-5. Control Instructions

Instruction	Mnemonic
Clear carry bit	CLC
Clear interrupt mask	CLI
No operation	NOP
Reset stack pointer	RSP
Return from interrupt	RTI
Return from subroutine	RTS
Set carry bit	SEC
Set interrupt mask	SEI
Stop oscillator and enable $\overline{\text{IRQ}}$ pin	STOP
Software interrupt	SWI
Transfer accumulator to index register	TAX
Transfer index register to accumulator	TXA
Stop CPU clock and enable interrupts	WAIT

13.9 5.0-Volt Control Timing

Characteristic[1]	Symbol	Min	Max	Unit
Frequency of operation Crystal option External clock option	f_{OSC}	— dc	4.2 4.2	MHz
Internal operating frequency Crystal ($f_{OSC} \div 2$) External clock ($f_{OSC} \div 2$)	f_{OP}	— dc	2.1 2.1	MHz
Cycle time (see **Figure 13-7**)	t_{CYC}	480	—	ns
Crystal oscillator startup time (see **Figure 13-7**)	t_{OXOV}	—	100	ms
Stop recovery startup time (crystal oscillator) (see **Figure 13-6**)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ pulse width (see **Figure 13-7**)	t_{RL}	8	—	t_{CYC}
Timer Resolution[2] Input capture pulse width (see **Figure 13-5**) Input capture pulse period (see **Figure 13-5**)	t_{RESL} t_{TH}, t_{TL} t_{TLTL}	4.0 125 [3]	— — —	t_{CYC} ns t_{CYC}
Interrupt pulse width low (edge-triggered) (see **Figure 4-2. External Interrupt Timing**)	t_{ILIH}	125	—	ns
Interrupt pulse period (see **Figure 4-2. External Interrupt Timing**)	t_{ILIL}	[4]	—	t_{CYC}
OSC1 pulse width	t_{OH}, t_{OL}	90	—	ns

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc; $T_A = T_L$ to T_H
2. Since a 2-bit prescaler in the timer must count four internal cycles (t_{CYC}), this is the limiting minimum factor in determining the timer resolution.
3. The minimum period, t_{TLTL}, should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{CYC}.
4. The minimum period, t_{ILIL}, should not be less than the number of cycle times it takes to execute the interrupt service routine plus 19 t_{CYC}.

A.7 5.0-Volt High-Speed SPI Timing

Diagram Number[1]	Characteristic[2]	Symbol	Min	Max	Unit
	Operating frequency Master Slave	$f_{OP(S)}$ $f_{OP(S)}$	dc dc	0.5 4.0	f_{OP} MHz
1	Cycle time Master Slave	$t_{CYC(M)}$ $t_{CYC(S)}$	2.0 250	— —	t_{CYC} ns
2	Enable lead time Master Slave	$t_{Lead(M)}$ $t_{Lead(S)}$	Note [3] 125	— —	ns
3	Enable lag time Master Slave	$t_{Lag(M)}$ $t_{Lag(S)}$	Note[2] 375	— —	ns
4	Clock (SCK) high time Master Slave	$t_{W(SCKH)M}$ $t_{W(SCKH)S}$	170 95	— —	ns
5	Clock (SCK) low time Master Slave	$t_{W(SCKL)M}$ $t_{W(SCKL)S}$	170 95	— —	ns
6	Data setup time (inputs) Master Slave	$t_{SU(M)}$ $t_{SU(S)}$	50 50	— —	ns
7	Data hold time (inputs) Master Slave	$t_{H(M)}$ $t_{H(S)}$	50 50	— —	ns
8	Access time[4] Slave	t_A	0	60	ns
9	Disable time[5] Slave	t_{DIS}	—	120	ns
10	Data valid time Master (before capture edge) Slave (after enable edge)[6]	$t_{V(M)}$ $t_{V(S)}$	0.25 —	— 120	$t_{CYC(M)}$ ns

Continued

D

E