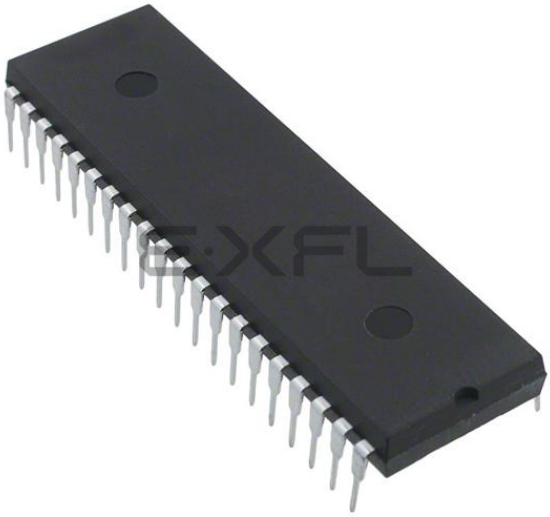


Welcome to E-XFL.COM

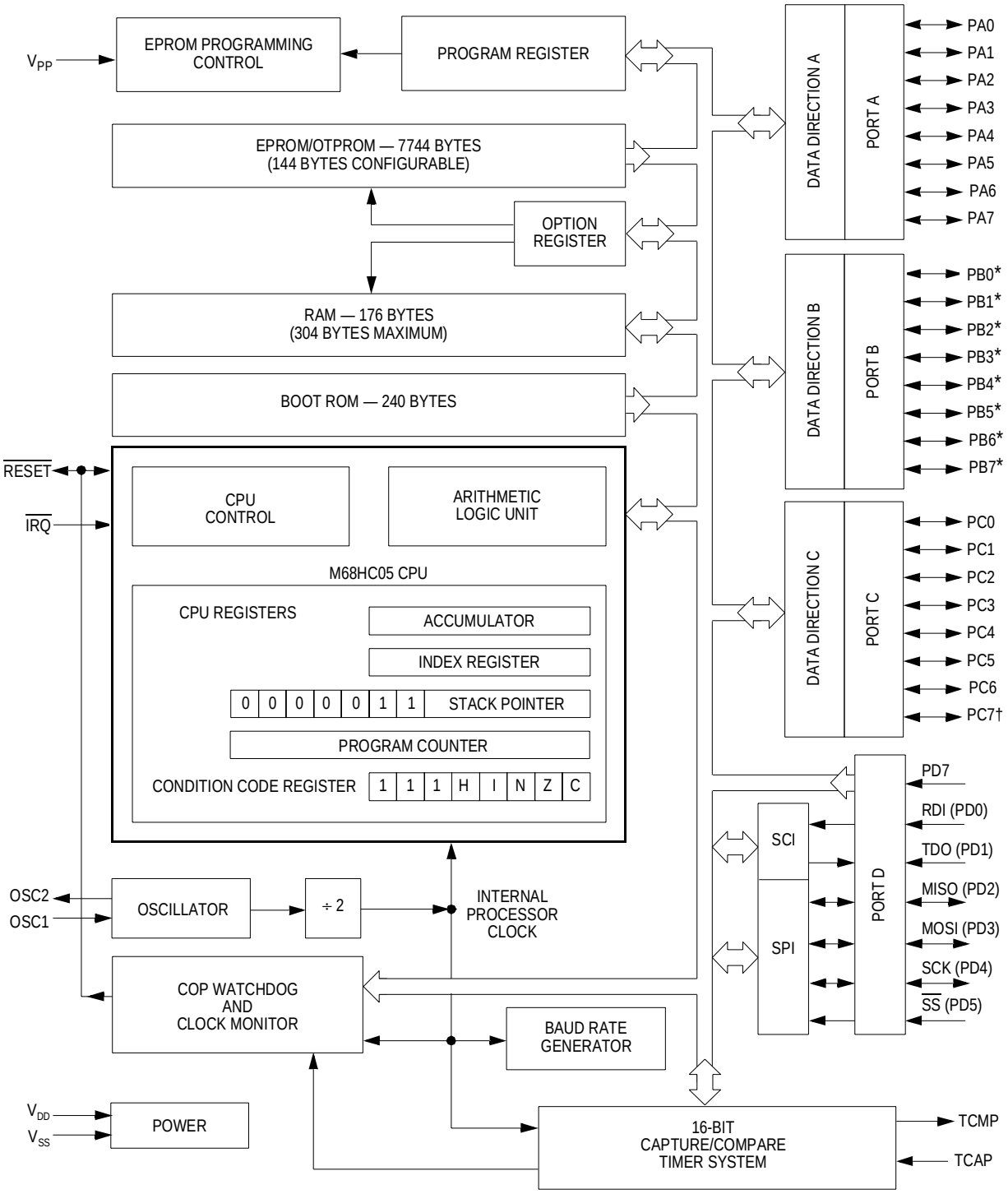
What is "Embedded - Microcontrollers"?

"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"

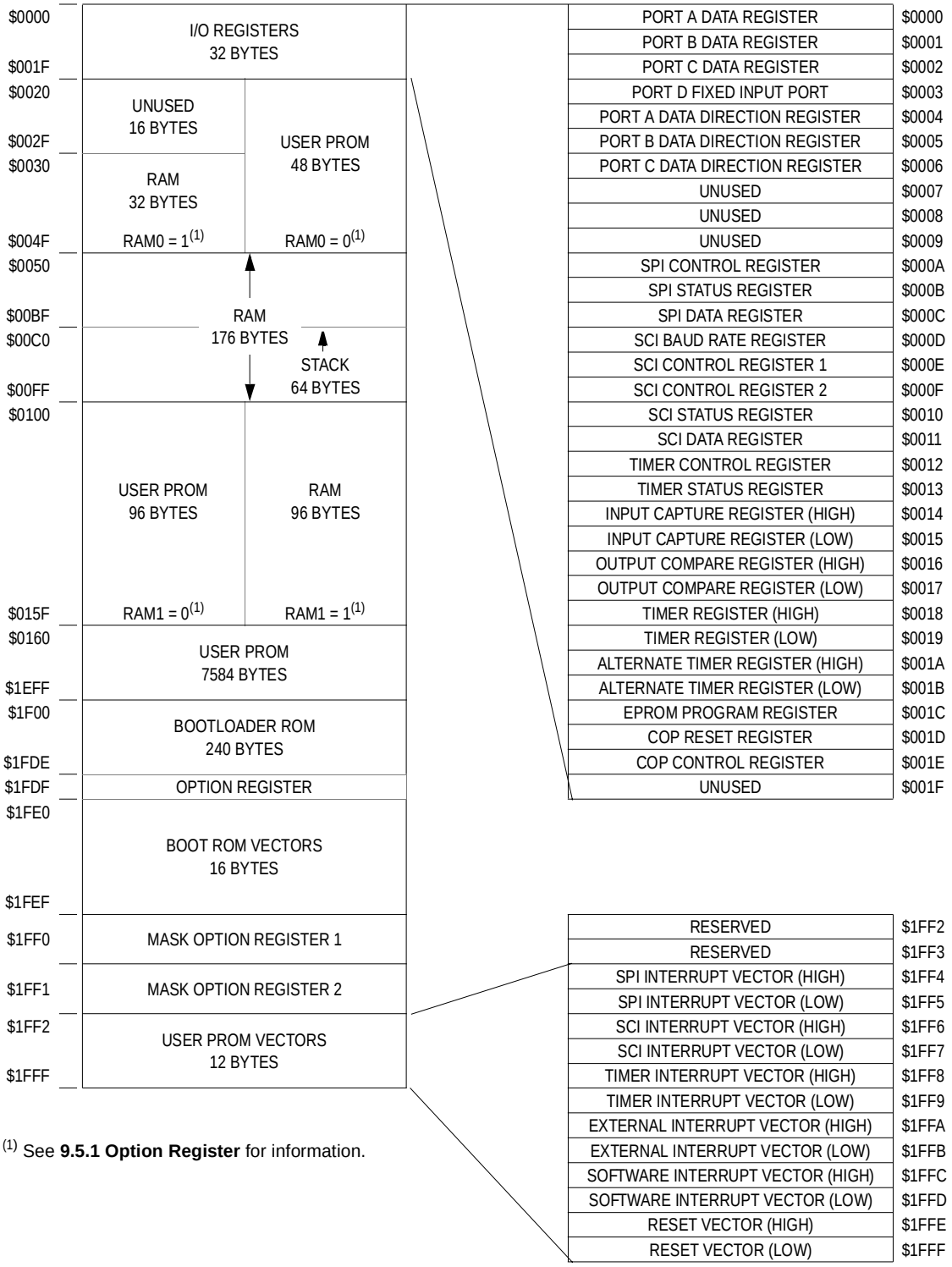
Details	
Product Status	Obsolete
Core Processor	HC05
Core Size	8-Bit
Speed	2.1MHz
Connectivity	SCI, SPI
Peripherals	POR, WDT
Number of I/O	24
Program Memory Size	8KB (8K x 8)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	304 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc705c8acpe

List of Tables



* Port B pins also function as external interrupts.
† PC7 has a high current sink and source capability.

Figure 1-2. MC68HC705C8A Block Diagram



(1) See **9.5.1 Option Register** for information.

Figure 2-1. Memory Map

Section 4. Interrupts

4.1 Contents

4.2 Introduction

This section describes how interrupts temporarily change the normal processing sequence.

4.3.2 External Interrupt ($\overline{IRQ}$)

An interrupt signal on the $\overline{IRQ}$ pin latches an external interrupt request. After completing the current instruction, the CPU tests these bits:

- IRQ latch
- I bit in the CCR

Setting the I bit in the CCR disables external interrupts.

If the IRQ latch is set and the I bit is clear, the CPU then begins the interrupt sequence. The CPU clears the IRQ latch while it fetches the interrupt vector, so that another external interrupt request can be latched during the interrupt service routine. As soon as the I bit is cleared during the return-from-interrupt (RTI) instruction, the CPU can recognize the new interrupt request. **Figure 4-1** shows the logic for external interrupts.

Figure 4-1 shows an external interrupt functional diagram. **Figure 4-2** shows an external interrupt timing diagram for the interrupt line. The timing diagram illustrates two treatments of the interrupt line to the processor.

1. Two single pulses on the interrupt line are spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service.

   Once a pulse occurs, the next pulse normally should not occur until an RTI occurs. This time (t_{ILIL}) is obtained by adding 19 instruction cycles to the total number of cycles needed to complete the service routine (not including the RTI instruction).

2. Many interrupt lines are “wire-ORed” to the $\overline{IRQ}$ line. If the interrupt line remains low after servicing an interrupt, then the CPU continues to recognize an interrupt.

NOTE: *The internal interrupt latch is cleared in the first part of the interrupt service routine. Therefore, a new external interrupt pulse could be latched and serviced as soon as the I bit is cleared.*

If the $\overline{IRQ}$ pin is not in use, connect it to the V_{DD} pin.

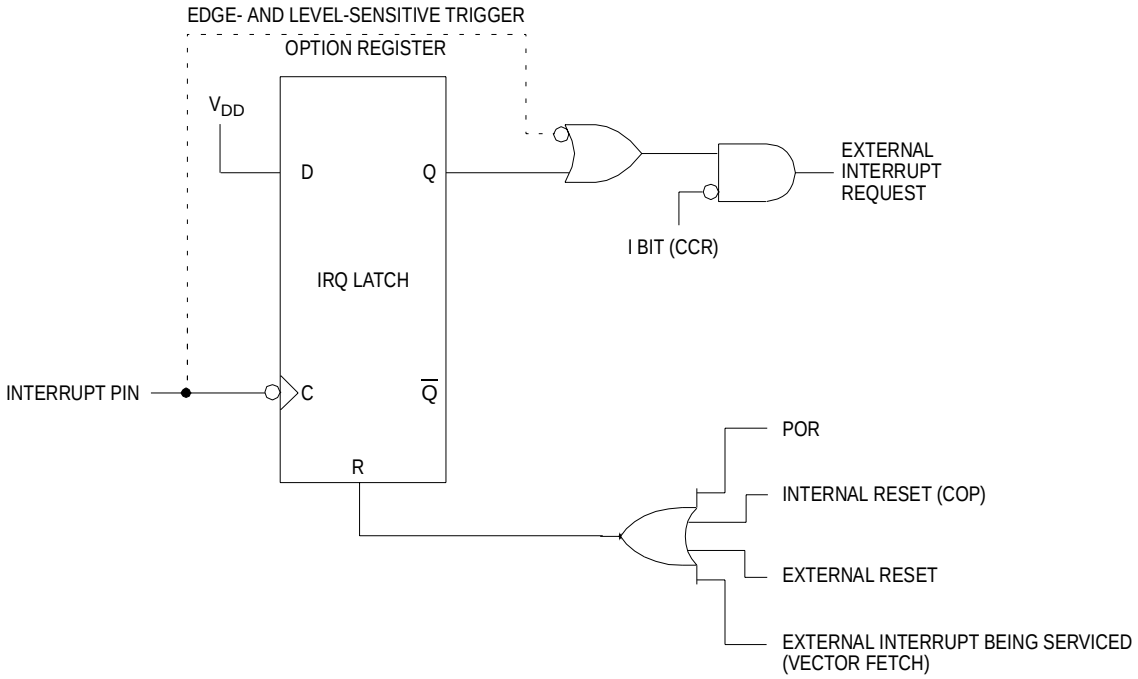
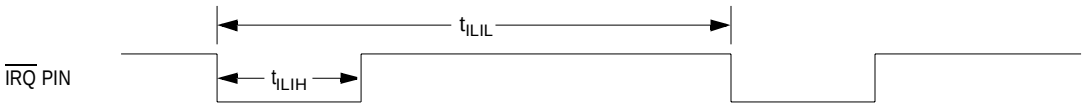
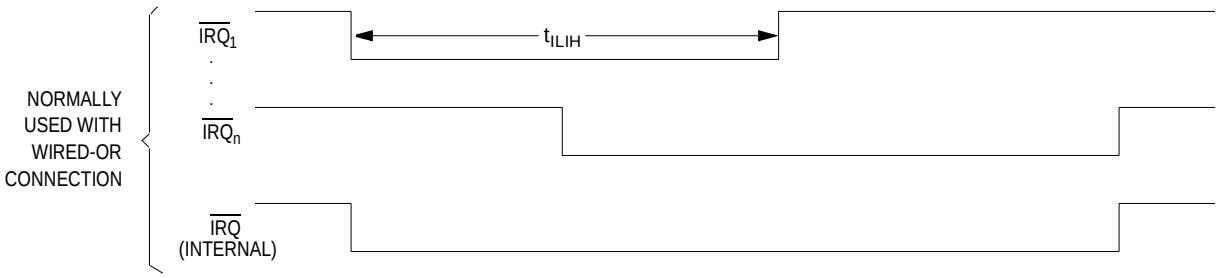


Figure 4-1. External Interrupt Internal Function Diagram



a. Edge-Sensitive Trigger Condition. The minimum pulse width (t_{ILIH}) is either 125 ns (f_{OP} = 2.1 MHz) or 250 ns (f_{OP} = 1 MHz). The period t_{ILIL} should not be less than the number of t_{CYC} cycles it takes to execute the interrupt service routine plus 19 t_{CYC} cycles.



b. Level-Sensitive Trigger Condition. If the interrupt line remains low after servicing an interrupt, then the CPU continues to recognize an interrupt.

Figure 4-2. External Interrupt Timing

Section 5. Resets

5.1 Contents

5.2 Introduction

This section describes how resets initialize the microcontroller unit (MCU).

5.3 Reset Sources

A reset immediately stops the operation of the instruction being executed, initializes certain control bits, and loads the program counter with a user-defined reset vector address. These conditions produce a reset:

- Power-on reset (POR) — Initial power-up
- External reset — A logic 0 applied to the $\overline{\text{RESET}}$ pin
- Internal programmable computer operating properly (COP) watchdog timer reset
- Internal non-programmable COP watchdog timer reset
- Internal clock monitor reset

Address: $001D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:								
Write:	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	U	U	U	U	U	U	U	U

= Unimplemented U = Unaffected

Figure 5-2. Programmable COP Reset Register (COPRST)

The programmable COP control register (COPCR) shown in **Figure 5-3** does these functions:

- Flags programmable COP watchdog resets
- Enables the clock monitor
- Enables the programmable COP watchdog
- Controls the timeout period of the programmable COP watchdog

Address: $001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	COPF	CME	PCOPE	CM1	CM0
Write:								
Reset:	0	0	0	U	0	0	0	0

= Unimplemented U = Unaffected

Figure 5-3. Programmable COP Control Register (COPCR)

COPF — COP Flag

This read-only bit is set when a timeout of the programmable COP watchdog occurs or when the clock monitor detects a slow or absent internal clock. Clear the COPF bit by reading the COP control register. Reset has no effect on the COPF bit.

1 = COP timeout or internal clock failure

0 = No COP timeout and no internal clock failure

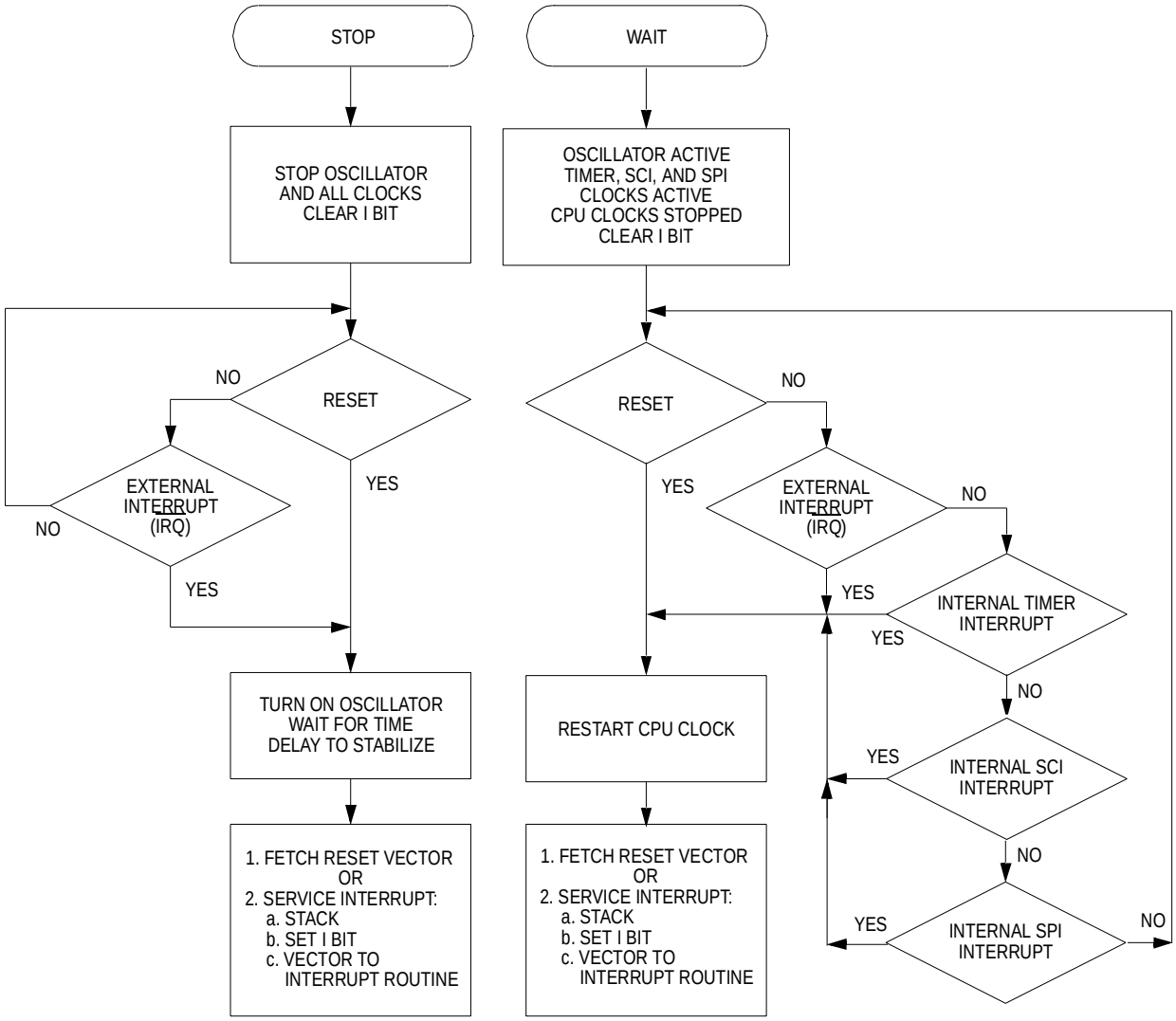


Figure 6-1. Stop/Wait Mode Function Flowchart

During stop mode, the I bit in the condition code register (CCR) is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output (I/O) lines remain unchanged. The processor can be brought out of stop mode only by an external interrupt or reset.

7.6 Port D

Port D is a 7-bit, special-purpose, input-only port that has no data register. Reading address $0003 returns the logic states of the port D pins.

Port D shares pins PD5–PD2 with the serial peripheral interface module (SPI). When the SPI is enabled, PD5–PD2 read as logic 0s. When the SPI is disabled, reading address $0003 returns the logic states of the PD5–PD2 pins.

Port D shares pins PD1 and PD0 with the SCI module. When the SCI is enabled, PD1 and PD0 read as logic 0s. When the SCI is disabled, reading address $0003 returns the logic states of the PD1 and PD0 pins.

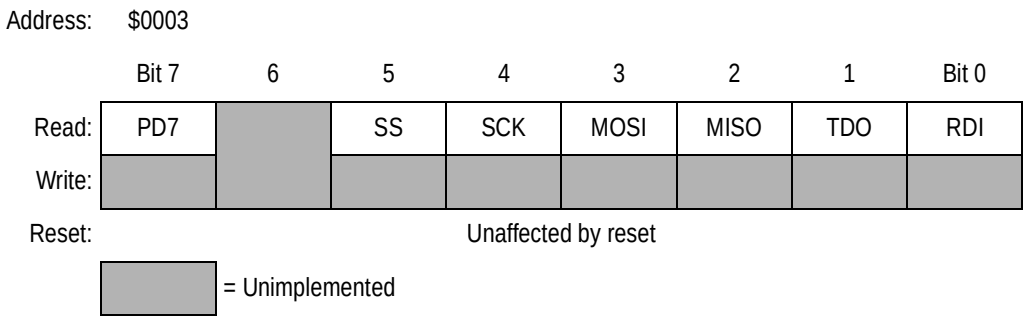


Figure 7-10. Port D Fixed Input Register (PORTD)

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.

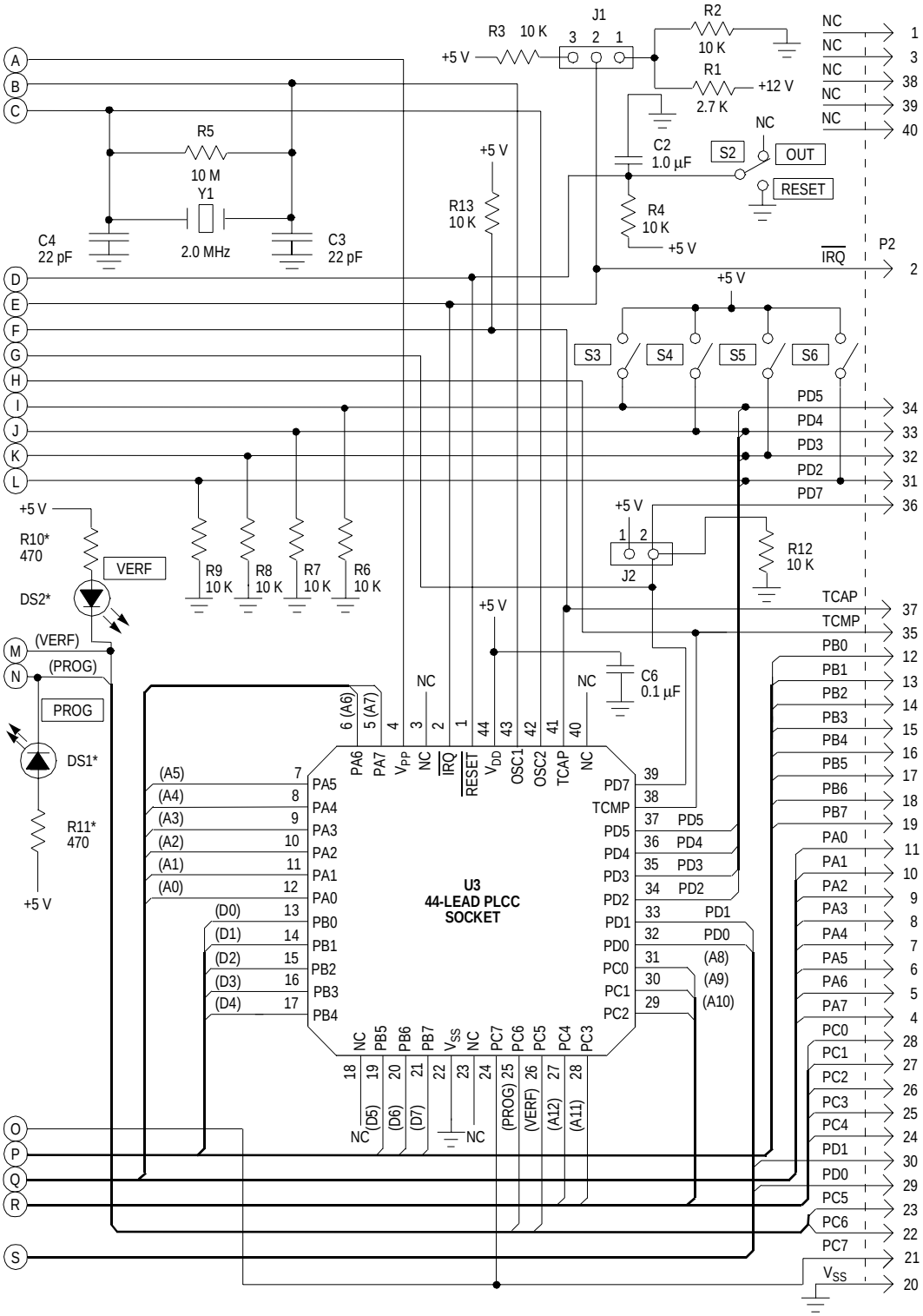


Figure 9-2. PROM Programming Circuit (Continued)

9.3.1 Program Register

The program register (PROG) shown in **Figure 9-3** is used for PROM programming.

Address: $001C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	0	0	0	0	0	LAT	0	PGM
Reset:	0	0	0	0	0	0	0	0

Figure 9-3. Program Register (PROG)

LAT — Latch Enable Bit

This bit is both readable and writable.

1 = Enables PROM data and address bus latches for programming on the next byte write cycle

0 = Latch disabled. PROM data and address buses are unlatched for normal CPU operations.

PGM — Program Bit

If LAT is cleared, PGM cannot be set.

1 = Enables V_{PP} power to the PROM for programming

0 = V_{PP} is disabled.

Bits 1 and 3–7 — Not used; always read 0

EPROM/OTPROM (PROM)

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

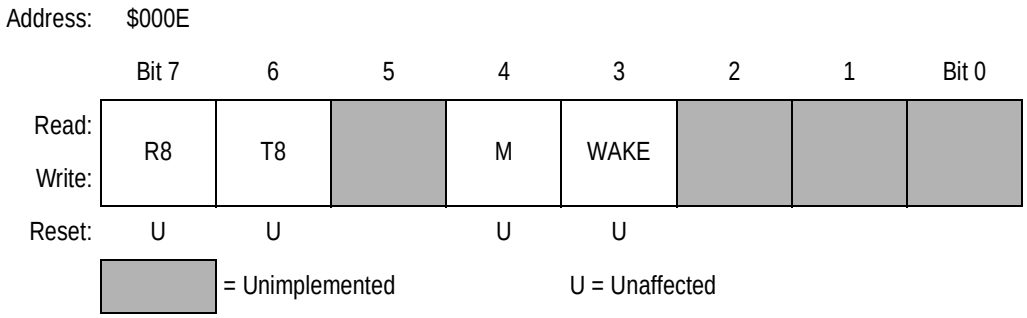


Figure 10-6. SCI Control Register 1 (SCCR1)

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

In a slave SPI, data enters the shift register under the control of the serial clock from the master SPI. After a byte enters the shift register of a slave SPI, it transfers to the SPDR. To prevent an overrun condition, slave software must then read the byte in the SPDR before another byte enters the shift register and is ready to transfer to the SPDR.

Figure 11-3 shows how a master SPI exchanges data with a slave SPI.

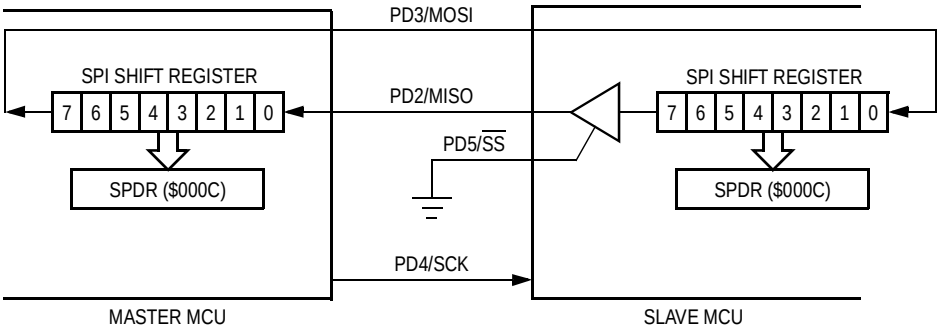


Figure 11-3. Master/Slave Connections

11.4.1 Pin Functions in Master Mode

Setting the MSTR bit in the SPI control register (SPCR) configures the SPI for operation in master mode. The master-mode functions of the SPI pins are:

- PD4/SCK (serial clock) — In master mode, the PD4/SCK pin is the synchronizing clock output.
- PD3/MOSI (master output, slave input) — In master mode, the PD3/MOSI pin is the serial output.
- PD2/MISO (master input, slave output) — In master mode, the PD2/MISO pin is configured as the serial input.
- PD5/$\overline{SS}$ (slave select) — In master mode, the PD5/$\overline{SS}$ pin protects against driver contention caused by the simultaneous operation of two SPIs in master mode. A logic 0 on the PD5/$\overline{SS}$ pin of a master SPI disables the SPI, clears the MSTR bit, and sets the mode-fault flag (MODF).

13.11 5.0-Volt Serial Peripheral Interface (SPI) Timing

Number[1]	Characteristic[2]	Symbol	Min	Max	Unit
	Operating frequency Master Slave	$f_{OP(M)}$ $f_{OP(S)}$	dc dc	0.5 2.1	f_{OP} MHz
1	Cycle time Master Slave	$t_{CYC(M)}$ $t_{CYC(S)}$	2.0 480	— —	t_{CYC} ns
2	Enable lead time Master Slave	$t_{Lead(M)}$ $t_{Lead(S)}$	(3) 240	— —	ns
3	Enable lag time Master Slave	$t_{Lag(M)}$ $t_{Lag(S)}$	(2) 720	— —	ns
4	Clock (SCK) high time Master Slave	$t_{W(SCKH)M}$ $t_{W(SCKH)S}$	340 190	— —	ns
5	Clock (SCK) low time Master Slave	$t_{W(SCKL)M}$ $t_{W(SCKL)S}$	340 190	— —	ns
6	Data setup time (inputs) Master Slave	$t_{SU(M)}$ $t_{SU(S)}$	100 100	— —	ns
7	Data hold time (inputs) Master Slave	$t_{H(M)}$ $t_{H(S)}$	100 100	— —	ns
8	Access time[4] Slave	t_A	0	120	ns
9	Disable time[5] Slave	t_{DIS}	—	240	ns
10	Data valid time Master (before capture edge) Slave (after enable edge)[6]	$t_{V(M)}$ $t_{V(S)}$	0.25 —	— 240	$t_{CYC(M)}$ ns

Continued

Diagram Number(1)	Characteristic(2)	Symbol	Min	Max	Unit
11	Data hold time (outputs) Master (after capture edge) Slave (after enable edge)	 $t_{HO(M)}$ $t_{HO(S)}$	 0.25 0	 — —	 $t_{CYC(M)}$ ns
12	Rise time(7) SPI outputs (SCK, MOSI, MISO) SPI inputs (SCK, MOSI, MISO, $\overline{SS}$)	 t_{RM} t_{RS}	 — —	 50 2.0	 ns µs
13	Fall time(8) SPI outputs (SCK, MOSI, MISO) SPI inputs (SCK, MOSI, MISO, $\overline{SS}$)	 t_{FM} t_{FS}	 — —	 50 2.0	 ns µs

1. Diagram numbers refer to dimensions in **Figure 13-8. SPI Master Timing** and **Figure 13-9. SPI Slave Timing**.
2. V_{DD} = 5 V ± 10%; V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted
3. Signal production depends on software.
4. Time to data active from high-impedance state
5. Hold time to high-impedance state
6. With 200 pF on all SPI pins.
7. 20% of V_{DD} to 70% of V_{DD}; C_L = 200 pF
8. 70% of V_{DD} to 20% of V_{DD}; C_L = 200 pF

Diagram Number(1)	Characteristic(2)	Symbol	Min	Max	Unit
11	Data hold time (outputs) Master (after capture edge) Slave (after enable edge)	 $t_{HO(M)}$ $t_{HO(S)}$	 0.25 0	 — —	 $t_{CYC(M)}$ ns
12	Rise time(7) SPI outputs (SCK, MOSI, MISO) SPI inputs (SCK, MOSI, MISO, $\overline{SS}$)	 t_{RM} t_{RS}	 — —	 100 2.0	 ns µs
13	Fall time(8) SPI outputs (SCK, MOSI, MISO) SPI inputs (SCK, MOSI, MISO, $\overline{SS}$)	 t_{FM} t_{FS}	 — —	 100 2.0	 ns µs

1. Diagram numbers refer to dimensions in **Figure 13-8. SPI Master Timing** and **Figure 13-9. SPI Slave Timing**.
2. V_{DD} = 3.3 V ± 10%; V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted
3. Signal production depends on software.
4. Time to data active from high-impedance state
5. Hold time to high-impedance state
6. With 200 pF on all SPI pins
7. 20% of V_{DD} to 70% of V_{DD}; C_L = 200 pF
8. 70% of V_{DD} to 20% of V_{DD}; C_L = 200 pF