MOTOROLA
intelligence everywhere™

digital dna™

MC68HC705C9A

Advance Information

M68HC05
Microcontrollers

MC68HC705C9A/D
Rev. 4, 2/2002

WWW.MOTOROLA.COM/SEMICONDUCTORS

MC68HC705C9A

Advance Information

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.motorola.com/semiconductors/

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.

Revision History

Date	Revision Level	Description	Page Number(s)
October, 2001	3.0	Format update to current publication standards	N/A
		Figure 12-10. SPI Slave Timing Diagram — Corrected labels for MISO and MOSI and subtitle for part b.	**145**
February, 2002	4.0	**Figure 8-3. Timer Status Register (TSR)** — Corrected address designator from $0012 to $0013.	**78**

List of Sections

List of Sections

Table of Contents

Section 1. General Description

Section 2. Memory

Section 3. Central Processor Unit (CPU)

Section 4. Interrupts

Section 5. Resets

Section 6. Low-Power Modes

Section 7. Input/Output (I/O) Ports

Section 8. Capture/Compare Timer

Section 9. Serial Communications Interface (SCI)

Section 10. Serial Peripheral Interface (SPI)

Section 11. Instruction Set

Section 12. Electrical Specifications

Section 13. Mechanical Specifications

Section 14. Ordering Information

Appendix A. EPROM Programming

Appendix B. M68HC05Cx Family Feature Comparisons

Table of Contents

List of Figures

List of Figures

Figure	Title	Page

List of Figures

List of Tables

List of Tables

Section 1. General Description

1.1 Contents

1.2 Introduction

The MC68HC705C9A HCMOS microcomputer is a member of the M68HC05 Family. The MC68HC705C9A is the EPROM version of the MC68HC05C9A and also can be configured as the EPROM version of

the MC68HC05C12A. The MC68HC705C9A memory map consists of 12,092 bytes of user EPROM and 176 bytes of RAM when it is configured as an MC68HC05C12A and 15,932 bytes of user EPROM and 352 bytes of RAM when configured as an MC68HC05C9A. The MC68HC705C9A includes a serial communications interface, a serial peripheral interface, and a 16-bit capture/compare timer.

1.3 Features

Features include:

- Programmable mask option register (MOR) for C9A/C12A configuration
- Programmable MOR for port B pullups and interrupts
- Popular M68HC05 central processor unit (CPU)
- 15,932 bytes of EPROM (12,092 bytes for C12A configuration)
- 352 bytes of RAM (176 for C12A configuration)
- Memory mapped input/output (I/O)
- 31 bidirectional I/O lines (24 I/O + 6 input only for C12A configuration) with high current sink and source on PC7
- Asynchronous serial communications interface (SCI)
- Synchronous serial peripheral interface (SPI)
- 16-bit capture/compare timer
- Computer operating properly (COP) watchdog timer and clock monitor
- Power-saving wait and stop modes
- On-chip crystal oscillator connections
- Single 3.0 volts to 5.5 volts power supply requirement
- EPROM contents security[1] feature

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

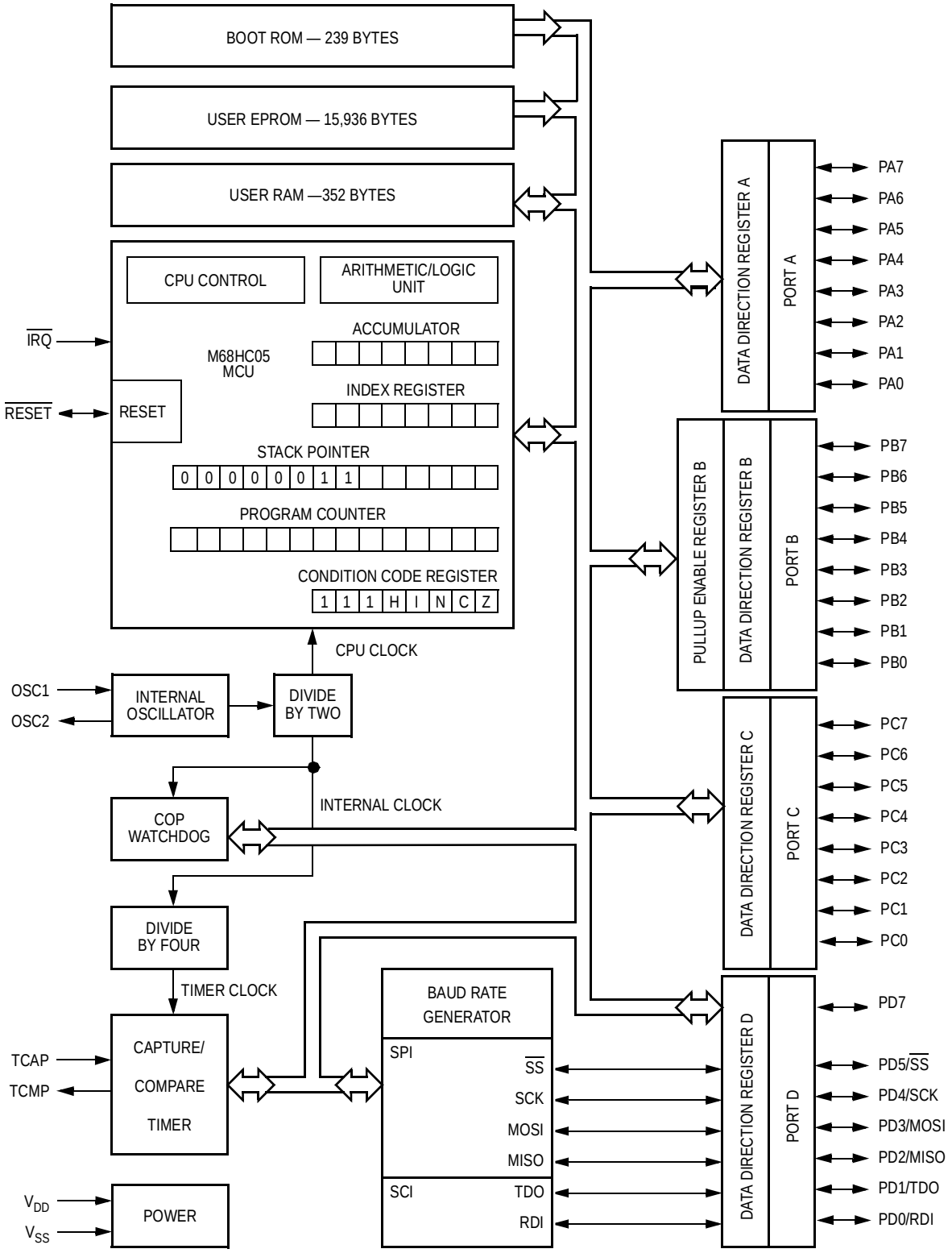


Figure 1-1. Block Diagram

1.4 Configuration Options

The options and functions of the MC68HC705C9A can be configured to emulate either the MC68HC05C9A or the MC68HC05C12A.

The ROM device MC68HC05C9A has eight ROM mask options to select external interrupt/internal pullup capability on each of the eight port B bits. Other optional features are controlled by software addressable registers during operation of the microcontroller. These features are IRQ sensitivity and memory map configuration.

On the ROM device MC68HC05C12A, all optional features are controlled by ROM mask options. These features are the eight port B interrupt/pullup options, IRQ sensitivity, STOP instruction disable, and COP enable.

On the MC68HC705C9A the ROM mask options of the MC68HC05C9A and the MC68HC05C12A are controlled by mask option registers (MORs). The MORs are EPROM registers which must be programmed appropriately prior to operation of the microcontroller. The software options of the MC68HC05C9A are implemented by identical software registers in the MC68HC705C9A.

When configured as an MC68HC05C9A:

- The entire 16K memory map of the C9A is enabled, including dual-mapped RAM and EPROM at locations $0020–$004F and $0100–$017F.
- C12A options in the C12MOR ($3FF1) are disabled.
- The C9A option register ($3FDF) is enabled, allowing software control over the IRQ sensitivity and the memory map configuration.
- The C9A COP reset register ($001D) and the C9A COP control register ($001E) are enabled, allowing software control over the C9A COP and clock monitor.
- The C12 COP clear register ($3FF0) is disabled.
- The port D data direction register ($0007) is enabled, allowing output capability on the seven port D pins.

- SPI output signals (MOSI, MISO, and SCK) require the corresponding bits in the port D data direction register to be set for output.
- The port D wire-OR mode control bit (bit 5 of SPCR $000A) is enabled, allowing open-drain configuration of port D.
- The $\overline{\text{RESET}}$ pin becomes bidirectional; this pin is driven low by a C9A COP or clock monitor timeout or during power-on reset.

When configured as an MC68HC05C12A:

- Memory locations $0100–$0FFF are disabled, creating a memory map identical to the MC68HC05C12A.
- C12A options in the C12MOR ($3FF1) are enabled; these bits control IRQ sensitivity, STOP instruction disable and C12 COP enable.
- The C9A option register ($3FDF) is disabled, preventing software control over the IRQ sensitivity and the memory map configuration.
- The C9A COP reset register ($001D) and the C9A COP control register ($001E) are disabled, preventing software control over the C9A COP and clock monitor.
- The C12 COP clear register ($3FF0) is enabled; this write-only register is used to clear the C12 COP.
- The port D data direction register ($0007) is disabled and the seven port D pins become input only.
- SPI output signals (MOSI, MISO, and SCK) do not require the data direction register control for output capability.
- The port D wire-OR mode control bit (bit 5 of SPCR $000A) is disabled, preventing open-drain configuration of port D.
- The $\overline{\text{RESET}}$ pin becomes input only.

1.5 Mask Options

The following two mask option registers are used to select features controlled by mask changes on the MC68HC05C9A and the MC68HC05C12A:

- Port B mask option register (PBMOR)
- C12 mask option register (C12MOR)

The mask option registers are EPROM locations which must be programmed prior to operation of the microcontroller.

1.5.1 Port B Mask Option Register (PBMOR)

The PBMOR register, shown in **Figure 1-2**, contains eight programmable bits which determine whether each port B bit (when in input mode) has the pullup and interrupt enabled. The port B interrupts share the vector and edge/edge-level sensitivity with the $\overline{\text{IRQ}}$ pin. For more details, (see **4.4 External Interrupt (IRQ or Port B)**).

$3FF0	Bit 7	6	5	4	3	2	1	Bit 0
	PBPU7	PBPU6	PBPU5	PBPU4	PBPU3	PBPU2	PBPU1	PBPU0

Figure 1-2. Port B Mask Option Register

PBPU7–PBPU0 — Port B Pullup/Interrupt Enable Bits
1 = Pullup and CPU interrupt enabled
0 = Pullup and CPU interrupt disabled

NOTE: *The current capability of the port B pullup devices is equivalent to the MC68HC05C9A, which is less than the MC68HC05C12A.*

1.5.2 C12 Mask Option Register (C12MOR)

The C12MOR register, shown in **Figure 1-3**, controls the following options:

- Select between MC68HC05C9A/C12A configuration
- Enable/disable stop mode (C12A mode only)
- Enable/disable COP (C12A mode only)
- Edge-triggered only or edge- and level-triggered external interrupt pin (IRQ pin) (C12A mode only).

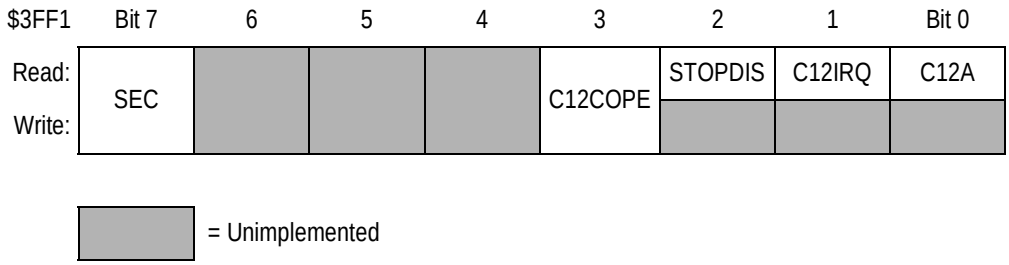


Figure 1-3. Mask Option Register 2

C12A — C12A/C9A Mode Select Bit

This read/write bit selects between C12A configuration and C9A configuration.

1 = Configured to emulate MC68HC05C12A
0 = Configured to emulate MC68HC05C9A

C12IRQ — C12A Interrupt Request Bit

This read/write bit selects between an edge-triggered only or edge- and level-triggered external interrupt pin. If configured in C9A mode, this bit has no effect and will be forced to 0 regardless of the programmed state.

1 = Edge and level interrupt option selected
0 = Edge-only interrupt option selected

NOTE: *Any Port B pin configured for interrupt capability will follow the same edge or edge/level trigger as the $\overline{IRQ}$ pin.*

STOPDIS — STOP Instruction Disable Bit

This read-only bit allows emulation of the “STOP disable” mask option on the MC68HC05C12A. (See **5.10 COP During Stop Mode**.) If configured in MC68HC05C9A mode, this bit has no effect and will be forced to 0 regardless of the programmed state.

1 = If the MCU enters stop mode, the clock monitor is enabled to force a system reset
0 = STOP instruction executed as normal

C12COPE — C12A COP Enable Bit

This read-only bit enables the COP function when configured in MC68HC05C12A mode. If configured in MC68HC05C9A mode, this bit has no effect and will be forced to 0 regardless of the programmed state.

1 = When in C12A mode, this enables the C12ACOP watchdog timer.
0 = When in C12A mode, this disables the C12ACOP watchdog timer.

SEC — Security Enable Bit

This read-only bit enables the EPROM security feature. Once programmed, this bit helps to prevent external access to the programmed EPROM data. The EPROM data cannot be verified or modified.

1 = Security enabled
0 = Security disabled

NOTE: *During power-on reset, the device always will be configured as MC68HC05C9A regardless of the state of the C12A bit.*

1.6 Software-Programmable Options (MC68HC05C9A Mode Only)

The C9A option register (OR), shown in **Figure 1-4**, is enabled only if configured in C9A mode. This register contains the programmable bits for the following options:

- Map two different areas of memory between RAM and EPROM, one of 48 bytes and one of 128 bytes
- Edge-triggered only or edge- and level-triggered external interrupt ($\overline{\text{IRQ}}$ pin and any port B pin configured for interrupt)

This register must be written to by user software during operation of the microcontroller.

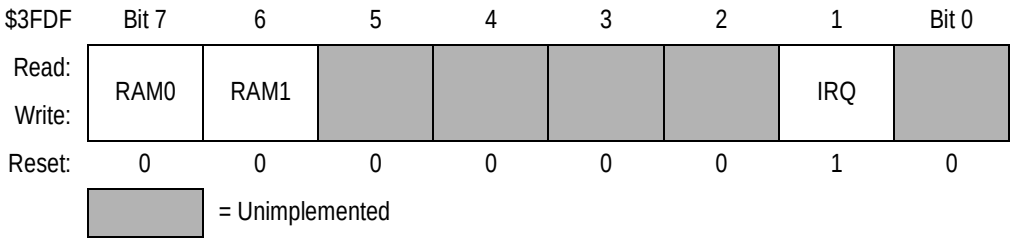


Figure 1-4. C9A Option Register

RAM0 — Random Access Memory Control Bit 0

This read/write bit selects between RAM or EPROM in location $0020 to $004F. This bit can be read or written at any time.

1 = RAM selected
0 = EPROM selected

RAM1— Random Access Memory Control Bit 1

This read/write bit selects between RAM or EPROM in location $0100 to $017F. This bit can be read or written at any time.

1 = RAM selected
0 = EPROM selected

IRQ — Interrupt Request Bit

This bit selects between an edge-triggered only or edge- and level-triggered external interrupt pin. This bit is set by reset, but can be cleared by software. This bit can be written only once.

1 = Edge and level interrupt option selected
0 = Edge-only interrupt option selected

1.7 Functional Pin Descriptions

Figure 1-5, **Figure 1-6**, **Figure 1-7**, and **Figure 1-8** show the pin assignments for the available packages. A functional description of the pins follows.

NOTE: *A line over a signal name indicates an active low signal. For example, RESET is active high and $\overline{RESET}$ is active low.*

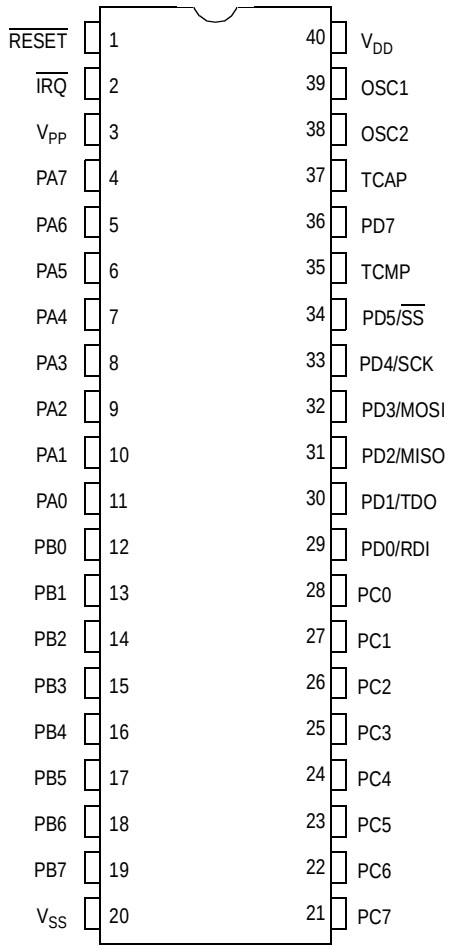


Figure 1-5. 40-Pin PDIP Pin Assignments

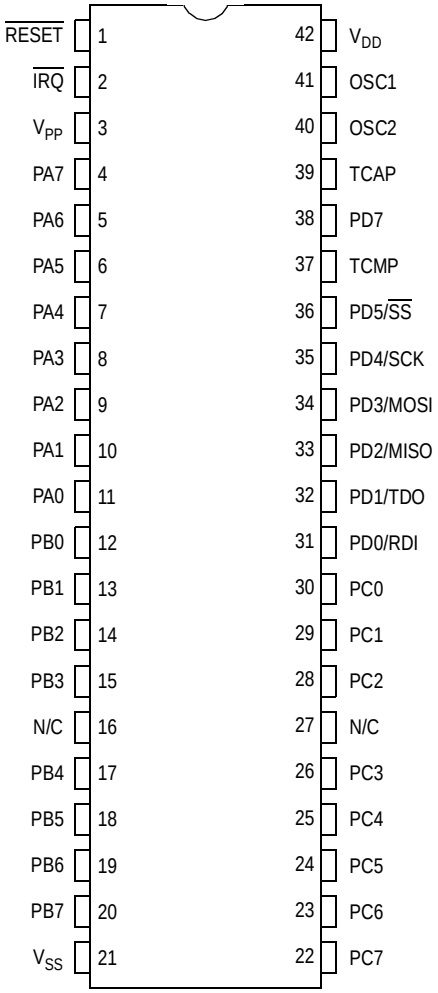


Figure 1-6. 42-Pin SDIP Pin Assignments

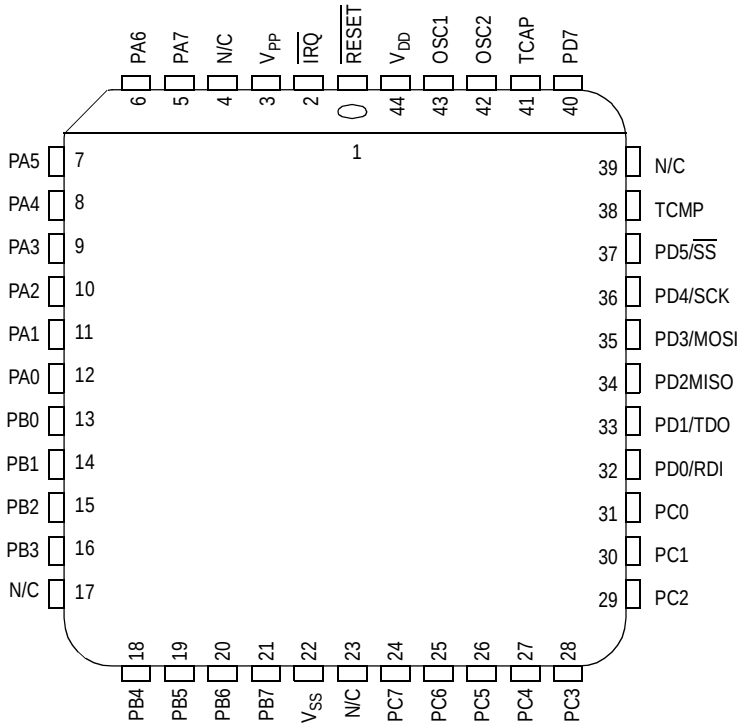


Figure 1-7. 44-Lead PLCC Pin Assignments

NOTE: *The above 44-pin PLCC pin assignment diagram is for compatibility with MC68HC05C9A. To allow compatibility with the 44-pin PLCC MC68HC05C12A, pin17 and pin18 must be tied together and pin 39 and pin 40 also must be tied together.*

To allow compatibility with MC68HC705C8A, pin 3 and pin 4 also should be tied together.

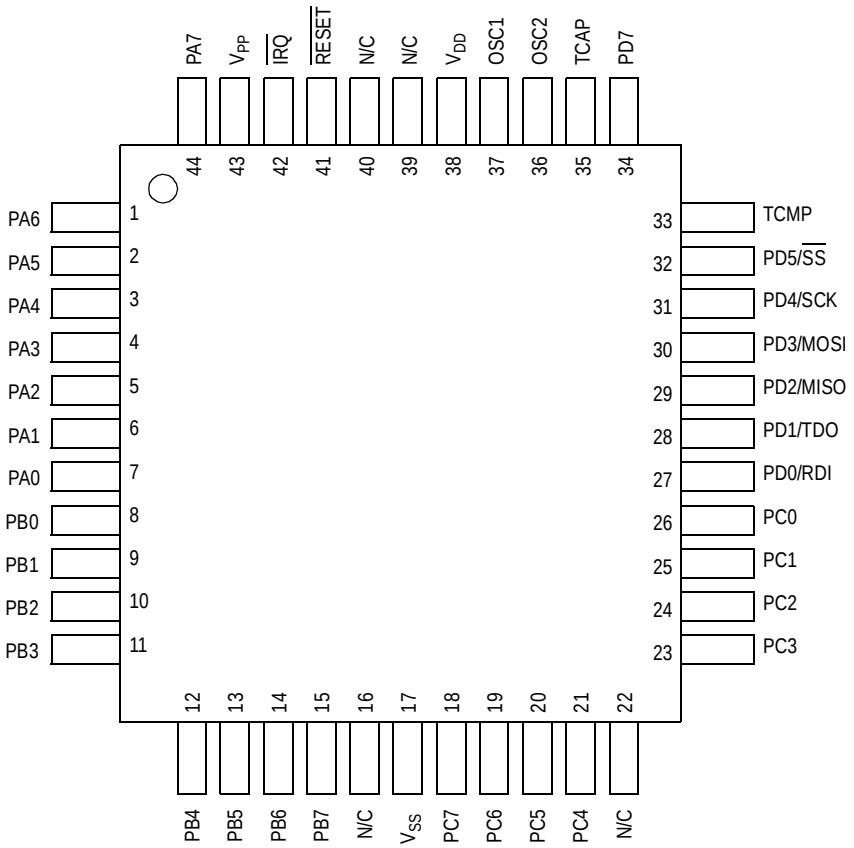


Figure 1-8. 44-Pin QFP Pin Assignments

1.7.1 V_{DD} and V_{SS}

Power is supplied to the MCU using these two pins. V_{DD} is the positive supply and V_{SS} is ground.

1.7.2 V_{PP}

This pin provides the programming voltage to the EPROM array. For normal operation, V_{PP} should be tied to V_{DD}.

1.7.3 $\overline{\text{IRQ}}$

This interrupt pin has an option that provides two different choices of interrupt triggering sensitivity. The $\overline{\text{IRQ}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. Refer to **Section 4. Interrupts** for more detail.

1.7.4 OSC1 andOSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal connected to these pins provides a system clock. The internal frequency is one-half the crystal frequency.

1.7.5 $\overline{\text{RESET}}$

As an input pin, this active low $\overline{\text{RESET}}$ pin is used to reset the MCU to a known startup state by pulling $\overline{\text{RESET}}$ low. As an output pin, when in MC68HC05C9A mode only, the $\overline{\text{RESET}}$ pin indicates that an internal MCU reset has occurred. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. Refer to **Section 5. Resets** for more detail.

1.7.6 TCAP

This pin controls the input capture feature for the on-chip programmable timer. The TCAP pin contains an internal Schmitt trigger as part of its input to improve noise immunity. Refer to **Section 8. Capture/Compare Timer** for more detail.

1.7.7 TCMP

The TCMP pin provides an output for the output compare feature of the on-chip programmable timer. Refer to **Section 8. Capture/Compare Timer** for more detail.

1.7.8 PA0–PA7

These eight I/O lines comprise port A. The state of each pin is software programmable and all port A pins are configured as inputs during reset. Refer to **Section 7. Input/Output Ports** for more detail.

1.7.9 PB0–PB7

These eight I/O lines comprise port B. The state of each pin is software programmable and all port B pins are configured as inputs during reset. Port B has mask option register enabled pullup devices and interrupt capability selectable for any pin. Refer to **Section 7. Input/Output Ports** for more detail.

1.7.10 PC0–PC7

These eight I/O lines comprise port C. The state of each pin is software programmable and all port C pins are configured as inputs during reset. PC7 has high current sink and source capability. Refer to **Section 7. Input/Output Ports** for more detail.

1.7.11 PD0–PD5 and PD7

These seven I/O lines comprise port D. When configured as a C9A the state of each pin is software programmable and all port D pins are configured as inputs during reset. When configured as a C12A, the port D pins are input only. Refer to **Section 7. Input/Output Ports** for more detail.

General Description

Section 2. Memory

2.1 Contents

2.2 Introduction

The MCU has a 16-Kbyte memory map when configured as either an MC68HC05C9A or an MC68HC05C12A. The memory map consists of registers (I/O, control, and status), user RAM, user EPROM, bootloader ROM, and reset and interrupt vectors as shown in **Figure 2-1** and **Figure 2-2**.

When configured as an MC68HC05C9A, two control bits in the option register ($3FDF) allow the user to switch between RAM and EPROM at any time in two special areas of the memory map, $0020-$004F (48 bytes) and $0100-$017F (128 bytes). When configured as an MC68HC05C12A, the section of the memory map from $0020 to $004F is fixed as EPROM and the section from $0100 to $0FFF becomes unused.

2.3 RAM

The main user RAM consists of 176 bytes at $0050–$00FF. This RAM area is always present in the memory map and includes a 64-byte stack area. The stack pointer can access 64 bytes of RAM in the range $00FF down to $00C0.

NOTE: *Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.*

In MC68HC05C9A configuration, two additional RAM areas are available at $0020–$004F (48 bytes) and $0100–$017F (128 bytes) (see **Figure 2-1** and **Figure 2-2**.) These may be accessed at any time by setting the RAM0 and RAM1 bits, respectively, in the C9A option register. Refer to **1.6 Software-Programmable Options (MC68HC05C9A Mode Only)** for additional information.

2.4 EPROM

When configured as a C12A the main user EPROM consists of 48 bytes of page zero EPROM from $0020 to $004F, 12,032 bytes of EPROM from $1000 to $3EFF, and 14 bytes of user vectors from $3FF4 to $3FFF. When configured as a C9A, an additional 3,840 bytes of user EPROM from $0100 to $0FFF are enabled.

Locations $3FF0 and $3FF1 are the mask option registers (MOR) (see **1.5 Mask Options**).

For detailed information on programming the EPROM see **Appendix A. EPROM Programming**.

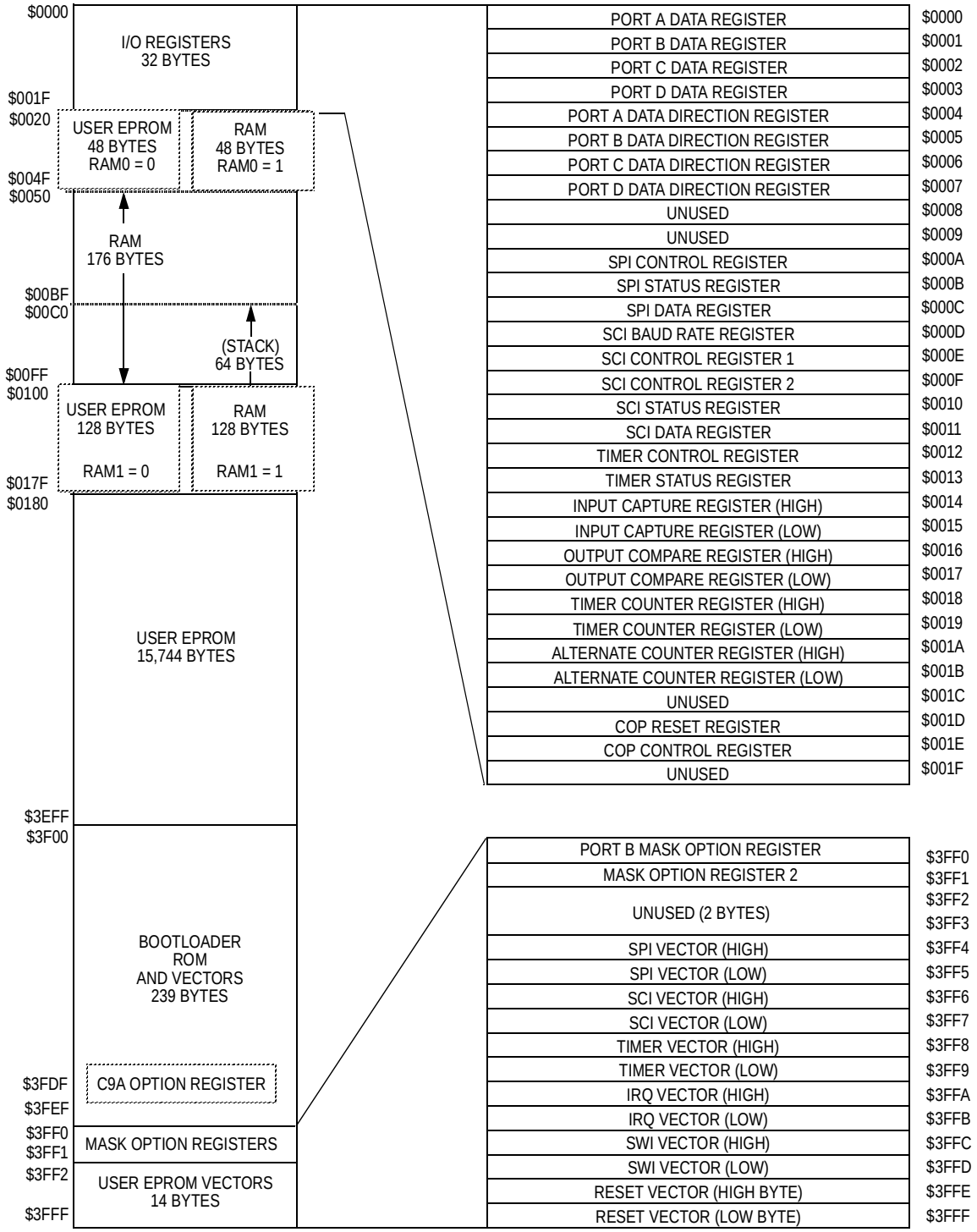


Figure 2-1. C9A Memory Map

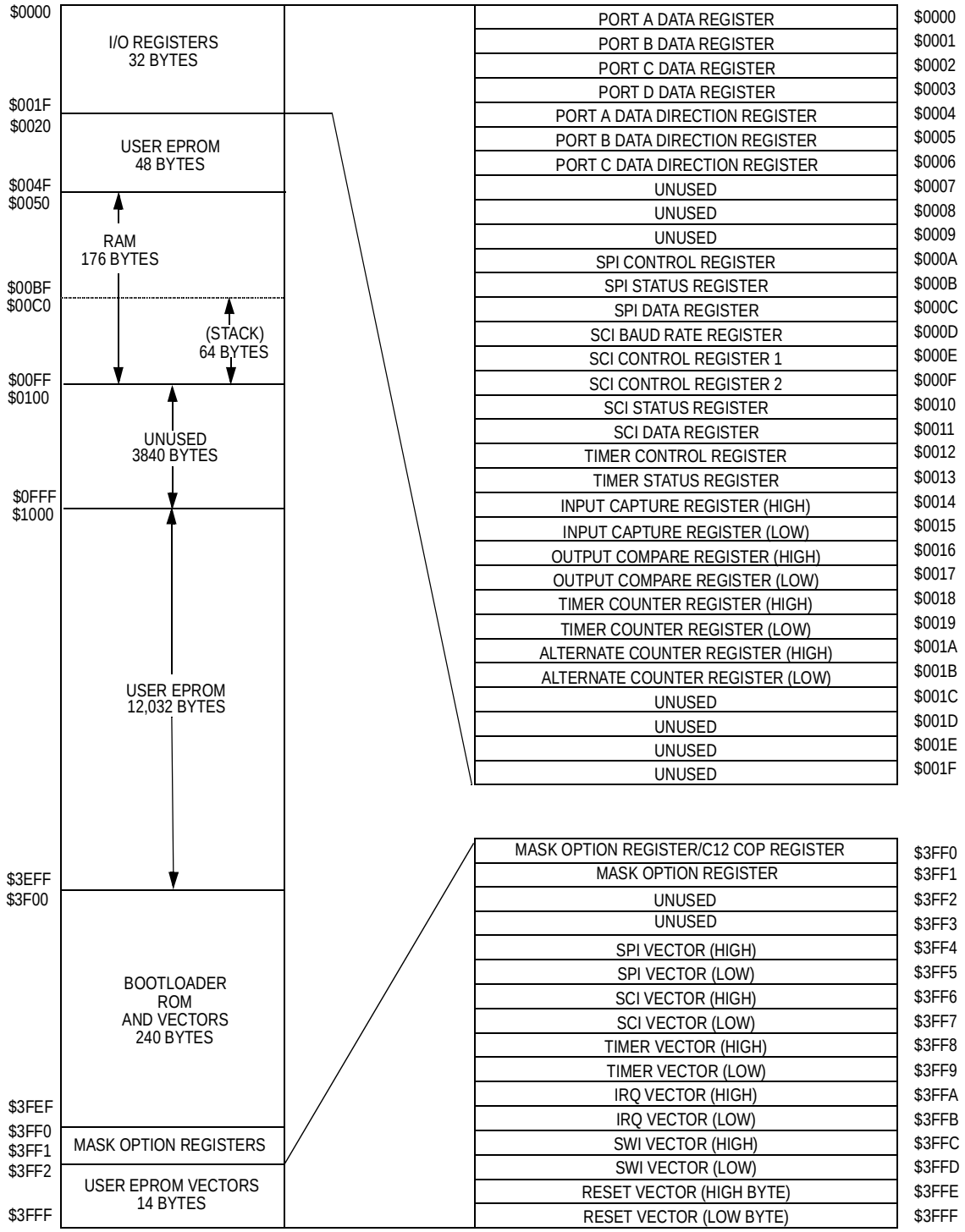


Figure 2-2. C12A Memory Map

2.5 EPROM Security

A security feature has been incorporated into the MC68HC705C9A to help prevent external access to the contents of the EPROM in any mode of operation. Once enabled, this feature can be disabled only by completely erasing the EPROM.

NOTE: *For OTP (plastic) packages, once the security feature has been enabled, it cannot be disabled.*

2.6 ROM

The bootloader ROM occupies 239 bytes in the memory space from $3F00 to $3FEF. The bootloader mode provides for self-programming of the EPROM array. See **Appendix A. EPROM Programming**.

2.7 I/O Registers

Except for the option register, mask option registers, and the C12 COP clear register, all I/O, control and status registers are located within one 32-byte block in page zero of the address space ($0000–$001F). A summary of these registers is shown in **Figure 2-3**. More detail about the contents of these registers is given **Figure 2-4**.

Addr	Register Name
$0000	Port A Data Register
$0001	Port B Data Register
$0002	Port C Data Register
$0003	Port D Data Register
$0004	Port A Data Direction Register
$0005	Port B Data Direction Register
$0006	Port C Data Direction Register
$0007	Port D Data Direction Register (C9A Only)
$0008	Unused
$0009	Unused
$000A	Serial Peripheral Control Register
$000B	Serial Peripheral Status Register
$000C	Serial Peripheral Data Register
$000D	Baud Rate Register
$000E	Serial Communications Control Register 1
$000F	Serial Communications Control Register 2
$0010	Serial Communications Status Register
$0011	Serial Communications Data Register
$0012	Timer Control Register
$0013	Timer Status Register
$0014	Input Capture Register High
$0015	Input Capture Register Low
$0016	Output Compare Register High
$0017	Output Compare Register Low
$0018	Timer Register High
$0019	Timer Register Low
$001A	Alternate Timer Register High
$001B	Alternate Timer Register Low
$001C	EPROM Programming Register
$001D	C9A COP Reset Register
$001E	C9A COP Control Register
$001F	Reserved

Figure 2-3. I/O Register Summary

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PORTA) See page 69.	Read: Write:	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PORTB) See page 70.	Read: Write:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
		Reset:	Unaffected by reset							
$0002	Port C Data Register (PORTC) See page 71.	Read: Write:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
		Reset:	Unaffected by reset							
$0003	Port D Data Register (PORTD) See page 71.	Read: Write:	PD7		PD5	PD4	PD3	PD2	PD1	PD0
		Reset:	Unaffected by reset							
$0004	Port A Data Direction Register (DDRA) See page 69.	Read: Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Reset:	0	0	0	0	0	0	0	0
$0005	Port B Data Direction Register (DDRB) See page 70.	Read: Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Reset:	0	0	0	0	0	0	0	0
$0006	Port C Data Direction Register (DDRC) See page 71.	Read: Write:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Reset:	0	0	0	0	0	0	0	0
$0007	Port D Data Direction Register (DDRD) **C9A Only** See page 71.	Read: Write:	DDRC7		DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Reset:	0	0	0	0	0	0	0	0
$0008	Unimplemented									

= Unimplemented R = Reserved U = Unaffected

Figure 2-4. Input/Output Registers (Sheet 1 of 4)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0009	Unimplemented									
$000A	SPI Control Register (SPCR) See page 109.	Read: / Write:	SPIE	SPE	DWOM (C9A)	MSTR	CPOL	CPHA	SPR1	SPR0
		Reset:	0	0	0	0	0	1	U	U
$000B	SPI Status Register (SPSR) See page 111.	Read:	SPIF	WCOL		MODF				
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000C	SPI Data Register (SPDR) See page 113.	Read: / Write:	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
		Reset:	Unaffected by reset							
$000D	SCI Baud Rate Register BAUD See page 101.	Read:			SCP1	SCP0		SCR2	SCR1	SCR0
		Write:								
		Reset:	—	—	0	0	—	U	U	U
$000E	SCI Control Register 1 (SCCR1) See page 95.	Read:	R8	T8		M	WAKE			
		Write:								
		Reset:	U	U	0	U	U	0	0	0
$000F	SCI Control Register 2 (SCCR2) See page 96.	Read: / Write:	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
		Reset:	0	0	0	0	0	0	0	0
$0010	SCI Status Register (SCSR) See page 99.	Read:	TDRE	TC	RDRF	IDLE	OR	NF	FE	
		Write:								
		Reset:	1	1	0	0	0	0	0	—
$0011	SCI Data Register (SCDR) See page 94.	Read: / Write:	SCD7	SDC6	SCD5	SCD4	SCD3	SCD2	SCD1	SCD0
		Reset:	Unaffected by reset							

= Unimplemented R = Reserved U = Unaffected

Figure 2-4. Input/Output Registers (Sheet 2 of 4)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0012	Timer Control Register (TCR) See page 76.	Read:	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
		Write:								
		Reset:	0	0	0	0	0	0	U	0
$0013	Timer Status Register (TSR) See page 78.	Read:	ICF	OCF	TOF	0	0	0	0	0
		Write:								
		Reset:	U	U	U	0	0	0	0	0
$0014	Input Capture Register High (ICRH) See page 81.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Unaffected by reset							
$0015	Input Capture Register Low (ICRL) See page 81.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Unaffected by reset							
$0016	Output Compare Register High (OCRH) See page 82.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Unaffected by reset							
$0017	Output Compare Register Low (OCRL) See page 82.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Unaffected by reset							
$0018	Timer Register High (TRH) See page 79.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0019	Timer Register Low (TRL) See page 79.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	0	0
$001A	Alternate Timer Register High (ATRH) See page 80.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1

= Unimplemented R = Reserved U = Unaffected

Figure 2-4. Input/Output Registers (Sheet 3 of 4)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001B	Alternate Timer Register Low (ATRL) See page 80.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	0	0
$001C	EPROM Programming Register (EPR)	Read:						LATCH		EPGM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$001D	COP Reset Register (COPRST) **C9A Only** See page 61.	Read:								
		Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	0	0	0	0	0	0	0	0
$001E	COP Control Register (COPCR) **C9A Only** See page 62.	Read:	0	0	0	COPF	CME	COPE	CM1	CM0
		Write:								
		Reset:	0	0	0	U	0	0	0	0
$001F	Reserved		R	R	R	R	R	R	R	R

= Unimplemented R = Reserved U = Unaffected

Figure 2-4. Input/Output Registers (Sheet 4 of 4)

Section 3. Central Processor Unit (CPU)

3.1 Contents

3.2 Introduction

This section contains the basic programmers model and the registers contained in the CPU.

3.3 CPU Registers

The MCU contains five registers as shown in the programming model of **Figure 3-1**. The interrupt stacking order is shown in **Figure 3-2**.

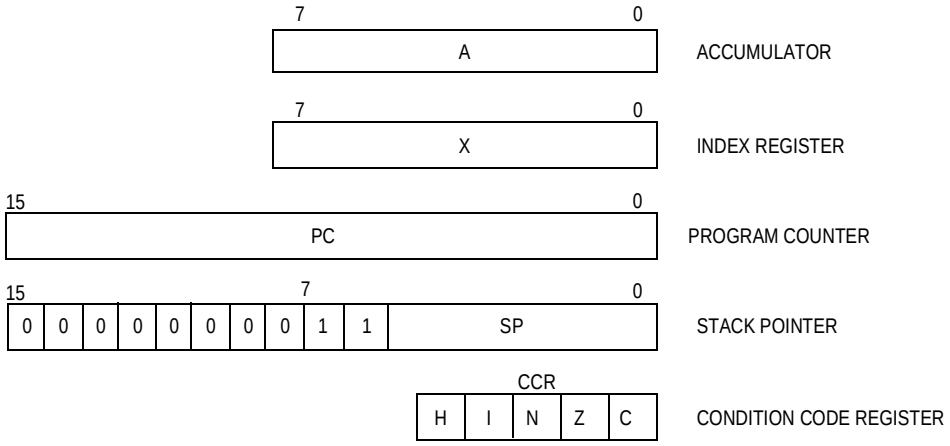


Figure 3-1. Programming Model

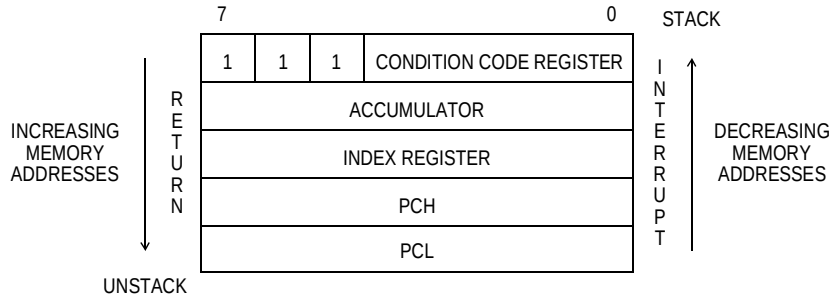


Figure 3-2. Interrupt Stacking Order

3.3.1 Accumulator (A)

The accumulator is a general-purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

3.3.2 Index Register (X)

The index register is an 8-bit register used for the indexed addressing value to create an effective address. The index register may also be used as a temporary storage area.

3.3.3 Program Counter (PC)

The program counter is a 16-bit register that contains the address of the next byte to be fetched.

3.3.4 Stack Pointer (SP)

The stack pointer contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $0FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the eight most significant bits are permanently set to 00000011. These eight bits are appended to the six least significant register bits to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.

3.3.5 Condition Code Register (CCR)

The CCR is a 5-bit register in which four bits are used to indicate the results of the instruction just executed, and the fifth bit indicates whether interrupts are masked. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

When this bit is set, the timer, SCI, SPI, and external interrupt are masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and processed as soon as the interrupt bit is cleared.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative.

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions and during shifts and rotates.

Section 4. Interrupts

4.1 Contents

4.2 Introduction

The MCU can be interrupted by five different sources, four maskable hardware interrupts, and one non-maskable software interrupt:

- External signal on the $\overline{\text{IRQ}}$ pin or port B pins
- 16-bit programmable timer
- Serial communications interface
- Serial peripheral interface
- Software interrupt instruction (SWI)

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume.

Unlike reset, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is complete.

NOTE: *The current instruction is the one already fetched and being operated on.*

When the current instruction is complete, the processor checks all pending hardware interrupts. If interrupts are not masked (CCR I bit clear) and if the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If an external interrupt and a timer, SCI, or SPI interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction, regardless of the I-bit state.

Table 4-1 shows the relative priority of all the possible interrupt sources. **Figure 4-1** shows the interrupt processing flow.

Table 4-1. Vector Addresses for Interrupts and Resets

Function	Source	Local Mask	Global Mask	Priority (1 = Highest)	Vector Address
Reset	Power-on reset	None	None	1	$3FFE–$3FFF
	$\overline{\text{RESET}}$ pin				
	COP watchdog				
Software interrupt (SWI)	User code	None	None	Same priority as instruction	$3FFC–$3FFD
External interrupt	$\overline{\text{IRQ}}$ pin port B pins	None	I bit	2	$3FFA–$3FFB
Timer interrupts	ICF bit	ICIE bit	I bit	3	$3FF8–$3FF9
	OCF bit	OCIE bit			
	TOF bit	TOIE bit			
SCI interrupts	TDRE bit	TCIE bit	I bit	4	$3FF6–$3FF7
	TC bit				
	RDRF bit	RIE bit			
	OR bit				
	IDLE bit	ILIE bit			
SPI interrupts	SPIF bit	SPIE	I bit	5	$3FF4–$3FF5
	MODF bit				

4.3 Non-Maskable Software Interrupt (SWI)

The SWI is an executable instruction and a non-maskable interrupt: It is executed regardless of the state of the I bit in the CCR. If the I bit is zero (interrupts enabled), SWI executes after interrupts which were pending when the SWI was fetched, but before interrupts generated after the SWI was fetched. The interrupt service routine address is specified by the contents of memory locations $3FFC and $3FFD.

4.4 External Interrupt ($\overline{\text{IRQ}}$ or Port B)

If the interrupt mask bit (I bit) of the CCR is set, all maskable interrupts (internal and external) are disabled. Clearing the I bit enables interrupts. The interrupt request is latched immediately following the falling edge of $\overline{\text{IRQ}}$. It is then synchronized internally and serviced as specified by the contents of $3FFA and $3FFB.

When any of the port B pullups are enabled, each pin becomes an additional external interrupt source which is executed identically to the $\overline{\text{IRQ}}$ pin. Port B interrupts follow the same edge/edge-level selection as the $\overline{\text{IRQ}}$ pin. The branch instructions BIL and BIH also respond to the port B interrupts in the same way as the $\overline{\text{IRQ}}$ pin. See **7.4 Port B**.

Either a level-sensitive and edge-sensitive trigger or an edge-sensitive-only trigger operation is selectable. In MC68HC05C9A mode, the sensitivity is software controlled by the IRQ bit in the C9A option register ($3FDF). In the MC68HC05C12A mode, the sensitivity is determined by the C12IRQ bit in the C12 mask option register ($3FF1).

NOTE: *The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, one external interrupt pulse can be latched and serviced as soon as the I bit is cleared.*

4.5 Timer Interrupt

Three different timer interrupt flags cause a timer interrupt whenever they are set and enabled. The interrupt flags are in the timer status register (TSR), and the enable bits are in the timer control register (TCR). Any of these interrupts will vector to the same interrupt service routine, located at the address specified by the contents of memory locations $3FF8 and $3FF9.

4.6 SCI Interrupt

Five different SCI interrupt flags cause an SCI interrupt whenever they are set and enabled. The interrupt flags are in the SCI status register (SCSR), and the enable bits are in the SCI control register 2 (SCCR2). Any of these interrupts will vector to the same interrupt service routine, located at the address specified by the contents of memory locations $3FF6 and $3FF7.

4.7 SPI Interrupt

Two different SPI interrupt flags cause an SPI interrupt whenever they are set and enabled. The interrupt flags are in the SPI status register (SPSR), and the enable bits are in the SPI control register (SPCR). Either of these interrupts will vector to the same interrupt service routine, located at the address specified by the contents of memory locations $3FF4 and $3FF5.

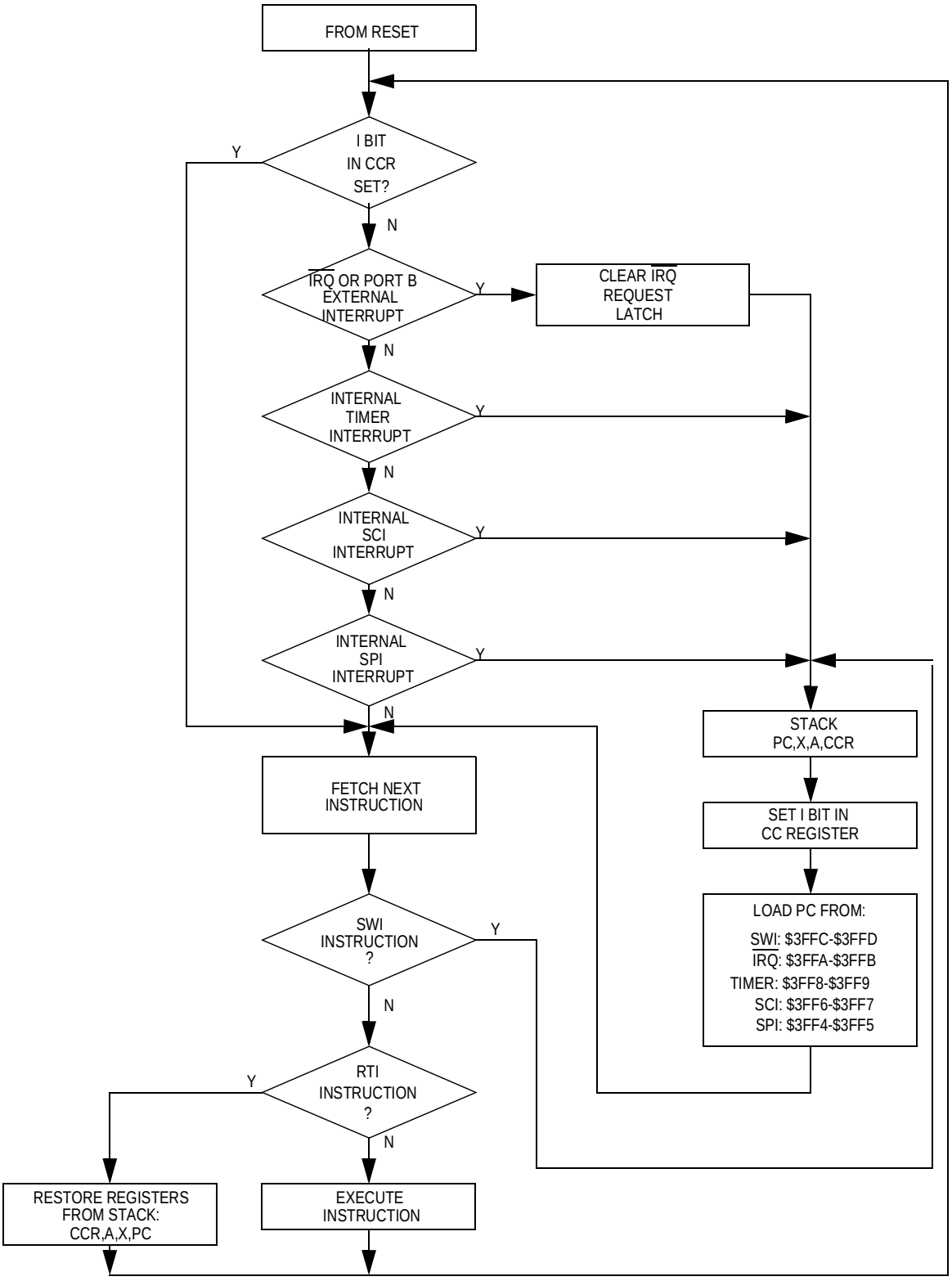


Figure 4-1. Interrupt Flowchart

Interrupts

Section 5. Resets

5.1 Contents

5.2 Introduction

The MCU can be reset four ways: by the initial power-on reset function, by an active low input to the $\overline{\text{RESET}}$ pin, by the COP, or by the clock monitor. A reset immediately stops the operation of the instruction being executed, initializes some control bits, and loads the program counter with a user-defined reset vector address. **Figure 5-1** is a block diagram of the reset sources.

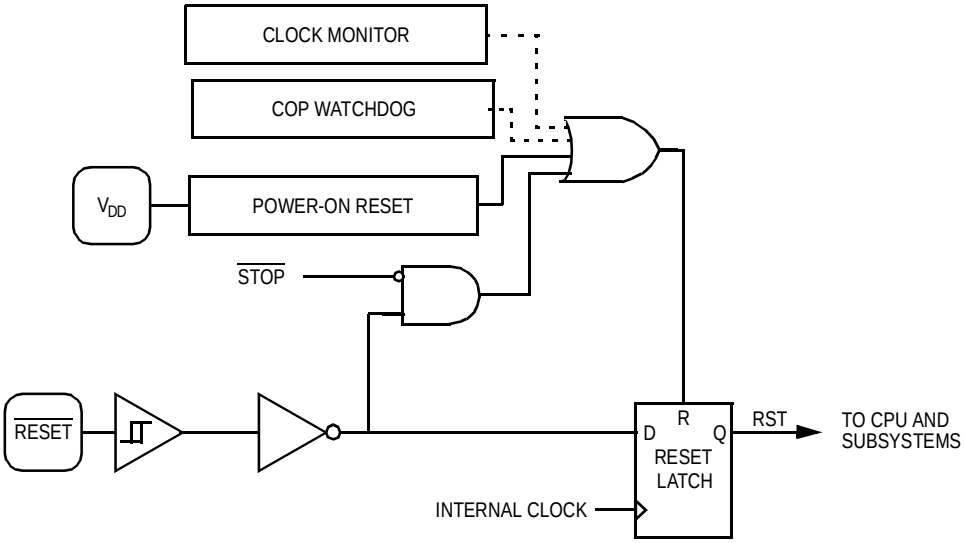


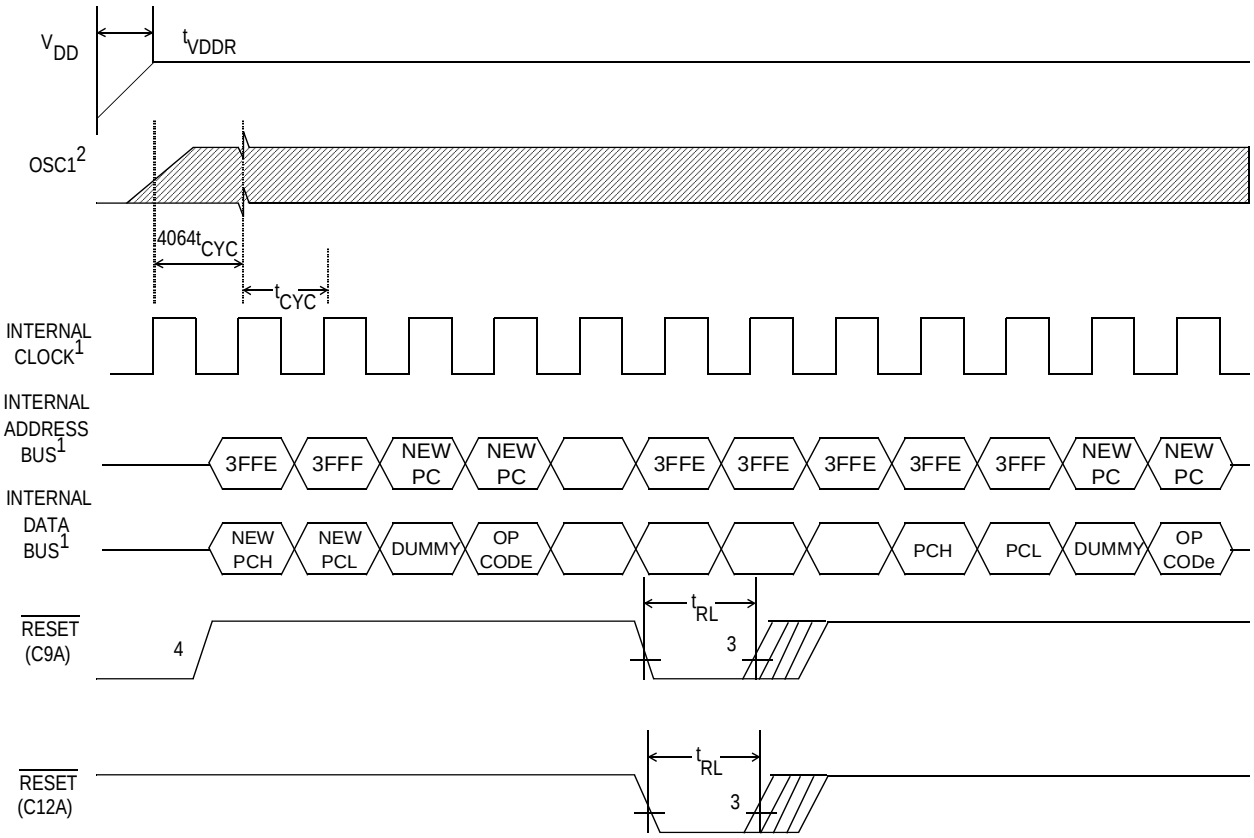
Figure 5-1. Reset Sources

5.3 Power-On Reset (POR)

A power-on-reset occurs when a positive transition is detected on V_{DD}. The power-on reset is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. There is a 4064 internal processor clock cycle (t_{cyc}) oscillator stabilization delay after the oscillator becomes active. (When configured as a C9A, the $\overline{\text{RESET}}$ pin will output a logic 0 during the 4064-cycle delay.) If the $\overline{\text{RESET}}$ pin is low after the end of this 4064-cycle delay, the MCU will remain in the reset condition until $\overline{\text{RESET}}$ is driven high externally.

5.4 $\overline{\text{RESET}}$ Pin

The function of the $\overline{\text{RESET}}$ pin is dependent on whether the device is configured as an MC68HC05C9A or an MC68HC05C12A. When it is in the MC68HC05C12A configuration, the pin is input only. When in MC68HC05C9A configuration the pin is bidirectional. In both cases the MCU is reset when a logic 0 is applied to the $\overline{\text{RESET}}$ pin for a period of one and one-half machine cycles (t_{RL}). For the MC68HC05C9A configuration, the $\overline{\text{RESET}}$ pin will be driven low by a COP, clock monitor, or power-on reset.



Notes:
1. Internal timing signal and bus information are not available externally.
2. OSC1 line is not meant to represent frequency. It is only meant to represent time.
3. The next rising edge of the internal processor clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence.
4. $\overline{\text{RESET}}$ outputs V_{OL} during 4064 power-on reset cycles when in C9A mode only.

Figure 5-2. Power-On Reset and $\overline{\text{RESET}}$

5.5 Computer Operating Properly (COP) Reset

This device includes a watchdog COP feature which guards against program run-away failures. A timeout of the computer operating properly (COP) timer generates a COP reset. The COP watchdog is a software error detection system that automatically times out and resets the MCU if not cleared periodically by a program sequence.

This device includes two COP types, one for C12A compatibility and the other for C9A compatibility. When configured as a C9A the COP can be enabled by user software by setting COPE in the C9A COP control register (C9ACOPCR). When configured as a C12A, the COP is enabled prior to operation by programming the C12COPE bit in the C12A mask option register (C12MOR). The function and control of both COPs is detailed below.

5.6 MC68HC05C9A Compatible COP

This COP is controlled with two registers; one to reset the COP timer and the other to enable and control COP and clock monitor functions. **Figure 5-3** shows a block diagram of the MC68HC05C9A COP.

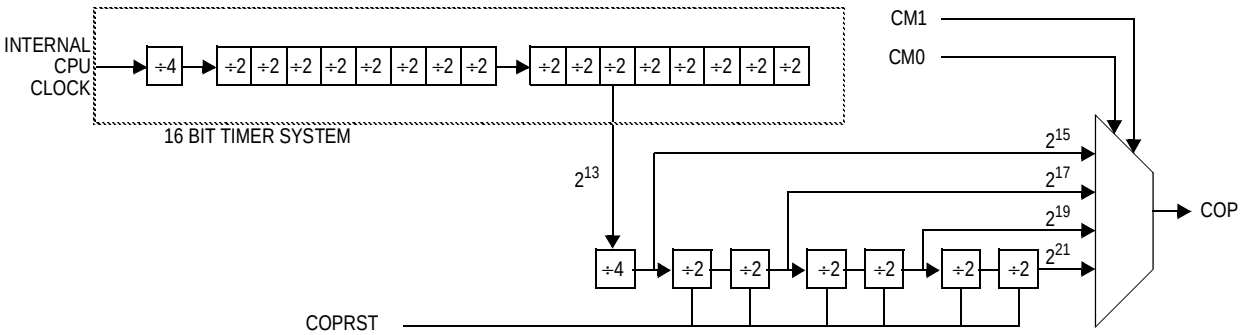


Figure 5-3. C9A COP Block Diagram

5.6.1 C9A COP Reset Register

This write-only register, shown in **Figure 5-4**, is used to reset the COP.

$001D	Bit 7	6	5	4	3	2	1	Bit 0
Read:								
Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 5-4. COP Reset Register (COPRST)

The sequence required to reset the COP timer is:

- Write $55 to the COP reset register
- Write $AA to the COP reset register

Both write operations must occur in the order listed, but any number of instructions may be executed between the two write operations provided that the COP does not time out between the two writes. The elapsed time between software resets must not be greater than the COP timeout period. If the COP should time out, a system reset will occur and the device will be re-initialized in the same fashion as a power-on reset or reset.

Reading this register does not return valid data.

5.6.2 C9A COP Control Register

The COP control register, shown in **Figure 5-5**, performs these functions:

- Enables clock monitor function
- Enables MC68HC05C9A compatible COP function
- Selects timeout duration of COP timer

and flags the following conditions:

- A COP timeout
- Clock monitor reset

$001E	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	COPF	CME	COPE	CM1	CM0
Write:								
Reset:	0	0	0	U	0	0	0	0

= Unimplemented U = Undetermined

Figure 5-5. COP Control Register (COPCR)

COPF — Computer Operating Properly Flag

Reading the COP control register clears COPF.

1 = COP or clock monitor reset has occurred.
0 = No COP or clock monitor reset has occurred.

CME — Clock Monitor Enable Bit

This bit is readable any time, but may be written only once.

1 = Clock monitor enabled
0 = Clock monitor disabled

COPE — COP Enable Bit

This bit is readable any time. COPE, CM1, and CM0 together may be written with a single write, only once, after reset. This bit is cleared by reset.

1 = COP enabled
0 = COP disabled

CM1 — COP Mode Bit 1

Used in conjunction with CM0 to establish the COP timeout period, this bit is readable any time. COPE, CM1, and CM0 together may be written with a single write, only once, after reset. This bit is cleared by reset.

CM0 — COP Mode Bit 0

Used in conjunction with CM1 to establish the COP timeout period, this bit is readable any time. COPE, CM1, and CM0 together may be written with a single write, only once, after reset. This bit is cleared by reset.

Bits 7–5 — Not Used

These bits always read as 0.

Table 5-1. COP Timeout Period

CM1	CM0	$f_{op}/2^{15}$ Divide By	Timeout Period (f_{osc} = 2.0 MHz)	Timeout Period (f_{osc} = 4.0 MHz)
0	0	1	32.77 ms	16.38 ms
0	1	4	131.07 ms	65.54 ms
1	0	16	524.29 ms	262.14 ms
1	1	64	2.097 sec	1.048 sec

5.7 MC68HC05C12A Compatible COP

This COP is implemented with an 18-bit ripple counter. This provides a timeout period of 64 milliseconds at a bus rate (f_{op}) of 2 MHz. If the COP should time out, a system reset will occur and the device will be re-initialized in the same fashion as a power-on reset or reset.

5.8 MC68HC05C12A Compatible COP Clear Register

The COP clear register, shown in **Figure 5-6**, resets the C12A COP counter.

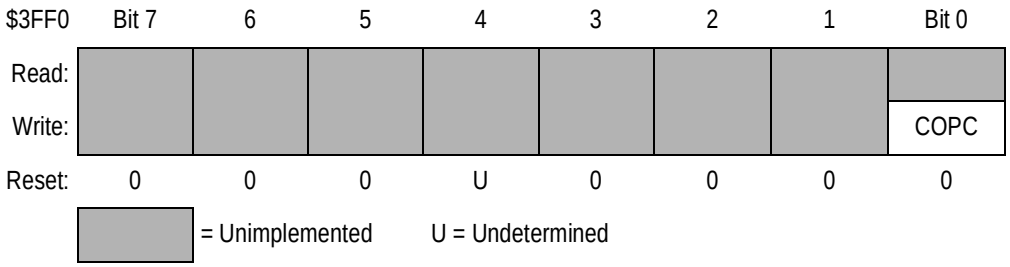


Figure 5-6. COP Clear Register (COPCLR)

COPC — Computer Operating Properly Clear Bit

Preventing a COP reset is achieved by writing a 0 to the COPC bit. This action will reset the counter and begin the timeout period again. The COPC bit is bit 0 of address $3FF0. A read of address $3FF0 will result in the data programmed into the mask option register PBMOR.

5.9 COP During Wait Mode

Either COP will continue to operate normally during wait mode. The software must pull the device out of wait mode periodically and reset the COP to prevent a system reset.

5.10 COP During Stop Mode

Stop mode disables the oscillator circuit and thereby turns the clock off for the entire device. The COP counter will be reset when stop mode is entered. If a reset is used to exit stop mode, the COP counter will be reset after the 4064 cycles of delay after stop mode. If an IRQ is used to exit stop mode, the COP counter will not be reset after the 4064-cycle delay and will have that many cycles already counted when control is returned to the program.

In the event that an inadvertent STOP instruction is executed, neither COP will allow the system to recover. The MC68HC705C9A offers two solutions to this problem, one available in C9A mode (see **5.10.1 Clock Monitor Reset**) and one available in C12A mode (see **5.10.2 STOP Instruction Disable Option**).

5.10.1 Clock Monitor Reset

When configured as a C9A, the clock monitor circuit can provide a system reset if the clock stops for any reason, including stop mode. When the CME bit in the C9A COP control register is set, the clock monitor detects the absence of the internal bus clock for a certain period of time. The timeout period is dependent on the processing parameters and varies from 5 µs to 100 µs, which implies that systems using a bus clock rate of 200 kHz or less should not use the clock monitor.

If a slow or absent clock is detected, the clock monitor causes a system reset. The reset is issued to the external system via the bidirectional $\overline{\text{RESET}}$ pin for four bus cycles if the clock is slow or until the clocks recover in the case where the clocks are absent.

5.10.2 STOP Instruction Disable Option

On the ROM device MC68HC05C12A, stop mode can be disabled by mask option. This causes the CPU to interpret the STOP instruction as a NOP so the device will never enter stop mode; if the user code is not being executed properly, the COP will provide a system reset.

To emulate this feature on the MC68HC705C9A (configured as an MC68HC05C12A), set the STOPDIS bit in the C12MOR. Stop mode will not actually be disabled as on the MC68HC05C12A, but the clock monitor circuit will be activated. If the CPU executes a STOP instruction, the clock monitor will provide a system reset.

NOTE: *This feature cannot be used with operating frequencies of 200 kHz or less.*

Resets

Section 6. Low-Power Modes

6.1 Contents

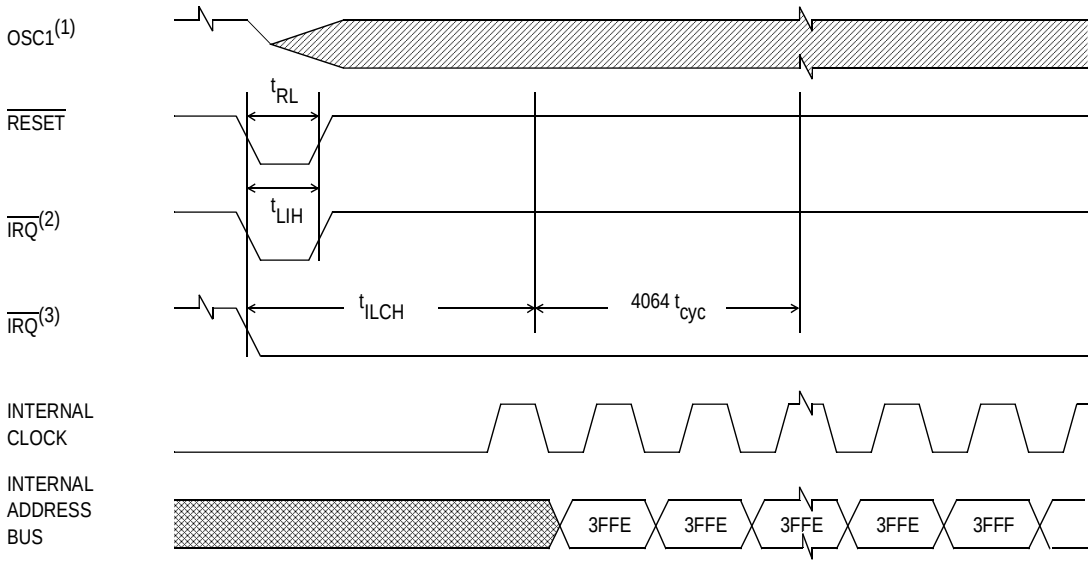
6.2 Introduction

This section describes the low-power modes.

6.3 Stop Mode

The STOP instruction places the MCU in its lowest-power consumption mode. In stop mode, the internal oscillator is turned off, halting all internal processing, including timer operation.

During the stop mode, the TCR bits are altered to remove any pending timer interrupt request and to disable any further timer interrupts. The timer prescaler is cleared. The I bit in the CCR is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the stop mode only by an external interrupt or reset. See **Figure 6-1**.



Notes:
1. Represents the internal gating of the OSC1 pin
2. IRQ pin edge-sensitive mask option
3. IRQ pin level and edge-sensitive mask option

Figure 6-1. Stop Recovery Timing Diagram

6.4 Wait Mode

The WAIT instruction places the MCU in a low-power consumption mode, but the wait mode consumes more power than the stop mode. All CPU action is suspended, but the timer, serial communications interface (SCI), serial peripheral interface (SPI), and the oscillator remain active. Any interrupt or reset will cause the MCU to exit the wait mode.

During wait mode, the I bit in the CCR is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer, SCI, and SPI may be enabled to allow a periodic exit from the wait mode.

Section 7. Input/Output (I/O) Ports

7.1 Contents

7.2 Introduction

This section briefly describes the 31 input/output (I/O) lines arranged as one 7-bit and three 8-bit ports. All of these port pins are programmable as either inputs or outputs under software control of the data direction registers.

NOTE: *To avoid a glitch on the output pins, write data to the I/O port data register before writing a one to the corresponding data direction register.*

7.3 Port A

Port A is an 8-bit bidirectional port which does not share any of its pins with other subsystems. The port A data register is at $0000 and the data direction register (DDR) is at $0004. The contents of the port A data register are indeterminate at initial powerup and must be initialized by user software. Reset does not affect the data registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a 1 to a DDR bit sets the corresponding port bit to output mode. A block diagram of the port logic is shown in **Figure 7-1**.

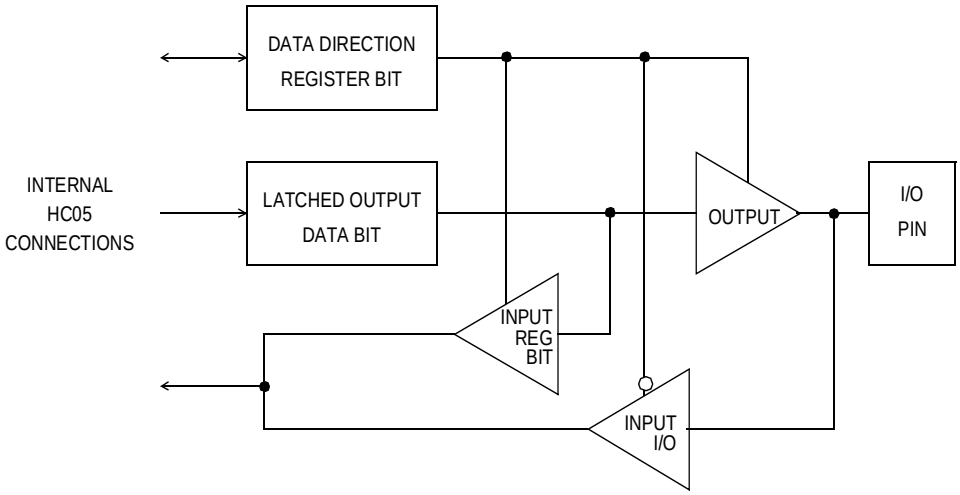


Figure 7-1. Port A I/O Circuit

7.4 Port B

Port B is an 8-bit bidirectional port. The port B data register is at $0001 and the data direction register (DDR) is at $0005. The contents of the port B data register are indeterminate at initial powerup and must be initialized by user software. Reset does not affect the data registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a 1 to a DDR bit sets the corresponding port pin to output mode. Each of the port B pins has an optional external interrupt capability that can be enabled by programming the corresponding bit in the port B mask option register ($3FF0).

The interrupt option also enables a pullup device when the pin is configured as an input. The edge or edge- and level-sensitivity of the $\overline{\text{IRQ}}$ pin will also pertain to the enabled port B pins. Care needs to be taken when using port B pins that have the pullup enabled. Before switching from an output to an input, the data should be preconditioned to a 1 to prevent an interrupt from occurring. The port B logic is shown in **Figure 7-2**.

7.5 Port C

Port C is an 8-bit bidirectional port. The port C data register is at $0002 and the data direction register (DDR) is at $0006. The contents of the port C data register are indeterminate at initial powerup and must be initialized by user software. Reset does not affect the data registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a 1 to a DDR bit sets the corresponding port bit to output mode. PC7 has a high current sink and source capability. **Figure 7-1** is also applicable to port C.

7.6 Port D

When configured as a C9A, port D is a 7-bit bidirectional port; when configured as a C12A, port D is a 7-bit fixed input port. Four of its pins are shared with the SPI subsystem and two more are shared with the SCI subsystem. The contents of the port D data register are indeterminate at initial powerup and must be initialized by user software. During reset all seven bits become valid input ports because the C9A DDR bits are cleared and the special function output drivers associated with the SCI and SPI subsystems are disabled, thereby returning the ports to inputs. Writing a 1 to a DDR bit sets the corresponding port bit to output mode only when configured as a C9A.

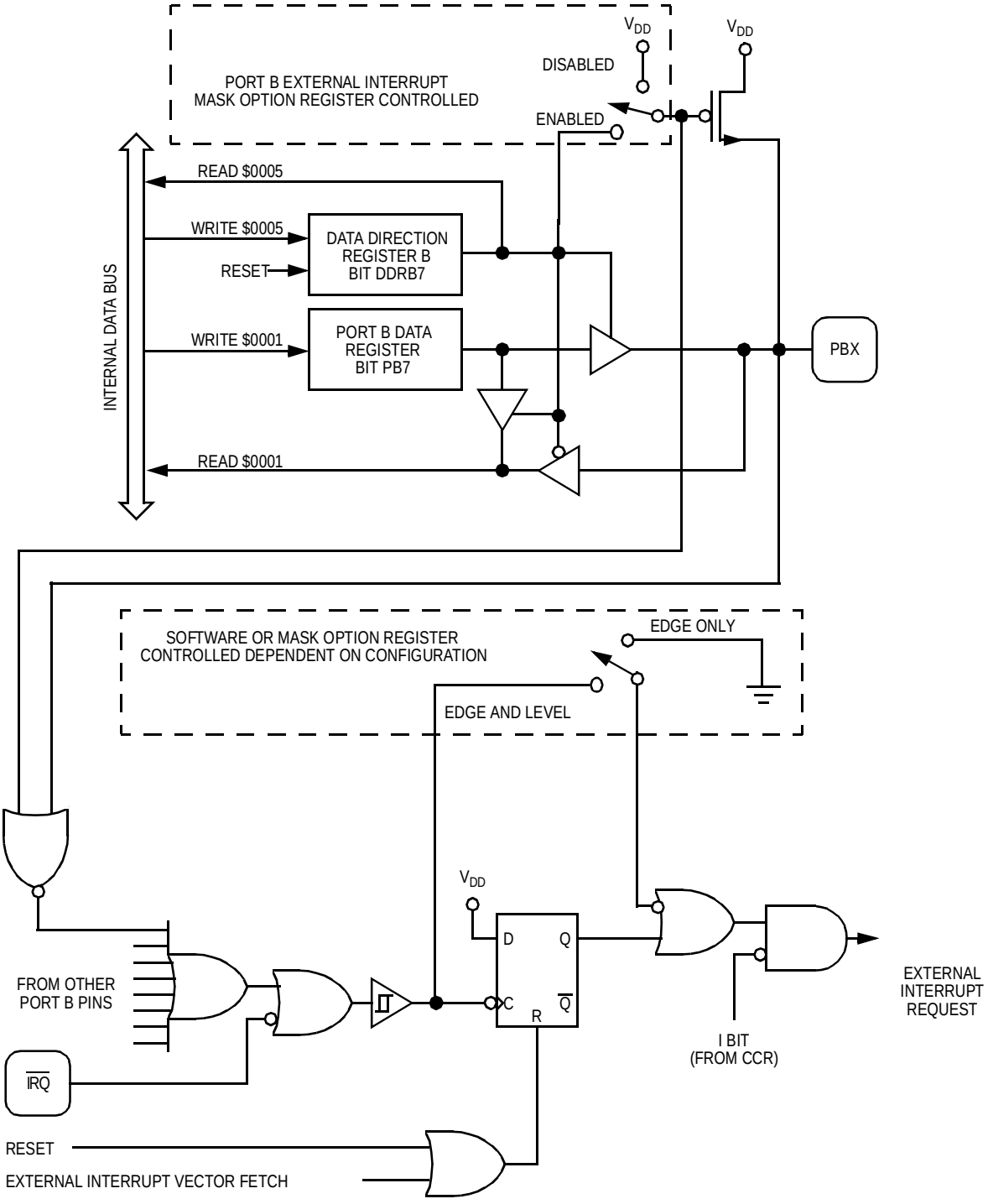


Figure 7-2. Port B I/O Logic

Section 8. Capture/Compare Timer

8.1 Content

8.2 Introduction

This section describes the operation of the 16-bit capture/compare timer. **Figure 8-1** shows the structure of the capture/compare subsystem.

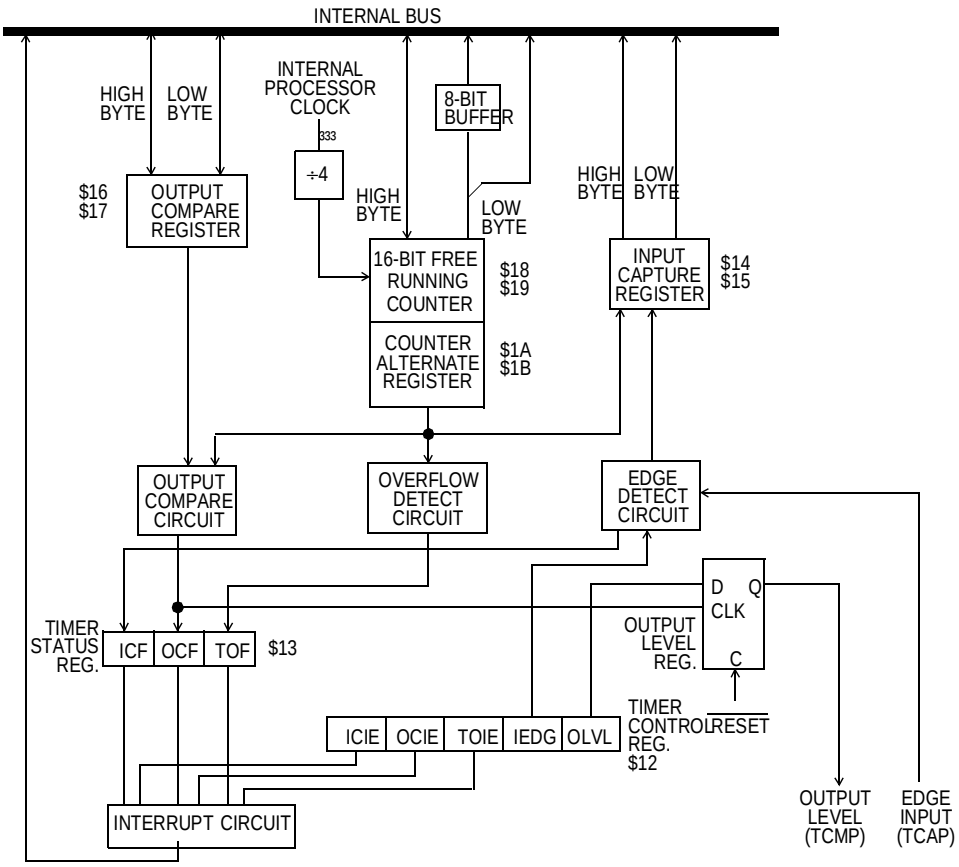


Figure 8-1. Capture/Compare Timer Block Diagram

8.3 Timer Operation

The core of the capture/compare timer is a 16-bit free-running counter. The counter provides the timing reference for the input capture and output compare functions. The input capture and output compare functions provide a means to latch the times at which external events occur, to measure input waveforms, and to generate output waveforms and timing delays. Software can read the value in the 16-bit free-running counter at any time without affecting the counter sequence.

Because of the 16-bit timer architecture, the I/O registers for the input capture and output compare functions are pairs of 8-bit registers.

Because the counter is 16 bits long and preceded by a fixed divide-by-4 prescaler, the counter rolls over every 262,144 internal clock cycles. Timer resolution with a 4-MHz crystal is 2 μs.

8.3.1 Input Capture

The input capture function is a means to record the time at which an external event occurs. When the input capture circuitry detects an active edge on the TCAP pin, it latches the contents of the timer registers into the input capture registers. The polarity of the active edge is programmable.

Latching values into the input capture registers at successive edges of the same polarity measures the period of the input signal on the TCAP pin. Latching values into the input capture registers at successive edges of opposite polarity measures the pulse width of the signal.

8.3.2 Output Compare

The output compare function is a means of generating an output signal when the 16-bit counter reaches a selected value. Software writes the selected value into the output compare registers. On every fourth internal clock cycle the output compare circuitry compares the value of the counter to the value written in the output compare registers. When a match occurs, the timer transfers the programmable output level bit (OLVL) from the timer control register to the TCMP pin.

The programmer can use the output compare register to measure time periods, to generate timing delays, or to generate a pulse of specific duration or a pulse train of specific frequency and duty cycle on the TCMP pin.

8.4 Timer I/O Registers

The following I/O registers control and monitor timer operation:

- Timer control register (TCR)
- Timer status register (TSR)
- Timer registers (TRH and TRL)
- Alternate timer registers (ATRH and ATRL)
- Input capture registers (ICRH and ICRL)
- Output compare registers (OCRH and OCRL)

8.4.1 Timer Control Register

The timer control register (TCR), shown in **Figure 8-2**, performs these functions:

- Enables input capture interrupts
- Enables output compare interrupts
- Enables timer overflow interrupts
- Controls the active edge polarity of the TCAP signal
- Controls the active level of the TCMP output

$0012	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
Write:								
Reset:	0	0	0	0	0	0	U	0

= Unimplemented U = Undetermined

Figure 8-2. Timer Control Register (TCR)

ICIE — Input Capture Interrupt Enable Bit

This read/write bit enables interrupts caused by an active signal on the TCAP pin. Resets clear 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. Resets clear 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 clear 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 register. Resets have 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 logic 0 appears on the TCMP pin when a successful output compare occurs. Resets clear the OLVL bit.

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

8.4.2 Timer Status Register

The timer status register (TSR), shown in **Figure 8-3**, contains flags to signal the following conditions:

- An active signal on the TCAP pin, transferring the contents of the timer registers to the input capture registers
- A match between the 16-bit counter and the output compare registers, transferring the OLVL bit to the TCMP pin
- A timer roll over from $FFFF to $0000

$0013	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ICF	OCF	TOF	0	0	0	0	0
Write:								
Reset:	U	U	U	0	0	0	0	0

= Unimplemented U = Undetermined

Figure 8-3. Timer Status Register (TSR)

ICF — Input Capture Flag

The ICF bit is set automatically when an edge of the selected polarity occurs on the TCAP pin. Clear the ICF bit by reading the timer status register with ICF set and then reading the low byte ($0015) of the input capture registers. Resets have no effect on ICF.

OCF — Output Compare Flag

The OCF bit is set automatically when the value of the timer registers matches the contents of the output compare registers. Clear the OCF bit by reading the timer status register with OCF set and then reading the low byte ($0017) of the output compare registers. Resets have no effect on OCF.

TOF — Timer Overflow Flag

The TOF bit is set automatically when the 16-bit counter rolls over from $FFFF to $0000. Clear the TOF bit by reading the timer status register with TOF set, and then reading the low byte ($0019) of the timer registers. Resets have no effect on TOF.

8.4.3 Timer Registers

The timer registers (TRH and TRL), shown in **Figure 8-4**, contains the current high and low bytes of the 16-bit counter. Reading TRH before reading TRL causes TRL to be latched until TRL is read. Reading TRL after reading the timer status register clears the timer overflow flag (TOF). Writing to the timer registers has no effect.

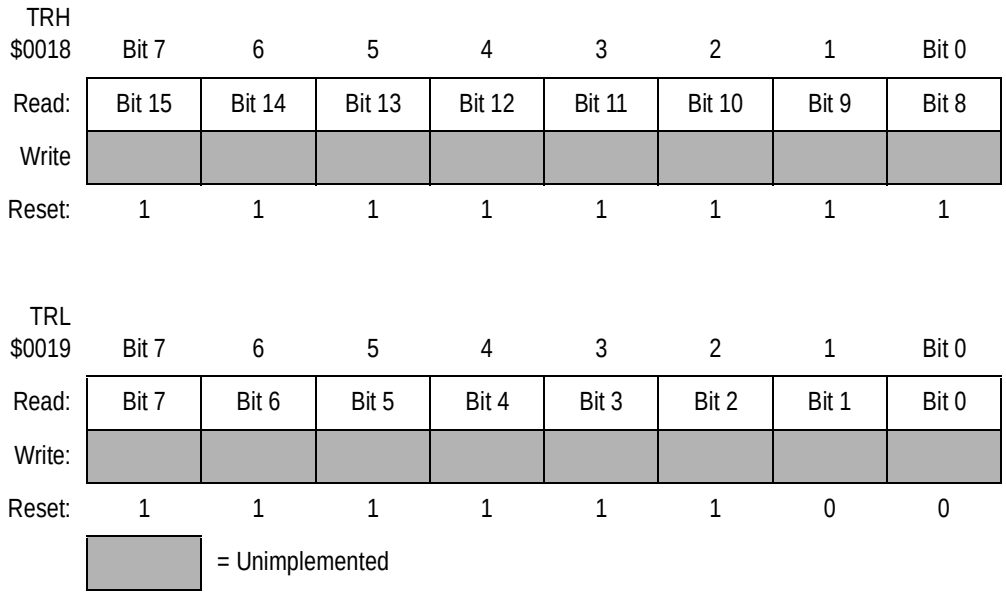


Figure 8-4. Timer Registers (TRH and TRL)

8.4.4 Alternate Timer Registers

The alternate timer registers (ATRH and ATRL), shown in **Figure 8-5**, contain the current high and low bytes of the 16-bit counter. Reading ATRH before reading ATRL causes ATRL to be latched until ATRL is read. Reading ATRL has no effect on the timer overflow flag (TOF). Writing to the alternate timer registers has no effect.

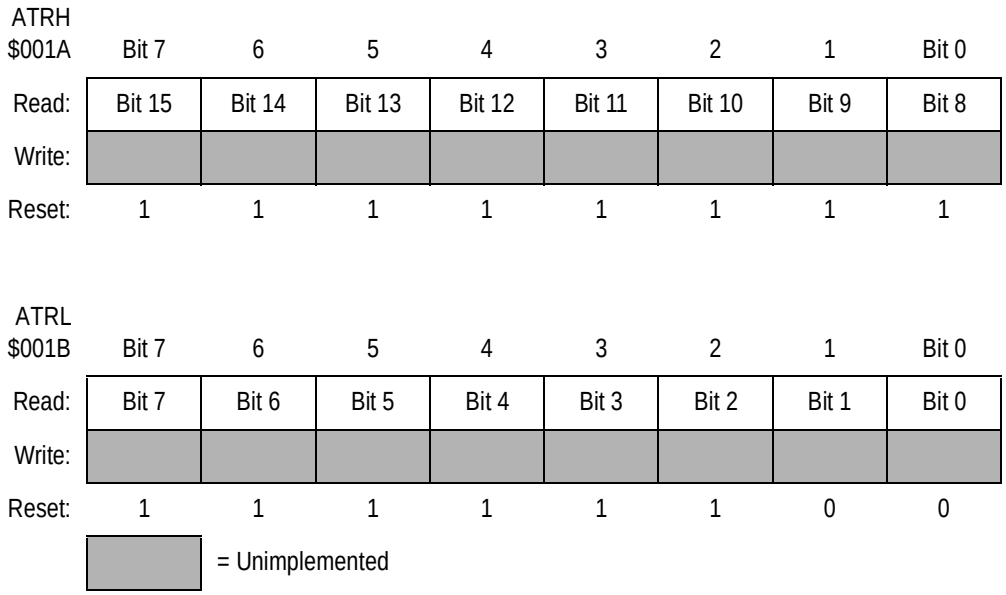


Figure 8-5. Alternate Timer Registers (ATRH and ATRL)

NOTE: *To prevent interrupts from occurring between readings of ATRH and ATRL, set the interrupt flag in the condition code register before reading ATRH, and clear the flag after reading ATRL.*

8.4.5 Input Capture Registers

When a selected edge occurs on the TCAP pin, the current high and low bytes of the 16-bit counter are latched into the input capture registers (ICRH and ICRL). Reading ICRH before reading ICRL inhibits further capture until ICRL is read. Reading ICRL after reading the status register clears the input capture flag (ICF). Writing to the input capture registers has no effect.

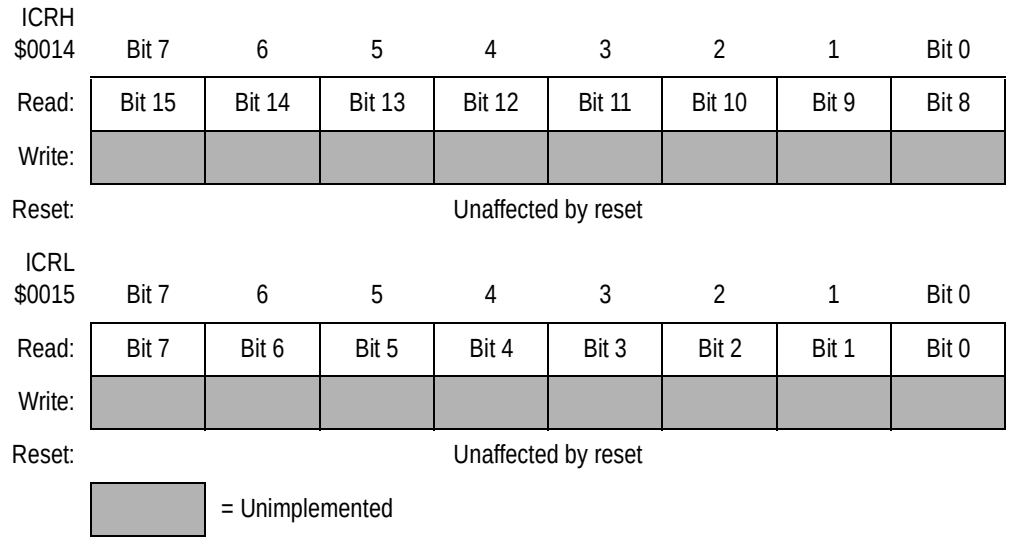


Figure 8-6. Input Capture Registers (ICRH and ICRL)

NOTE: *To prevent interrupts from occurring between readings of ICRH and ICRL, set the interrupt flag in the condition code register before reading ICRH, and clear the flag after reading ICRL.*

8.4.6 Output Compare Registers

When the value of the 16-bit counter matches the value in the output compare registers (OCRH and OCRL), the planned TCMP pin action takes place. Writing to OCRH before writing to OCRL inhibits timer compares until OCRL is written. Reading or writing to OCRL after the timer status register clears the output compare flag (OCF).

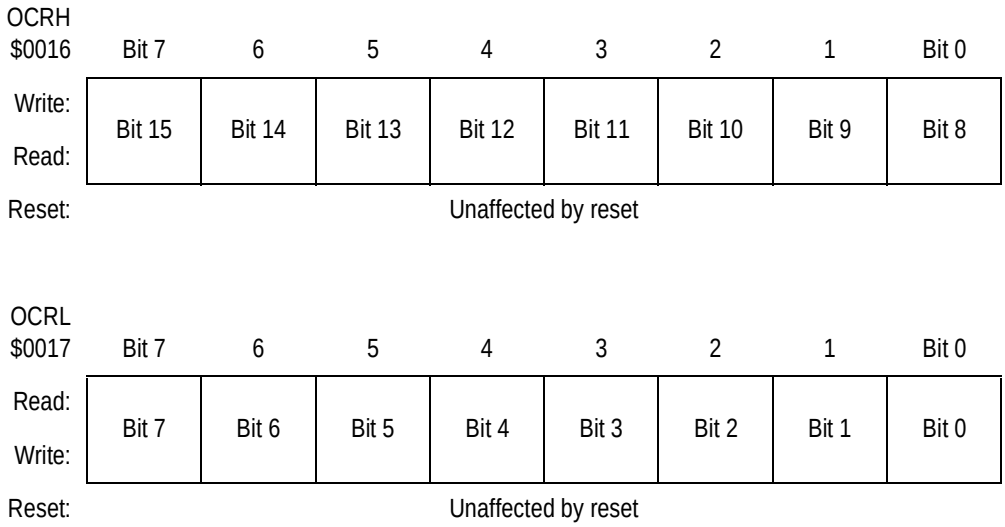


Figure 8-7. Output Compare Registers (OCRH and OCRL)

To prevent OCF from being set between the time it is read and the time the output compare registers are updated, use this procedure:

1. Disable interrupts by setting the I bit in the condition code register.
2. Write to OCRH. Compares are now inhibited until OCRL is written.
3. Clear bit OCF by reading timer status register (TSR).
4. Enable the output compare function by writing to OCRL.
5. Enable interrupts by clearing the I bit in the condition code register.

8.5 Timer During Wait Mode

The CPU clock halts during the wait mode, but the timer remains active. If interrupts are enabled, a timer interrupt will cause the processor to exit the wait mode.

8.6 Timer During Stop Mode

In the stop mode, the timer stops counting and holds the last count value if STOP is exited by an interrupt. If STOP is exited by reset, the counters are forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pins, the input capture detect circuit is armed. This does not set any timer flags or wake up the MCU, but if an interrupt is used to exit stop mode, there is an active input capture flag and data from the first valid edge that occurred during the stop mode. If reset is used to exit stop mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

Capture/Compare Timer

Section 9. Serial Communications Interface (SCI)

9.1 Content

9.2 Introduction

This section describes the on-chip asynchronous serial communications interface (SCI). The SCI allows full-duplex, asynchronous, RS232 or RS422 serial communication between the MCU and remote devices, including other MCUs. The transmitter and receiver of the SCI operate independently, although they use the same baud rate generator.

9.3 Features

Features of the SCI include:

- Standard mark/space non-return-to-zero format
- Full-duplex operation
- 32 programmable baud rates
- Programmable 8-bit or 9-bit character length
- Separately enabled transmitter and receiver
- Two receiver wakeup methods:
  - Idle line wakeup
  - Address mark wakeup
- Interrupt-driven operation capability with five interrupt flags:
  - Transmitter data register empty
  - Transmission complete
  - Transmission data register full
  - Receiver overrun
  - Idle receiver input
- Receiver framing error detection
- 1/16 bit-time noise detection

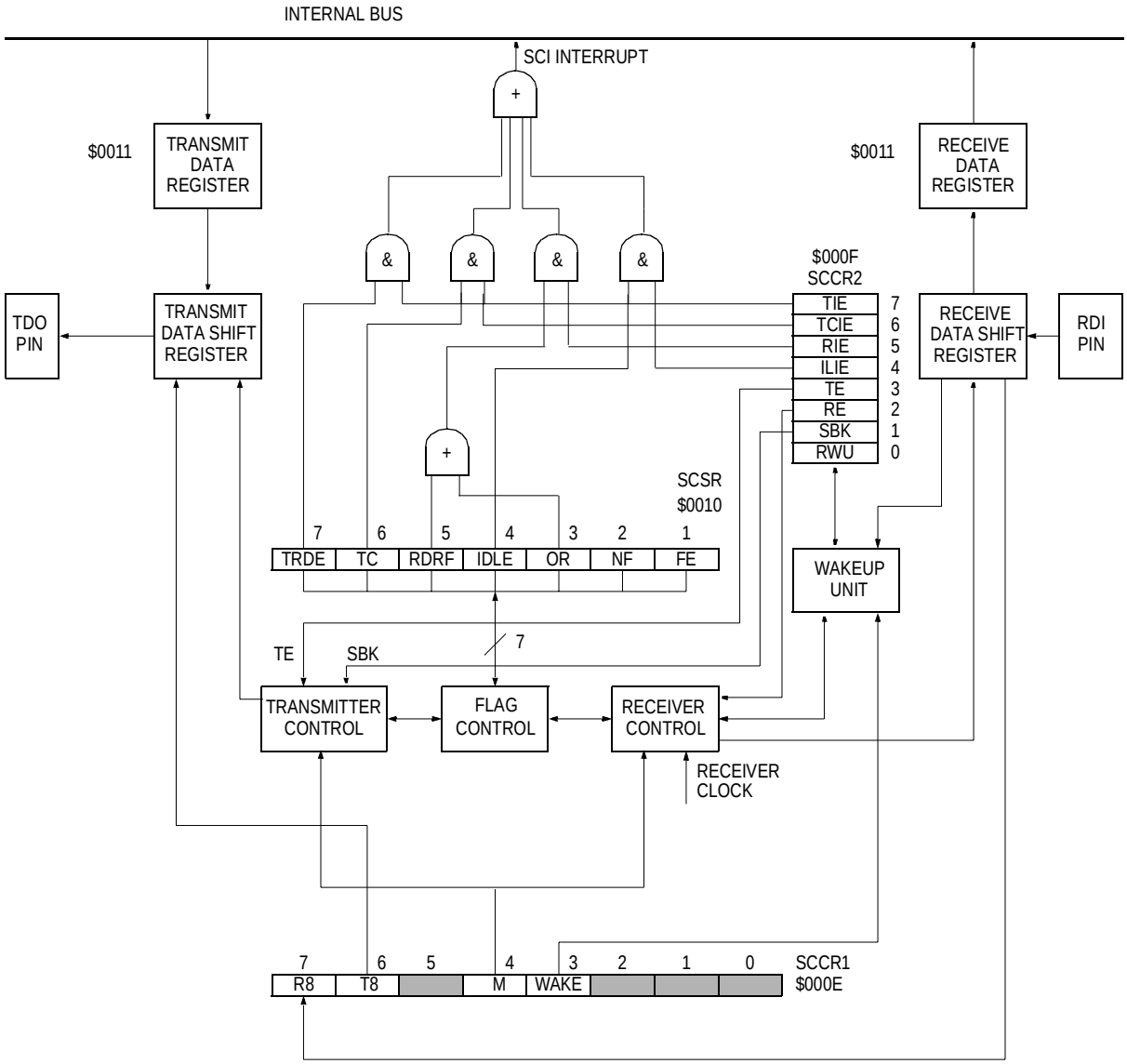


Figure 9-1. Serial Communications Interface Block Diagram

NOTE: *The serial communications data register (SCI SCDR) is controlled by the internal R/W signal. It is the transmit data register when written to and the receive data register when read.*

9.4 SCI Receiver Features

Features of the SCI receiver include:

- Receiver wakeup function (idle line or address bit)
- Idle line detection
- Framing error detection
- Noise detection
- Overrun detection
- Receiver data register full flag

9.5 SCI Transmitter Features

Features of the SCI transmitter include:

- Transmit data register empty flag
- Transmit complete flag
- Send break

9.6 Functional Description

A block diagram of the SCI is shown in **Figure 9-1**. Option bits in serial control register1 (SCCR1) select the wakeup method (WAKE bit) and data word length (M bit) of the SCI. SCCR2 provides control bits that individually enable the transmitter and receiver, enable system interrupts, and provide the wakeup enable bit (RWU) and the send break code bit (SBK). Control bits in the baud rate register (BAUD) allow the user to select one of 32 different baud rates for the transmitter and receiver.

Data transmission is initiated by writing to the serial communications data register (SCDR). Provided the transmitter is enabled, data stored in the SCDR is transferred to the transmit data shift register. This transfer of data sets the transmit data register empty flag (TDRE) in the SCI status register (SCSR) and generates an interrupt (if transmitter interrupts are enabled). The transfer of data to the transmit data shift

register is synchronized with the bit rate clock (see **Figure 9-2**). All data is transmitted least significant bit first. Upon completion of data transmission, the transmission complete flag (TC) in the SCSR is set (provided no pending data, preamble, or break is to be sent) and an interrupt is generated (if the transmit complete interrupt is enabled). If the transmitter is disabled, and the data, preamble, or break (in the transmit data shift register) has been sent, the TC bit will be set also. This will also generate an interrupt if the transmission complete interrupt enable bit (TCIE) is set. If the transmitter is disabled during a transmission, the character being transmitted will be completed before the transmitter gives up control of the TDO pin.

When SCDR is read, it contains the last data byte received, provided that the receiver is enabled. The receive data register full flag bit (RDRF) in the SCSR is set to indicate that a data byte has been transferred from the input serial shift register to the SCDR; this will cause an interrupt if the receiver interrupt is enabled. The data transfer from the input serial shift register to the SCDR is synchronized by the receiver bit rate clock. The OR (overrun), NF (noise), or FE (framing) error flags in the SCSR may be set if data reception errors occurred.

An idle line interrupt is generated if the idle line interrupt is enabled and the IDLE bit (which detects idle line transmission) in SCSR is set. This allows a receiver that is not in the wakeup mode to detect the end of a message, or the preamble of a new message, or to re-synchronize with the transmitter. A valid character must be received before the idle line condition or the IDLE bit will not be set and idle line interrupt will not be generated.

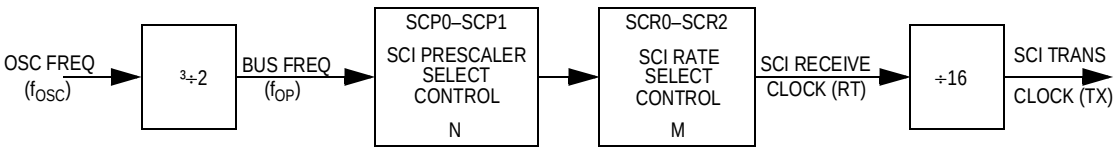


Figure 9-2. Rate Generator Division

9.7 Data Format

Receive data or transmit data is the serial data that is transferred to the internal data bus from the receive data input pin (RDI) or from the internal bus to the transmit data output pin (TDO). The non-return-to-zero (NRZ) data format shown in **Figure 9-3** is used and must meet the following criteria:

- The idle line is brought to a logic 1 state prior to transmission/reception of a character.
- A start bit (logic 0) is used to indicate the start of a frame.
- The data is transmitted and received least significant bit first.
- A stop bit (logic 1) is used to indicate the end of a frame. A frame consists of a start bit, a character of eight or nine data bits, and a stop bit.
- A break is defined as the transmission or reception of a low (logic 0) for at least one complete frame time.

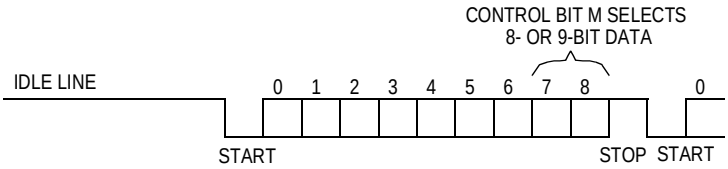


Figure 9-3. Data Format

9.8 Receiver Wakeup Operation

The receiver logic hardware also supports a receiver wakeup function which is intended for systems having more than one receiver. With this function a transmitting device directs messages to an individual receiver or group of receivers by passing addressing information as the initial byte(s) of each message. The wakeup function allows receivers not addressed to remain in a dormant state for the remainder of the unwanted message. This eliminates any further software overhead to service the remaining characters of the unwanted message and thus improves system performance.

The receiver is placed in wakeup mode by setting the receiver wakeup bit (RWU) in the SCCR2 register. While RWU is set, all of the receiver-related status flags (RDRF, IDLE, OR, NF, and FE) are inhibited (cannot become set).

NOTE: *The idle line detect function is inhibited while the RWU bit is set. Although RWU may be cleared by a software write to SCCR2, it would be unusual to do so.*

Normally, RWU is set by software and is cleared automatically in hardware by one of these methods: idle line wakeup or address mark wakeup.

9.9 Idle Line Wakeup

In idle line wakeup mode, a dormant receiver wakes up as soon as the RDI line becomes idle. Idle is defined as a continuous logic high level on the RDI line for 10 (or 11) full bit times. Systems using this type of wakeup must provide at least one character time of idle between messages to wake up sleeping receivers, but must not allow any idle time between characters within a message.

9.10 Address Mark Wakeup

In address mark wakeup, the most significant bit (MSB) in a character is used to indicate whether it is an address (logic 1) or data (logic 0) character. Sleeping receivers will wake up whenever an address character is received. Systems using this method for wakeup would set the MSB of the first character of each message and leave it clear for all other characters in the message. Idle periods may be present within messages and no idle time is required between messages for this wakeup method.

9.11 Receive Data In (RDI)

Receive data is the serial data that is applied through the input line and the SCI to the internal bus. The receiver circuitry clocks the input at a rate equal to 16 times the baud rate. This time is referred to as the RT rate in **Figure 9-4** and as the receiver clock in **Figure 9-6**.

The receiver clock generator is controlled by the baud rate register; however, the SCI is synchronized by the start bit, independent of the transmitter.

Once a valid start bit is detected, the start bit, each data bit, and the stop bit are sampled three times at RT intervals 8 RT, 9 RT, and 10 RT (1 RT is the position where the bit is expected to start), as shown in **Figure 9-5**. The value of the bit is determined by voting logic which takes the value of the majority of the samples. A noise flag is set when all three samples on a valid start bit or data bit or the stop bit do not agree.

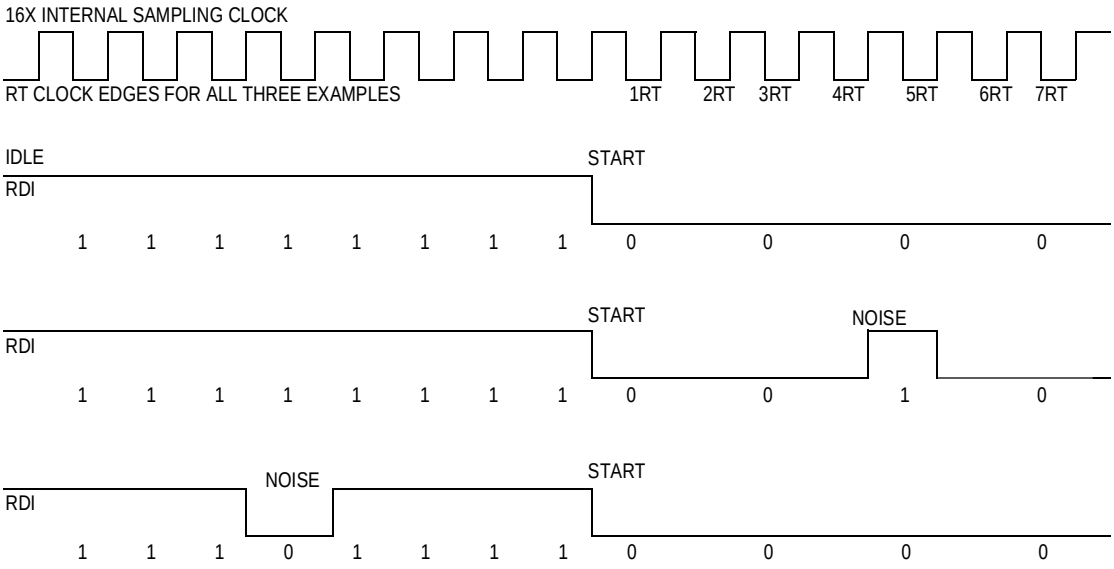


Figure 9-4. SCI Examples of Start Bit Sampling Techniques

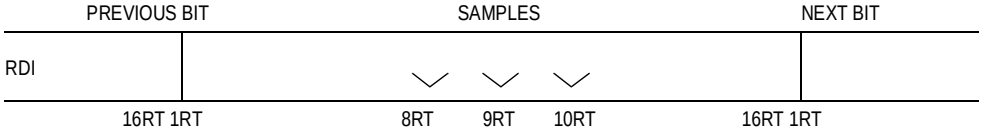


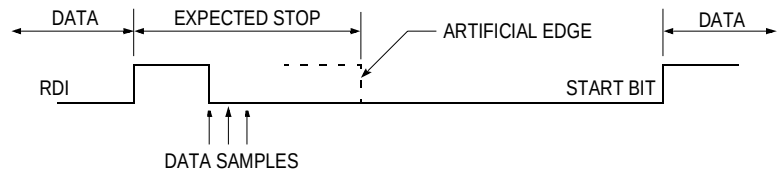
Figure 9-5. SCI Sampling Technique Used on All Bits

9.12 Start Bit Detection

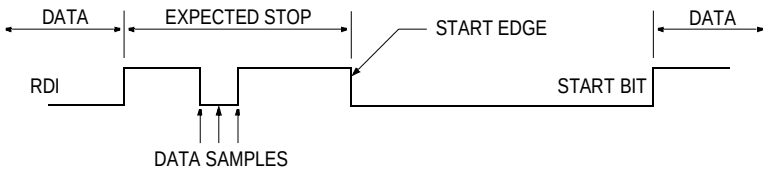
When the input (idle) line is detected low, it is tested for three more sample times (referred to as the start edge verification samples in **Figure 9-4**). If at least two of these three verification samples detect a logic 0, a valid start bit has been detected; otherwise, the line is assumed to be idle. A noise flag is set if all three verification samples do not detect a logic 0. Thus, a valid start bit could be assumed with a set noise flag present.

If a framing error has occurred without detection of a break (10 0s for 8-bit format or 11 0s for 9-bit format), the circuit continues to operate as if there actually was a stop bit, and the start edge will be placed artificially. The last bit received in the data shift register is inverted to a logic 1, and the three logic 1 start qualifiers (shown in **Figure 9-4**) are forced into the sample shift register during the interval when detection of a start bit is anticipated (see **Figure 9-6**); therefore, the start bit will be accepted no sooner than it is anticipated.

If the receiver detects that a break (RDRF = 1, FE = 1, receiver data register = $003B) produced the framing error, the start bit will not be artificially induced and the receiver must actually detect a logic 1 before the start bit can be recognized (see **Figure 9-7**).



a) Case 1: Receive line low during artificial edge



b) Case 2: Receive line high during expected start edge

Figure 9-6. SCI Artificial Start Following a Frame Error

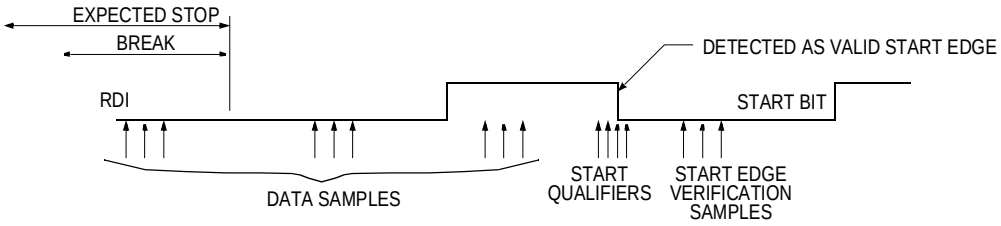


Figure 9-7. SCI Start Bit Following a Break

9.13 Transmit Data Out (TDO)

Transmit data is the serial data from the internal data bus that is applied through the SCI to the output line. Data format is as discussed in **9.7 Data Format** and shown in **Figure 9-3**. The transmitter generates a bit time by using a derivative of the RT clock, thus producing a transmission rate equal to 1/16th that of the receiver sample clock.

9.14 SCI I/O Registers

The following I/O registers control and monitor SCI operation:

- SCI data register (SCDR)
- SCI control register 1 (SCCR1)
- SCI control register 2 (SCCR2)
- SCI status register (SCSR)

9.14.1 SCI Data Register

The SCI data register (SCDR), shown in **Figure 9-8**, is the buffer for characters received and for characters transmitted.

$0011	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	BIT7	BIT6	BIT55	BIT4	BIT3	BIT2	BIT1	BIT0
Reset:	Unaffected by reset							

Figure 9-8. SCI Data Register (SCDR)

9.14.2 SCI Control Register 1

The SCI control register 1 (SCCR1), shown in **Figure 9-9**, has these functions:

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

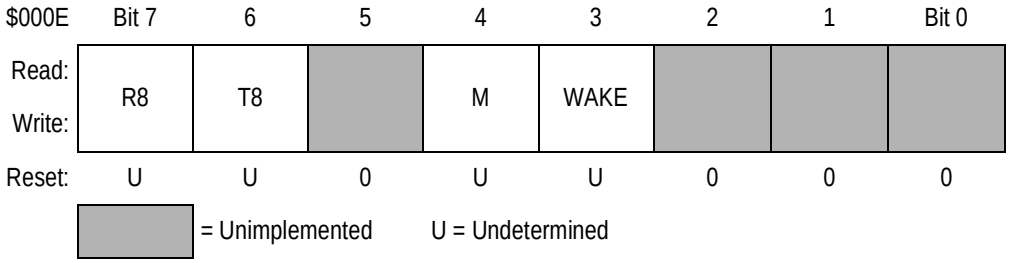


Figure 9-9. SCI Control Register 1 (SCCR1)

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. Resets have 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 the SCDR is loaded into the transmit register. Resets have no effect on the T8 bit.

M — Character Length Bit

This read/write bit determines whether SCI characters are 8 bits long or 9 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. Resets have no effect on the M bit.

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

WAKE — Wakeup Method Bit

This read/write bit determines which condition wakes up the SCI: a logic 1 (address mark) in the most significant bit (MSB) position of a received character or an idle condition on the PD0/RDI pin. Resets have no effect on the WAKE bit.

1 = Address mark wakeup
0 = Idle line wakeup

9.14.3 SCI Control Register 2

SCI control register 2 (SCCR2), shown in **Figure 9-10**, has these functions:

- Enables the SCI receiver and SCI receiver interrupts
- Enables the SCI transmitter and SCI transmitter interrupts
- Enables SCI receiver idle interrupts
- Enables SCI transmission complete interrupts
- Enables SCI wakeup
- Transmits SCI break characters

$000F	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset:	0	0	0	0	0	0	0	0

Figure 9-10. SCI Control Register 2 (SCCR2)

TIE — Transmit Interrupt Enable Bit

This read/write bit enables SCI interrupt requests when the TDRE flag becomes set. Resets clear the TIE bit.

1 = TDRE interrupt requests enabled
0 = TDRE interrupt requests disabled

TCIE — Transmission Complete Interrupt Enable Bit

This read/write bit enables SCI interrupt requests when the TC flag becomes set. Resets clear the TCIE bit.

1 = TC interrupt requests enabled
0 = TC interrupt requests disabled

RIE — Receiver Interrupt Enable Bit

This read/write bit enables SCI interrupt requests when the RDRF flag or the OR flag becomes set. Resets clear the RIE bit.

1 = RDRF interrupt requests enabled
0 = RDRF interrupt requests disabled

ILIE — Idle Line Interrupt Enable Bit

This read/write bit enables SCI interrupt requests when the IDLE bit becomes set. Resets clear the ILIE bit.

1 = IDLE interrupt requests enabled
0 = IDLE interrupt requests disabled

TE — Transmitter Enable Bit

Setting this read/write bit begins the transmission by sending a preamble of 10 or 11 logic 1s from the transmit shift register to the PD1/TDO pin. Resets clear the TE bit.

1 = Transmission enabled
0 = Transmission disabled

RE — Receiver Enable Bit

Setting this read/write bit enables the receiver. Clearing the RE bit disables the receiver and receiver interrupts but does not affect the receiver interrupt flags. Resets clear the RE bit.

1 = Receiver enabled
0 = Receiver disabled

RWU — Receiver Wakeup Enable Bit

This read/write bit puts the receiver in a standby state. Typically, data transmitted to the receiver clears the RWU bit and returns the receiver to normal operation. The WAKE bit in SCCR1 determines whether an idle input or an address mark brings the receiver out of standby state. Reset clears the RWU bit.

1 = Standby state
0 = Normal operation

SBK — Send Break Bit

Setting this read/write bit continuously transmits break codes in the form of 10-bit or 11-bit groups of logic 0s. Clearing the SBK bit stops the break codes and transmits a logic 1 as a start bit. Reset clears the SBK bit.

1 = Break codes being transmitted
0 = No break codes being transmitted

9.14.4 SCI Status Register

The SCI status register (SCSR), shown in **Figure 9-11**, contains flags to signal the following conditions:

- Transfer of SCDR data to transmit shift register complete
- Transmission complete
- Transfer of receive shift register data SCDR complete
- Receiver input idle
- Noisy data
- Framing error

$0010	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TDRE	TC	RDRF	IDLE	OR	NF	FE	
Write:								
Reset:	1	1	0	0	0	0	0	—

= Unimplemented

Figure 9-11. SCI Status Register (SCSR)

TDRE — Transmit Data Register Empty Flag

This clearable, read-only flag is set when the data in the SCDR transfers to the transmit shift register. TDRE generates an interrupt request if the TIE bit in SCCR2 is also set. Clear the TDRE bit by reading the SCSR with TDRE set and then writing to the SCDR. Reset sets the TDRE bit. Software must initialize the TDRE bit to logic 0 to avoid an instant interrupt request when turning the transmitter on.

1 = SCDR data transferred to transmit shift register
0 = SCDR data not transferred to transmit shift register

TC — Transmission Complete Flag

This clearable, read-only flag is set when the TDRE bit is set, and no data, preamble, or break character is being transmitted. TDRE generates an interrupt request if the TCIE bit in SCCR2 is also set. Clear the TC bit by reading the SCSR with TC set, and then writing to the SCDR. Reset sets the TC bit. Software must initialize the TC bit to logic 0 to avoid an instant interrupt request when turning the transmitter on.

1 = No transmission in progress
0 = Transmission in progress

RDRF — Receive Data Register Full Flag

This clearable, read-only flag is set when the data in the receive shift register transfers to the SCI data register. RDRF generates an interrupt request if the RIE bit in the SCCR2 is also set. Clear the RDRF bit by reading the SCSR with RDRF set and then reading the SCDR.

1 = Received data available in SCDR
0 = Received data not available in SCDR

IDLE — Receiver Idle Flag

This clearable, read-only flag is set when 10 or 11 consecutive logic 1s appear on the receiver input. IDLE generates an interrupt request if the ILIE bit in the SCCR2 is also set. Clear the ILIE bit by reading the SCSR with IDLE set and then reading the SCDR.

1 = Receiver input idle
0 = Receiver input not idle

OR — Receiver Overrun Flag

This clearable, read-only flag is set if the SCDR is not read before the receive shift register receives the next word. OR generates an interrupt request if the RIE bit in the SCCR2 is also set. The data in the shift register is lost, but the data already in the SCDR is not affected. Clear the OR bit by reading the SCSR with OR set and then reading the SCDR.

1 = Receive shift register full and RDRF = 1
0 = No receiver overrun

NF — Receiver Noise Flag

This clearable, read-only flag is set when noise is detected in data received in the SCI data register. Clear the NF bit by reading the SCSR and then reading the SCDR.

1 = Noise detected in SCDR
0 = No noise detected in SCDR

FE — Receiver Framing Error Flag

This clearable, read-only flag is set when there is a logic 0 where a stop bit should be in the character shifted into the receive shift register. If the received word causes both a framing error and an overrun error, the OR flag is set and the FE flag is not set. Clear the FE bit by reading the SCSR and then reading the SCDR.

1 = Framing error
0 = No framing error

9.14.5 Baud Rate Register

The baud rate register (BAUD), shown in **Figure 9-12**, selects the baud rate for both the receiver and the transmitter.

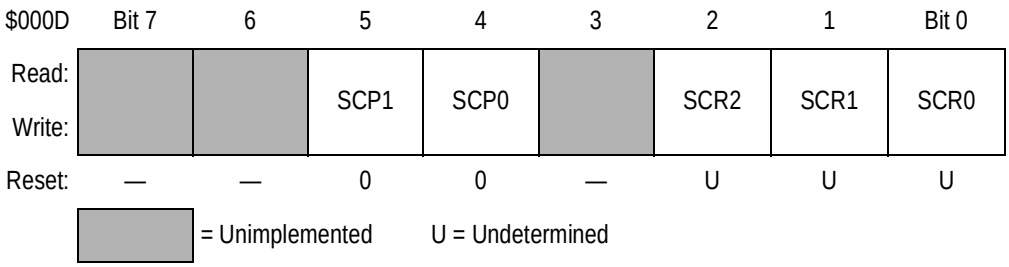


Figure 9-12. Baud Rate Register (BAUD)

SCP1 — SCP0–SCI Prescaler Select Bits

These read/write bits control prescaling of the baud rate generator clock, as shown in **Table 9-1**. Reset clears both SCP1 and SCP0.

Table 9-1. Baud Rate Generator Clock Prescaling

SCP[1:0]	Baud Rate Generator Clock
00	Internal Clock ÷ 1
01	Internal Clock ÷ 3
10	Internal Clock ÷ 4
11	Internal Clock ÷ 13

SCR2 — SCR0–SCI Baud Rate Select Bits

These read/write bits select the SCI baud rate, as shown in **Table 9-2**. Resets have no effect on the SCR2–SCR0 bits.

Table 9-2. Baud Rate Selection

SCR[2:0]	SCI Baud Rate (Baud)
000	Prescaled Clock ÷ 1
001	Prescaled Clock ÷ 2
010	Prescaled Clock ÷ 4
011	Prescaled Clock ÷ 8
100	Prescaled Clock ÷ 16
101	Prescaled Clock ÷ 32
110	Prescaled Clock ÷ 64
111	Prescaled Clock ÷ 128

Section 10. Serial Peripheral Interface (SPI)

10.1 Content

10.2 Introduction

The serial peripheral interface (SPI) is an interface built into the device which allows several MC68HC05 MCUs, or MC68HC05 MCU plus peripheral devices, to be interconnected within a single printed circuit board. In an SPI, separate wires are required for data and clock. In the SPI format, the clock is not included in the data stream and must be furnished as a separate signal. An SPI system may be configured in one containing one master MCU and several slave MCUs, or in a system in which an MCU is capable of being a master or a slave.

10.3 Features

Features include:

- Full-duplex, four-wire synchronous transfers
- Master or slave operation
- Bus frequency divided by 2 (maximum) master bit frequency
- Bus frequency (maximum) slave bit frequency
- Four programmable master bit rates
- Programmable clock polarity and phase
- End of transmission interrupt flag
- Write collision flag protection
- Master-master mode fault protection capability

10.4 SPI Signal Description

The four basic signals (MOSI, MISO, SCK, and $\overline{SS}$) are described in the following paragraphs. Each signal function is described for both the master and slave modes.

NOTE: *In C9A mode, any SPI output line has to have its corresponding data direction register bit set. If this bit is clear, the line is disconnected from the SPI logic and becomes a general-purpose input line. When the SPI is enabled, any SPI input line is forced to act as an input regardless of what is in the corresponding data direction register bit.*

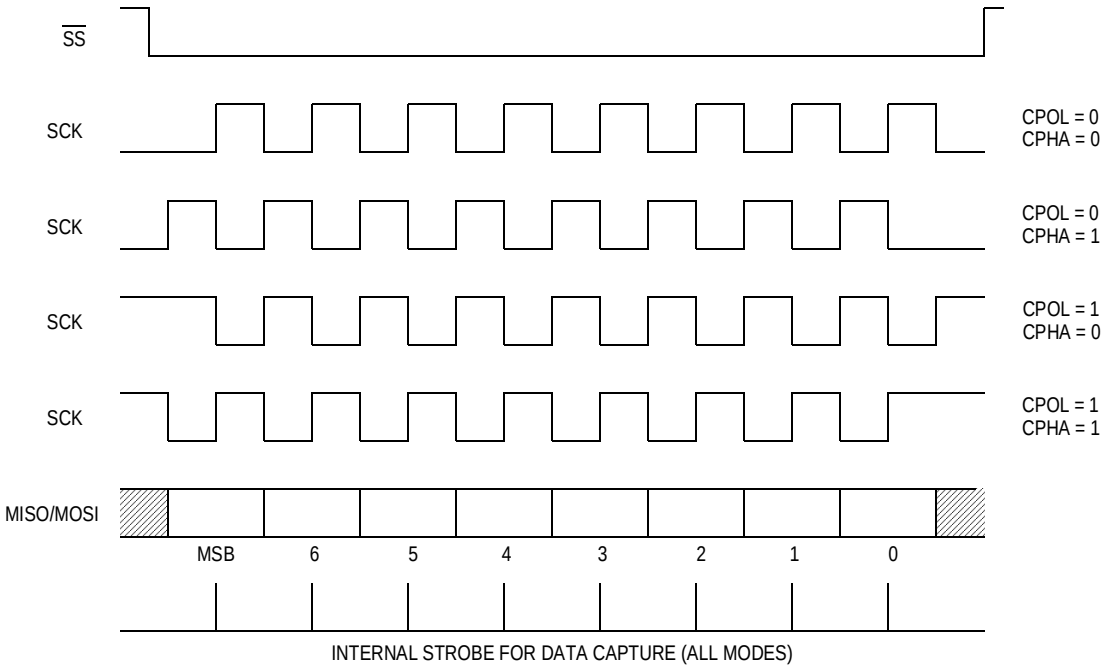


Figure 10-1. Data Clock Timing Diagram

10.4.1 Master In Slave Out (MISO)

The MISO line is configured as an input in a master device and as an output in a slave device. It is one of the two lines that transfer serial data in one direction, with the most significant bit sent first. The MISO line of a slave device is placed in the high-impedance state if the slave is not selected.

10.4.2 Master Out Slave In (MOSI)

The MOSI line is configured as an output in a master device and as an input in a slave device. It is one of the two lines that transfer serial data in one direction with the most significant bit sent first.

10.4.3 Serial Clock (SCK)

The master clock is used to synchronize data movement both in and out of the device through its MOSI and MISO lines. The master and slave devices are capable of exchanging a byte of information during a sequence of eight clock cycles. Since SCK is generated by the master device, this line becomes an input on a slave device.

As shown in **Figure 10-1**, four possible timing relationships may be chosen by using control bits CPOL and CPHA in the serial peripheral control register (SPCR). Both master and slave devices must operate with the same timing. The master device always places data on the MOSI line a half cycle before the clock edge (SCK), in order for the slave device to latch the data.

Two bits (SPR0 and SPR1) in the SPCR of the master device select the clock rate. In a slave device, SPR0 and SPR1 have no effect on the operation of the SPI.

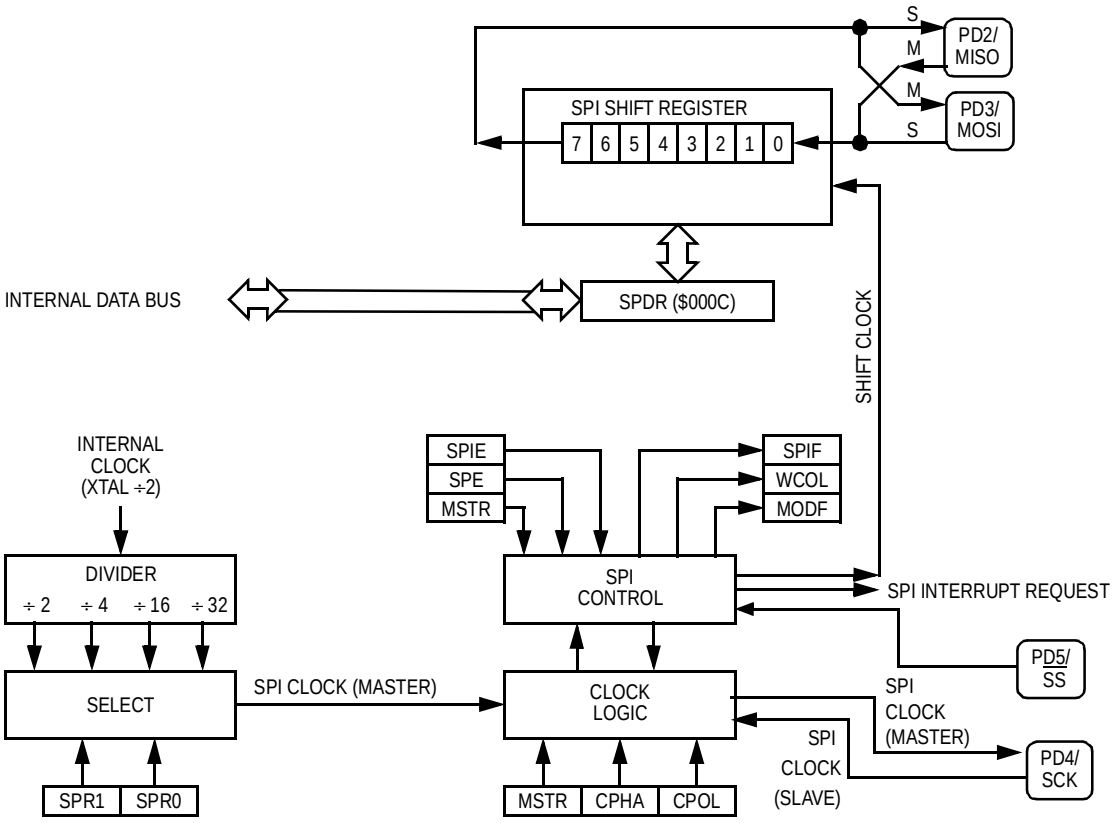
10.4.4 Slave Select ($\overline{SS}$)

The slave select ($\overline{SS}$) input line is used to select a slave device. It has to be low prior to data transactions and must stay low for the duration of the transaction.The $\overline{SS}$ line on the master must be tied high. In master mode, if the $\overline{SS}$ pin is pulled low during a transmission, a mode fault error flag (MODF) is set in the SPSR. In master mode the $\overline{SS}$ pin can be selected to be a general-purpose output (when configured as an MC68HC05C9A) by writing a 1 in bit 5 of the port D data direction register, thus disabling the mode fault circuit.

When CPHA = 0, the shift clock is the OR of $\overline{SS}$ with SCK. In this clock phase mode, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA = 1, $\overline{SS}$ may be left low for several SPI characters. In cases where there is only one SPI slave MCU, its $\overline{SS}$ line could be tied to V_{SS} as long as CPHA = 1 clock modes are used.

10.5 Functional Description

Figure 10-2 shows a block diagram of the serial peripheral interface circuitry. When a master device transmits data to a slave via the MOSI line, the slave device responds by sending data to the master device via the master's MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal. Thus, the byte transmitted is replaced by the byte received and eliminates the need for separate transmit-empty and receive-full status bits. A single status bit (SPIF) is used to signify that the I/O operation has been completed.



	7	6	5	4	3	2	1	0	
SPI CONTROL REGISTER (SPCR)	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR2	$000A
SPI STATUS REGISTER (SPSR)	SPIF	WCOL	0	MODF	0	0	0	0	$000B
SPI DATA REGISTER (SPDR)	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	$000C

Figure 10-2. Serial Peripheral Interface Block Diagram

The SPI is double buffered on read, but not on write. If a write is performed during data transfer, the transfer occurs uninterrupted, and the write will be unsuccessful. This condition will cause the write collision (WCOL) status bit in the SPSR to be set. After a data byte is shifted, the SPIF flag of the SPSR is set.

In the master mode, the SCK pin is an output. It idles high or low, depending on the CPOL bit in the SPCR, until data is written to the shift register, at which point eight clocks are generated to shift the eight bits of data and then SCK goes idle again.

In a slave mode, the slave select start logic receives a logic low at the $\overline{SS}$ pin and a clock at the SCK pin. Thus, the slave is synchronized with the master. Data from the master is received serially at the MOSI line and loads the 8-bit shift register. After the 8-bit shift register is loaded, its data is parallel transferred to the read buffer. During a write cycle, data is written into the shift register, then the slave waits for a clock train from the master to shift the data out on the slave's MISO line.

Figure 10-3 illustrates the MOSI, MISO, SCK, and $\overline{SS}$ master-slave interconnections.

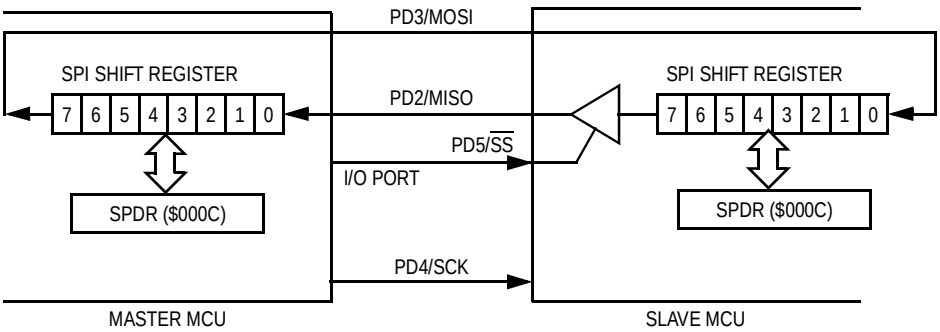


Figure 10-3. Serial Peripheral Interface Master-Slave Interconnection

10.6 SPI Registers

Three registers in the SPI provide control, status, and data storage functions. These registers are called the serial peripheral control register (SPCR), serial peripheral status register (SPSR), and serial peripheral data I/O register (SPDR) and are described in the following paragraphs.

10.6.1 Serial Peripheral Control Register

The SPI control register (SPCR), shown in **Figure 10-4**, controls these functions:

- Enables SPI interrupts
- Enables the SPI system
- Selects between standard CMOS or open drain outputs for port D (C9A mode only)
- Selects between master mode and slave mode
- Controls the clock/data relationship between master and slave
- Determines the idle level of the clock pin

$000A	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SPIE	SPE	DWOM (C9A)	MSTR	CPOL	CPHA	SPR1	SPR0
Reset:	0	0	0	0	0	1	U	U

U = Undetermined

Figure 10-4. SPI Control Register (SPCR)

SPIE — Serial Peripheral Interrupt Enable Bit

This read/write bit enables SPI interrupts. Reset clears the SPIE bit.

1 = SPI interrupts enabled
0 = SPI interrupts disabled

SPE — Serial Peripheral System Enable Bit

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

1 = SPI system enabled
0 = SPI system disabled

DWOM — Port D Wire-OR Mode Option Bit

This read/write bit disables the high side driver transistors on port D outputs so that port D outputs become open-drain drivers. DWOM affects all seven port D pins together. This option is only available when configured as a C9A.

1 = Port D outputs act as open-drain outputs.
0 = Port D outputs are normal CMOS outputs.

MSTR — Master Mode Select 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

When the clock polarity bit is cleared and data is not being transferred, a steady state low value is produced at the SCK pin of the master device. Conversely, if this bit is set, the SCK pin will idle high. This bit is also used in conjunction with the clock phase control bit to produce the desired clock-data relationship between master and slave. See **Figure 10-1**.

CPHA — Clock Phase Bit

The clock phase bit, in conjunction with the CPOL bit, controls the clock-data relationship between master and slave. The CPOL bit can be thought of as simply inserting an inverter in series with the SCK line. The CPHA bit selects one of two fundamentally different clocking protocols. When CPHA = 0, the shift clock is the OR of SCK with $\overline{SS}$. As soon as $\overline{SS}$ goes low, the transaction begins and the first edge on SCK invokes the first data sample. When CPHA=1, the $\overline{SS}$ pin may be thought of as a simple output enable control. See **Figure 10-1**.

SPR1 and SPR0 — SPI Clock Rate Selects

These read/write bits select one of four master mode serial clock rates, as shown in **Table 10-1**. They have no effect in the slave mode.

Table 10-1. 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

10.6.2 Serial Peripheral Status Register

The SPI status register (SPSR), shown in **Figure 10-5**, contains flags to signal the following conditions:

- SPI transmission complete
- Write collision
- Mode fault

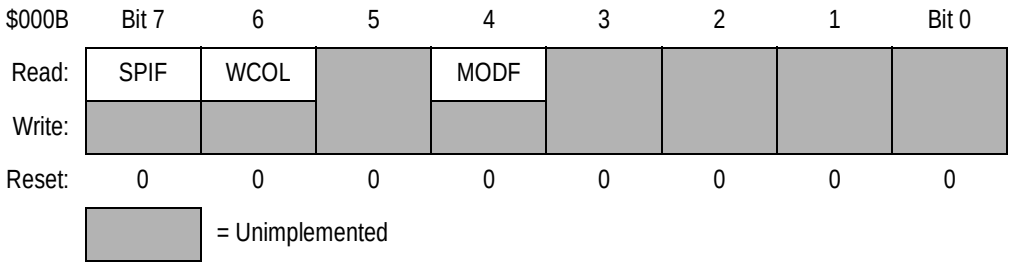


Figure 10-5. SPI Status Register

SPIF — SPI Transfer Complete Flag

The serial peripheral data transfer flag bit is set upon completion of data transfer between the processor and external device. If SPIF goes high, and if SPIE is set, a serial peripheral interrupt is generated. Clearing the SPIF bit is accomplished by reading the SPSR (with

SPIF set) followed by an access of the SPDR. Following the initial transfer, unless SPSR is read (with SPIF set) first, attempts to write to SPDR are inhibited.

WCOL — Write Collision Bit

The write collision bit is set when an attempt is made to write to the serial peripheral data register while data transfer is taking place. If CPHA is 0, a transfer is said to begin when $\overline{SS}$ goes low and the transfer ends when $\overline{SS}$ goes high after eight clock cycles on SCK. When CPHA is 1, a transfer is said to begin the first time SCK becomes active while $\overline{SS}$ is low and the transfer ends when the SPIF flag gets set. Clearing the WCOL bit is accomplished by reading the SPSR (with WCOL set) followed by an access to SPDR.

MODF — Mode Fault

The mode fault flag indicates that there may have been a multi-master conflict for system control and allows a proper exit from system operation to a reset or default system state. The MODF bit is normally clear, and is set only when the master device has its $\overline{SS}$ pin pulled low. Setting the MODF bit affects the internal serial peripheral interface system in the following ways.

1. An SPI interrupt is generated if SPIE = 1.
2. The SPE bit is cleared. This disables the SPI.
3. The MSTR bit is cleared, thus forcing the device into the slave mode.

Clearing the MODF bit is accomplished by reading the SPSR (with MODF set), followed by a write to the SPCR. Control bits SPE and MSTR may be restored by user software to their original state during this clearing sequence or after the MODF bit has been cleared. When configured as an MC68HC05C9A, it is also necessary to restore DDRD after a mode fault.

Bits 5 and 3–0 — Not Implemented

These bits always read 0.

10.6.3 Serial Peripheral Data I/O Register

The serial peripheral data I/O register (SPDR), shown in **Figure 10-6**, is used to transmit and receive data on the serial bus. Only a write to this register will initiate transmission/reception of another byte and this will only occur in the master device. At the completion of transmitting a byte of data, the SPIF status bit is set in both the master and slave devices.

When the user reads the serial peripheral data I/O register, a buffer is actually being read. The first SPIF must be cleared by the time a second transfer of the data from the shift register to the read buffer is initiated or an overrun condition will exist. In cases of overrun, the byte which causes the overrun is lost.

A write to the serial peripheral data I/O register is not buffered and places data directly into the shift register for transmission.

$000C	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
Reset:	Unaffected by reset							

Figure 10-6. PI Data Register (SPDR)

Section 11. Instruction Set

11.1 Contents

11.2 Introduction

The MCU instruction set has 62 instructions and uses eight addressing modes. The instructions include all those of the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. The MUL instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is

stored in the index register, and the low-order product is stored in the accumulator.

11.3 Addressing Modes

The CPU uses eight addressing modes for flexibility in accessing data. The addressing modes provide eight different ways for the CPU to find the data required to execute an instruction. The eight addressing modes are:

- Inherent
- Immediate
- Direct
- Extended
- Indexed, no offset
- Indexed, 8-bit offset
- Indexed, 16-bit offset
- Relative

11.3.1 Inherent

Inherent instructions are those that have no operand, such as return from interrupt (RTI) and stop (STOP). Some of the inherent instructions act on data in the CPU registers, such as set carry flag (SEC) and increment accumulator (INCA). Inherent instructions require no operand address and are one byte long.

11.3.2 Immediate

Immediate instructions are those that contain a value to be used in an operation with the value in the accumulator or index register. Immediate instructions require no operand address and are two bytes long. The opcode is the first byte, and the immediate data value is the second byte.

11.3.3 Direct

Direct instructions can access any of the first 256 memory locations with two bytes. The first byte is the opcode, and the second is the low byte of the operand address. In direct addressing, the CPU automatically uses $00 as the high byte of the operand address.

11.3.4 Extended

Extended instructions use three bytes and can access any address in memory. The first byte is the opcode; the second and third bytes are the high and low bytes of the operand address.

When using the Motorola assembler, the programmer does not need to specify whether an instruction is direct or extended. The assembler automatically selects the shortest form of the instruction.

11.3.5 Indexed, No Offset

Indexed instructions with no offset are 1-byte instructions that can access data with variable addresses within the first 256 memory locations. The index register contains the low byte of the effective address of the operand. The CPU automatically uses $00 as the high byte, so these instructions can address locations $0000–$00FF.

Indexed, no offset instructions are often used to move a pointer through a table or to hold the address of a frequently used RAM or I/O location.

11.3.6 Indexed, 8-Bit Offset

Indexed, 8-bit offset instructions are 2-byte instructions that can access data with variable addresses within the first 511 memory locations. The CPU adds the unsigned byte in the index register to the unsigned byte following the opcode. The sum is the effective address of the operand. These instructions can access locations $0000–$01FE.

Indexed 8-bit offset instructions are useful for selecting the kth element in an n-element table. The table can begin anywhere within the first 256 memory locations and could extend as far as location 510 ($01FE). The k value is typically in the index register, and the address of the beginning of the table is in the byte following the opcode.

11.3.7 Indexed,16-Bit Offset

Indexed, 16-bit offset instructions are 3-byte instructions that can access data with variable addresses at any location in memory. The CPU adds the unsigned byte in the index register to the two unsigned bytes following the opcode. The sum is the effective address of the operand. The first byte after the opcode is the high byte of the 16-bit offset; the second byte is the low byte of the offset.

Indexed, 16-bit offset instructions are useful for selecting the kth element in an n-element table anywhere in memory.

As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

11.3.8 Relative

Relative addressing is only for branch instructions. If the branch condition is true, the CPU finds the effective branch destination by adding the signed byte following the opcode to the contents of the program counter. If the branch condition is not true, the CPU goes to the next instruction. The offset is a signed, two’s complement byte that gives a branching range of –128 to +127 bytes from the address of the next location after the branch instruction.

When using the Motorola assembler, the programmer does not need to calculate the offset, because the assembler determines the proper offset and verifies that it is within the span of the branch.

11.4 Instruction Types

The MCU instructions fall into the following five categories:

- Register/memory instructions
- Read-modify-write instructions
- Jump/branch instructions
- Bit manipulation instructions
- Control instructions

11.4.1 Register/Memory Instructions

These instructions operate on CPU registers and memory locations. Most of them use two operands. One operand is in either the accumulator or the index register. The CPU finds the other operand in memory.

Table 11-1. Register/Memory Instructions

Instruction	Mnemonic
Add Memory Byte and Carry Bit to Accumulator	ADC
Add Memory Byte to Accumulator	ADD
AND Memory Byte with Accumulator	AND
Bit Test Accumulator	BIT
Compare Accumulator	CMP
Compare Index Register with Memory Byte	CPX
EXCLUSIVE OR Accumulator with Memory Byte	EOR
Load Accumulator with Memory Byte	LDA
Load Index Register with Memory Byte	LDX
Multiply	MUL
OR Accumulator with Memory Byte	ORA
Subtract Memory Byte and Carry Bit from Accumulator	SBC
Store Accumulator in Memory	STA
Store Index Register in Memory	STX
Subtract Memory Byte from Accumulator	SUB

11.4.2 Read-Modify-Write Instructions

These instructions read a memory location or a register, modify its contents, and write the modified value back to the memory location or to the register.

NOTE: *Do not use read modify-write operations on write-only registers.*

Table 11-2. Read-Modify-Write Instructions

Instruction	Mnemonic
Arithmetic Shift Left (Same as LSL)	ASL
Arithmetic Shift Right	ASR
Bit Clear	BCLR(1)
Bit Set	BSET(1)
Clear Register	CLR
Complement (One's Complement)	COM
Decrement	DEC
Increment	INC
Logical Shift Left (Same as ASL)	LSL
Logical Shift Right	LSR
Negate (Two's Complement)	NEG
Rotate Left through Carry Bit	ROL
Rotate Right through Carry Bit	ROR
Test for Negative or Zero	TST(2)

1. Unlike other read-modify-write instructions, BCLR and BSET use only direct addressing.
2. TST is an exception to the read-modify-write sequence because it does not write a replacement value.

11.4.3 Jump/Branch Instructions

Jump instructions allow the CPU to interrupt the normal sequence of the program counter. The unconditional jump instruction (JMP) and the jump-to-subroutine instruction (JSR) have no register operand. Branch instructions allow the CPU to interrupt the normal sequence of the program counter when a test condition is met. If the test condition is not met, the branch is not performed.

The BRCLR and BRSET instructions cause a branch based on the state of any readable bit in the first 256 memory locations. These 3-byte instructions use a combination of direct addressing and relative addressing. The direct address of the byte to be tested is in the byte following the opcode. The third byte is the signed offset byte. The CPU finds the effective branch destination by adding the third byte to the program counter if the specified bit tests true. The bit to be tested and its condition (set or clear) is part of the opcode. The span of branching is from –128 to +127 from the address of the next location after the branch instruction. The CPU also transfers the tested bit to the carry/borrow bit of the condition code register.

Table 11-3. Jump and Branch Instructions

Instruction	Mnemonic
Branch if Carry Bit Clear	BCC
Branch if Carry Bit Set	BCS
Branch if Equal	BEQ
Branch if Half-Carry Bit Clear	BHCC
Branch if Half-Carry Bit Set	BHCS
Branch if Higher	BHI
Branch if Higher or Same	BHS
Branch if $\overline{\text{IRQ}}$ Pin High	BIH
Branch if $\overline{\text{IRQ}}$ Pin Low	BIL
Branch if Lower	BLO
Branch if Lower or Same	BLS
Branch if Interrupt Mask Clear	BMC
Branch if Minus	BMI
Branch if Interrupt Mask Set	BMS
Branch if Not Equal	BNE
Branch if Plus	BPL
Branch Always	BRA
Branch if Bit Clear	BRCLR
Branch Never	BRN
Branch if Bit Set	BRSET
Branch to Subroutine	BSR
Unconditional Jump	JMP
Jump to Subroutine	JSR

11.4.4 Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory, which includes I/O registers and on-chip RAM locations. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations.

Table 11-4. Bit Manipulation Instructions

Instruction	Mnemonic
Bit Clear	BCLR
Branch if Bit Clear	BRCLR
Branch if Bit Set	BRSET
Bit Set	BSET

11.4.5 Control Instructions

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

Table 11-5. 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

11.5 Instruction Set Summary

Table 11-6. Instruction Set Summary (Sheet 1 of 6)

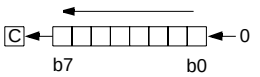
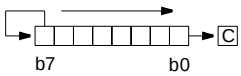
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr* ADD *opr* ADD *opr*,X ADD *opr*,X ADD ,X	Add without Carry	A ← (A) + (M)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	AB BB CB DB EB FB	ii dd hh ll ee ff ff	2 3 4 5 4 3
AND #*opr* AND *opr* AN*D opr* AND *opr*,X AND *opr*,X AND ,X	Logical AND	A ← (A) ∧ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A4 B4 C4 D4 E4 F4	ii dd hh ll ee ff ff	2 3 4 5 4 3
ASL *opr* ASLA ASLX ASL *opr*,X ASL ,X	Arithmetic Shift Left (Same as LSL)		—	—	↕	↕	↕	DIR INH INH IX1 IX	38 48 58 68 78	dd ff 	5 3 3 6 5
ASR *opr* ASRA ASRX ASR *opr*,X ASR ,X	Arithmetic Shift Right		—	—	↕	↕	↕	DIR INH INH IX1 IX	37 47 57 67 77	dd ff 	5 3 3 6 5
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BCLR *n opr*	Clear Bit n	Mn ← 0	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + 2 + *rel* ? Z = 1	—	—	—	—	—	REL	27	rr	3
BHCC *rel*	Branch if Half-Carry Bit Clear	PC ← (PC) + 2 + *rel* ? H = 0	—	—	—	—	—	REL	28	rr	3
BHCS *rel*	Branch if Half-Carry Bit Set	PC ← (PC) + 2 + *rel* ? H = 1	—	—	—	—	—	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? C ∨ Z = 0	—	—	—	—	—	REL	22	rr	3
BHS *rel*	Branch if Higher or Same	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3

Table 11-6. Instruction Set Summary (Sheet 2 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BIH *rel*	Branch if IRQ Pin High	PC ← (PC) + 2 + *rel* ? IRQ = 1	—	—	—	—	—	REL	2F	rr	3
BIL *rel*	Branch if IRQ Pin Low	PC ← (PC) + 2 + *rel* ? IRQ = 0	—	—	—	—	—	REL	2E	rr	3
BIT #*opr* BIT *opr* BIT *opr* BIT *opr*,X BIT *opr*,X BIT ,X	Bit Test Accumulator with Memory Byte	(A) ∧ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A5 B5 C5 D5 E5 F5	ii dd hh ll ee ff ff	2 3 4 5 4 3
BLO *rel*	Branch if Lower (Same as BCS)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	PC ← (PC) + 2 + *rel* ? C ∨ Z = 1	—	—	—	—	—	REL	23	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	PC ← (PC) + 2 + *rel* ? I = 0	—	—	—	—	—	REL	2C	rr	3
BMI *rel*	Branch if Minus	PC ← (PC) + 2 + *rel* ? N = 1	—	—	—	—	—	REL	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	PC ← (PC) + 2 + *rel* ? I = 1	—	—	—	—	—	REL	2D	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + 2 + *rel* ? Z = 0	—	—	—	—	—	REL	26	rr	3
BPL *rel*	Branch if Plus	PC ← (PC) + 2 + *rel* ? N = 0	—	—	—	—	—	REL	2A	rr	3
BRA *rel*	Branch Always	PC ← (PC) + 2 + *rel* ? 1 = 1	—	—	—	—	—	REL	20	rr	3
BRCLR *n opr rel*	Branch if Bit n Clear	PC ← (PC) + 2 + *rel* ? Mn = 0	—	—	—	—	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN *rel*	Branch Never	PC ← (PC) + 2 + *rel* ? 1 = 0	—	—	—	—	—	REL	21	rr	3
BRSET *n opr rel*	Branch if Bit n Set	PC ← (PC) + 2 + *rel* ? Mn = 1	—	—	—	—	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BSET *n opr*	Set Bit n	Mn ← 1	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BSR *rel*	Branch to Subroutine	PC ← (PC) + 2; push (PCL) SP ← (SP) – 1; push (PCH) SP ← (SP) – 1 PC ← (PC) + *rel*	—	—	—	—	—	REL	AD	rr	6
CLC	Clear Carry Bit	C ← 0	—	—	—	—	0	INH	98		2
CLI	Clear Interrupt Mask	I ← 0	—	0	—	—	—	INH	9A		2

Table 11-6. Instruction Set Summary (Sheet 3 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
CLR *opr* CLRA CLRX CLR *opr*,X CLR ,X	Clear Byte	M ← $00 A ← $00 X ← $00 M ← $00 M ← $00	—	—	0	1	—	DIR INH INH IX1 IX	3F 4F 5F 6F 7F	dd ff 	5 3 3 6 5
CMP #*opr* CMP *opr* CMP *opr* CMP *opr*,X CMP *opr*,X CMP ,X	Compare Accumulator with Memory Byte	(A) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A1 B1 C1 D1 E1 F1	ii dd hh ll ee ff ff 	2 3 4 5 4 3
COM *opr* COMA COMX COM *opr*,X COM ,X	Complement Byte (One's Complement)	M ← ($\overline{M}$) = $FF – (M) A ← ($\overline{A}$) = $FF – (A) X ← ($\overline{X}$) = $FF – (X) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M)	—	—	↕	↕	1	DIR INH INH IX1 IX	33 43 53 63 73	dd ff 	5 3 3 6 5
CPX #*opr* CPX *opr* CPX *opr* CPX *opr*,X CPX *opr*,X CPX ,X	Compare Index Register with Memory Byte	(X) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A3 B3 C3 D3 E3 F3	ii dd hh ll ee ff ff 	2 3 4 5 4 3
DEC *opr* DECA DECX DEC *opr*,X DEC ,X	Decrement Byte	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1	—	—	↕	↕	—	DIR INH INH IX1 IX	3A 4A 5A 6A 7A	dd ff 	5 3 3 6 5
EOR #*opr* EOR *opr* EOR *opr* EOR *opr*,X EOR *opr*,X EOR ,X	EXCLUSIVE OR Accumulator with Memory Byte	A ← (A) ⊕ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A8 B8 C8 D8 E8 F8	ii dd hh ll ee ff ff 	2 3 4 5 4 3
INC *opr* INCA INCX INC *opr*,X INC ,X	Increment Byte	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1	—	—	↕	↕	—	DIR INH INH IX1 IX	3C 4C 5C 6C 7C	dd ff 	5 3 3 6 5
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Unconditional Jump	PC ← Jump Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff 	2 3 4 3 2

Table 11-6. Instruction Set Summary (Sheet 4 of 6)

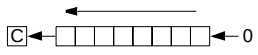
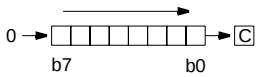
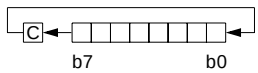
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Effective Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff	5 6 7 6 5
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X	Load Accumulator with Memory Byte	A ← (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A6 B6 C6 D6 E6 F6	ii dd hh ll ee ff ff	2 3 4 5 4 3
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X	Load Index Register with Memory Byte	X ← (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AE BE CE DE EE FE	ii dd hh ll ee ff ff	2 3 4 5 4 3
LSL *opr* LSLA LSLX LSL *opr*,X LSL ,X	Logical Shift Left (Same as ASL)		—	—	↕	↕	↕	DIR INH INH IX1 IX	38 48 58 68 78	dd ff 	5 3 3 6 5
LSR *opr* LSRA LSRX LSR *opr*,X LSR ,X	Logical Shift Right		—	—	0	↕	↕	DIR INH INH IX1 IX	34 44 54 64 74	dd ff 	5 3 3 6 5
MUL	Unsigned Multiply	X : A ← (X) × (A)	0	—	—	—	0	INH	42		11
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X	Negate Byte (Two's Complement)	M ← –(M) = $00 – (M) A ← –(A) = $00 – (A) X ← –(X) = $00 – (X) M ← –(M) = $00 – (M) M ← –(M) = $00 – (M)	—	—	↕	↕	↕	DIR INH INH IX1 IX	30 40 50 60 70	dd ff 	5 3 3 6 5
NOP	No Operation		—	—	—	—	—	INH	9D		2
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X	Logical OR Accumulator with Memory	A ← (A) ∨ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff	2 3 4 5 4 3
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X	Rotate Byte Left through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	39 49 59 69 79	dd ff 	5 3 3 6 5

Table 11-6. Instruction Set Summary (Sheet 5 of 6)

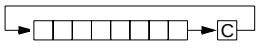
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ROR *opr* RORA RORX ROR *opr*,X ROR ,X	Rotate Byte Right through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	36 46 56 66 76	dd ff 	5 3 3 6 5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↕	↕	↕	↕	↕	INH	80		9
RTS	Return from Subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	—	—	—	—	—	INH	81		6
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X	Subtract Memory Byte and Carry Bit from Accumulator	A ← (A) – (M) – (C)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A2 B2 C2 D2 E2 F2	ii dd hh ll ee ff ff 	2 3 4 5 4 3
SEC	Set Carry Bit	C ← 1	—	—	—	—	1	INH	99		2
SEI	Set Interrupt Mask	I ← 1	—	1	—	—	—	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X	Store Accumulator in Memory	M ← (A)	—	—	↕	↕	—	DIR EXT IX2 IX1 IX	B7 C7 D7 E7 F7	dd hh ll ee ff ff 	4 5 6 5 4
STOP	Stop Oscillator and Enable IRQ Pin		—	0	—	—	—	INH	8E		2
STX *opr* STX *opr* STX *opr*,X STX *opr*,X STX ,X	Store Index Register In Memory	M ← (X)	—	—	↕	↕	—	DIR EXT IX2 IX1 IX	BF CF DF EF FF	dd hh ll ee ff ff 	4 5 6 5 4
SUB #*opr* SUB *opr* SUB *opr* SUB *opr*,X SUB *opr*,X SUB ,X	Subtract Memory Byte from Accumulator	A ← (A) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A0 B0 C0 D0 E0 F0	ii dd hh ll ee ff ff 	2 3 4 5 4 3
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	—	1	—	—	—	INH	83		10

Table 11-6. Instruction Set Summary (Sheet 6 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
TAX	Transfer Accumulator to Index Register	X ← (A)	—	—	—	—	—	INH	97		2
TST *opr* TSTA TSTX TST *opr*,X TST ,X	Test Memory Byte for Negative or Zero	(M) – $00	—	—	↕	↕	—	DIR INH INH IX1 IX	3D 4D 5D 6D 7D	dd ff 	4 3 3 5 4
TXA	Transfer Index Register to Accumulator	A ← (X)	—	—	—	—	—	INH	9F		2
WAIT	Stop CPU Clock and Enable Interrupts		—	0	—	—	—	INH	8F		2

A	Accumulator	*opr*	Operand (one or two bytes)
C	Carry/borrow flag	PC	Program counter
CCR	Condition code register	PCH	Program counter high byte
dd	Direct address of operand	PCL	Program counter low byte
dd rr	Direct address of operand and relative offset of branch instruction	REL	Relative addressing mode
DIR	Direct addressing mode	*rel*	Relative program counter offset byte
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing	rr	Relative program counter offset byte
EXT	Extended addressing mode	SP	Stack pointer
ff	Offset byte in indexed, 8-bit offset addressing	X	Index register
H	Half-carry flag	Z	Zero flag
hh ll	High and low bytes of operand address in extended addressing	#	Immediate value
I	Interrupt mask	∧	Logical AND
ii	Immediate operand byte	∨	Logical OR
IMM	Immediate addressing mode	⊕	Logical EXCLUSIVE OR
INH	Inherent addressing mode	( )	Contents of
IX	Indexed, no offset addressing mode	–( )	Negation (two's complement)
IX1	Indexed, 8-bit offset addressing mode	←	Loaded with
IX2	Indexed, 16-bit offset addressing mode	?	If
M	Memory location	:	Concatenated with
N	Negative flag	↕	Set or cleared
n	Any bit	—	Not affected

11.6 Opcode Map

See **Table 11-7**.

Table 11-7. Opcode Map

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	DIR	DIR	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
MSB / LSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	MSB / LSB
0	5 BRSET0 3 DIR	5 BSET0 2 DIR	3 BRA 2 REL	5 NEG 2 DIR	3 NEGA 1 INH	3 NEGX 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	**0**
1	5 BRCLR0 3 DIR	5 BCLR0 2 DIR	3 BRN 2 REL						6 RTS 1 INH		2 CMP 2 IMM	3 CMP 2 DIR	4 CMP 3 EXT	5 CMP 3 IX2	4 CMP 2 IX1	3 CMP 1 IX	**1**
2	5 BRSET1 3 DIR	5 BSET1 2 DIR	3 BHI 2 REL		11 MUL 1 INH						2 SBC 2 IMM	3 SBC 2 DIR	4 SBC 3 EXT	5 SBC 3 IX2	4 SBC 2 IX1	3 SBC 1 IX	**2**
3	5 BRCLR1 3 DIR	5 BCLR1 2 DIR	3 BLS 2 REL	5 COM 2 DIR	3 COMA 1 INH	3 COMX 1 INH	6 COM 2 IX1	5 COM 1 IX	10 SWI 1 INH		2 CPX 2 IMM	3 CPX 2 DIR	4 CPX 3 EXT	5 CPX 3 IX2	4 CPX 2 IX1	3 CPX 1 IX	**3**
4	5 BRSET2 3 DIR	5 BSET2 2 DIR	3 BCC 2 REL	5 LSR 2 DIR	3 LSRA 1 INH	3 LSRX 1 INH	6 LSR 2 IX1	5 LSR 1 IX			2 AND 2 IMM	3 AND 2 DIR	4 AND 3 EXT	5 AND 3 IX2	4 AND 2 IX1	3 AND 1 IX	**4**
5	5 BRCLR2 3 DIR	5 BCLR2 2 DIR	3 BCS/BLO 2 REL								2 BIT 2 IMM	3 BIT 2 DIR	4 BIT 3 EXT	5 BIT 3 IX2	4 BIT 2 IX1	3 BIT 1 IX	**5**
6	5 BRSET3 3 DIR	5 BSET3 2 DIR	3 BNE 2 REL	5 ROR 2 DIR	3 RORA 1 INH	3 RORX 1 INH	6 ROR 2 IX1	5 ROR 1 IX			2 LDA 2 IMM	3 LDA 2 DIR	4 LDA 3 EXT	5 LDA 3 IX2	4 LDA 2 IX1	3 LDA 1 IX	**6**
7	5 BRCLR3 3 DIR	5 BCLR3 2 DIR	3 BEQ 2 REL	5 ASR 2 DIR	3 ASRA 1 INH	3 ASRX 1 INH	6 ASR 2 IX1	5 ASR 1 IX		2 TAX 1 INH		4 STA 2 DIR	5 STA 3 EXT	6 STA 3 IX2	5 STA 2 IX1	4 STA 1 IX	**7**
8	5 BRSET4 3 DIR	5 BSET4 2 DIR	3 BHCC 2 REL	5 ASL/LSL 2 DIR	3 ASLA/LSLA 1 INH	3 ASLX/LSLX 1 INH	6 ASL/LSL 2 IX1	5 ASL/LSL 1 IX		2 CLC 1 INH	2 EOR 2 IMM	3 EOR 2 DIR	4 EOR 3 EXT	5 EOR 3 IX2	4 EOR 2 IX1	3 EOR 1 IX	**8**
9	5 BRCLR4 3 DIR	5 BCLR4 2 DIR	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	**9**
A	5 BRSET5 3 DIR	5 BSET5 2 DIR	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	**A**
B	5 BRCLR5 3 DIR	5 BCLR5 2 DIR	3 BMI 2 REL							2 SEI 1 INH	2 ADD 2 IMM	3 ADD 2 DIR	4 ADD 3 EXT	5 ADD 3 IX2	4 ADD 2 IX1	3 ADD 1 IX	**B**
C	5 BRSET6 3 DIR	5 BSET6 2 DIR	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	**C**
D	5 BRCLR6 3 DIR	5 BCLR6 2 DIR	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	**D**
E	5 BRSET7 3 DIR	5 BSET7 2 DIR	3 BIL 2 REL						2 STOP 1 INH		2 LDX 2 IMM	3 LDX 2 DIR	4 LDX 3 EXT	5 LDX 3 IX2	4 LDX 2 IX1	3 LDX 1 IX	**E**
F	5 BRCLR7 3 DIR	5 BCLR7 2 DIR	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	**F**

INH = Inherent
IMM = Immediate
DIR = Direct
EXT = Extended
REL = Relative
IX = Indexed, No Offset
IX1 = Indexed, 8-Bit Offset
IX2 = Indexed, 16-Bit Offset

LSB of Opcode in Hexadecimal

MSB / LSB	0
0	5 BRSET0 3 DIR

MSB of Opcode in Hexadecimal
Number of Cycles
Opcode Mnemonic
Number of Bytes/Addressing Mode

Section 12. Electrical Specifications

12.1 Contents

12.2 Maximum Ratings

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

The MCU contains circuitry to protect the inputs against damage from high static voltages; however, do not apply voltages higher than those shown in the table below. Keep V_{In} and V_{Out} within the range $V_{SS} \le (V_{In}$ or $V_{Out}) \le V_{DD}$. Connect unused inputs to the appropriate voltage level, either V_{SS} or V_{DD}.

Rating	Symbol	Value	Unit
Supply voltage	V_{DD}	–0.3 to +7.0	V
Input voltage Normal operation Bootloader mode ($\overline{IRQ}$ pin only)	V_{In}	V_{SS} –0.3 to V_{DD} + 0.3 V_{SS} –0.3 to 2 x V_{DD} + 0.3	V
Current drain per pin (Excluding V_{DD} and V_{SS})	I	25	mA
Storage temperature range	T_{STG}	–65 to +150	°C

NOTE: *This device is not guaranteed to operate properly at the maximum ratings. Refer to* ***12.6 5.0-Vdc Electrical Characteristics*** *for guaranteed operating conditions.*

12.3 Operating Temperature

Characteristic	Symbol	Value	Unit
Operating temperature range	T_A	–40 to +85	°C

12.4 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal resistance plastic dual in-line (PDIP)	θ_{JA}	60	°C/W
Thermal resistance plastic leaded chip carrier (PLCC)	θ_{JA}	70	°C/W
Thermal resistance quad flat pack (QFP)	θ_{JA}	95	°C/W
Thermal resistance plastic shrink DIP (SDIP)	θ_{JA}	60	°C/W

12.5 Power Considerations

The average chip-junction temperature, T_J, in °C, can be obtained from:

$$T_J = T_A + (P_D \times \theta_{JA}) \quad (1)$$

where:

T_A = Ambient temperature, °C
θ_{JA} = Package thermal resistance, junction to ambient, °C/W
$P_D = P_{INT} + P_{I/O}$
$P_{INT} = I_{DD} \times V_{DD}$ watts (chip internal power)
$P_{I/O}$ = Power dissipation on input and output pins (user determined)

For most applications $P_{I/O} \ll P_{INT}$ and can be neglected.

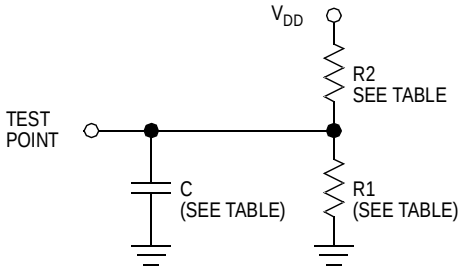
The following is an approximate relationship between P_D and T_J (neglecting P_J):

$$P_D = K \div (T_J + 273\ °C) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \times (T_A + 273\ °C) + \theta_{JA} \times (P_D)^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.



V_{DD} = 4.5 V

Pins	R1	R2	C
PA7–PA0 PB7–PB0 PC7–PC0 PD5–PD0, PD7	3.26 Ω	2.38 Ω	50 pF

V_{DD} = 3.0 V

Pins	R1	R2	C
PA7–PA0 PB7–PB0 PC7–PC0 PD5–PD0, PD7	10.91 Ω	6.32 Ω	50 pF

Figure 12-1. Test Load

12.6 5.0-Vdc Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output voltage I_{Load} = 10.0 µA I_{Load} = –10.0 µA	 V_{OL} V_{OH}	 — V_{DD} –0.1	 — —	 0.1 —	 V
Output high voltage (I_{Load} = –0.8 mA) PA7–PA0, PB7–PB0, PC6–PC0, TCMP, PD7, PD0 (I_{Load} = –1.6 mA) PD5–PD1 (I_{Load} = –5.0 mA) PC7	V_{OH}	V_{DD} –0.8 V_{DD} –0.8 V_{DD} –0.8	— — —	— — —	V
Output low voltage (I_{Load} = 1.6 mA) PA7–PA0, PB7–PB0, PC6–PC0, PD7, PD5–PD0, TCMP (I_{Load} = 10 mA) PC7	V_{OL}	— —	— —	0.4 0.4	V
Input high voltage PA7–PA0, PB7–PB0, PC7–PC0, PD7, PD5–PD0, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input low voltage PA7–PA0, PB7–PB0, PC7–PC0, PD7, PD5–PD0, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 × V_{DD}	V
Supply current (4.5–5.5 Vdc @ f_{OP} = 2.1 MHz) Run[3] Wait[4] Stop[5] 25°C –40 to 85 °C	 I_{DD}	 — — — —	 3.5 1.0 1.0 7.0	 5.25 3.25 20.0 50.0	 mA mA µA µA
I/O ports hi-Z leakage current PA7–PA0, PB7–PB0 (without pullup) PC7–PC0, PD7, PD5–PD0	I_{OZ}	—	—	10	µA
Input current $\overline{RESET}$, $\overline{IRQ}$, OSC1, TCAP, PD7, PD5–PD0	I_{In}	—	—	1	µA
Input pullup current[6] PB7–PB0 (with pullup)	I_{In}	5	—	60	µA
Capacitance Ports (as input or output) $\overline{RESET}$, $\overline{IRQ}$, OSC1, TCAP, PD7, PD5, PD0	C_{Out} C_{In}	— —	— —	12 8	pF
Programming voltage (25°C)	V_{PP}	15.0	16.0	17.0	V
Programming current (25°C)	I_{PP}	—	—	200	mA

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = –40 to +85 °C, unless otherwise noted
2. Typical values reflect measurements taken on average processed devices at the midpoint of voltage range, 25 °C only.
3. Run (operating) I_{DD} measured using external square wave clock source; all I/O pins configured as inputs, port B = V_{DD}, all other inputs V_{IL} = 0.2 V, V_{IH} = V_{DD}–0.2 V; no DC loads; less than 50 pF on all outputs; C_L = 20 pF on OSC2
4. Wait I_{DD} measured using external square wave clock source; all I/O pins configured as inputs, port B = V_{DD}, all other inputs V_{IL} = 0.2 V, V_{IH} = V_{DD} –0.2 V; no DC loads; less than 50 pF on all outputs; C_L = 20 pF on OSC2. Wait I_{DD} is affected linearly by the OSC2 capacitance.
5. Stop I_{DD} measured with OSC1 = 0.2 V; all I/O pins configured as inputs, port B = V_{DD}, all other inputs V_{IL} = 0.2 V, V_{IH} = V_{DD} –0.2 V.
6. Input pullup current measured with V_{IL} = 0.2 V.

12.7 3.3-Vdc Electrical Characteristics

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output voltage I_{Load} = 10.0 µA I_{Load} = –10.0 µA	 V_{OL} V_{OH}	 — V_{DD} –0.1	 — —	 0.1 —	V
Output high voltage (I_{Load} = –0.2 mA) PA7–PA0, PB7–PB0, PC6–PC0, TCMP, PD7, PD0 (I_{Load} = –0.4 mA) PD5–PD1 (I_{Load} = –1.5 mA) PC7	V_{OH}	V_{DD} –0.3 V_{DD} –0.3 V_{DD} –0.3	— — —	— — —	V
Output low voltage (I_{Load} = 0.4mA) PA7–PA0, PB7–PB0, PC6–PC0, PD7, PD5–PD0, TCMP (I_{Load} = 6 mA) PC7	V_{OL}	— —	— —	0.3 0.3	V
Input high voltage PA7–PA0, PB7–PB0, PC7–PC0, PD7, PD5–PD0, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input low voltage PA7–PA0, PB7–PB0, PC7–PC0, PD7, PD5–PD0, TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 × V_{DD}	V
Supply current (3.0–3.6 Vdc @ f_{OP} = 1.0 MHz) Run[3] Wait[4] Stop[5] 25°C –40 to 85°C	 I_{DD}	 — — — —	 1.0 500 1.0 2.5	 1.6 900 8 20	 mA µA µA µA
I/O ports hi-Z leakage current PA7–PA0, PB7–PB0 (without pullup) PC7–PC0, PD7, PD5–PD0	I_{OZ}	—	—	10	µA
Input current $\overline{RESET}$, $\overline{IRQ}$, OSC1, TCAP, PD7, PD5–PD0	I_{In}	—	—	1	µA
Input pullup current[6] PB7–PB0 (with pullup)	I_{In}	0.5	—	20	µA
Capacitance Ports (as input or output) $\overline{RESET}$, $\overline{IRQ}$, OSC1, TCAP, PD7, PD5, PD0	C_{Out} C_{In}	— —	— —	12 8	pF

1. V_{DD} = 3.3 Vdc ± 0.3 Vdc, V_{SS} = 0 Vdc, T_A = –40 to +85 °C, unless otherwise noted
2. Typical values reflect measurements taken on average processed devices at the midpoint of voltage range, 25 °C only.
3. Run (operating) I_{DD} measured using external square wave clock source; all I/O pins configured as inputs, port B = V_{DD}, all other inputs V_{IL} = 0.2 V, V_{IH} = V_{DD}–0.2 V; no DC loads; less than 50 pF on all outputs; C_L = 20 pF on OSC2
4. Wait I_{DD} measured using external square wave clock source; all I/O pins configured as inputs, port B = V_{DD}, all other inputs V_{IL} = 0.2 V, V_{IH} = V_{DD} –0.2 V; no DC loads; less than 50 pF on all outputs; C_L = 20 pF on OSC2. Wait I_{DD} is affected linearly by the OSC2 capacitance.
5. Stop I_{DD} measured with OSC1 = 0.2 V; all I/O pins configured as inputs, port B = V_{DD}, all other inputs V_{IL} = 0.2 V, V_{IH} = V_{DD}–0.2 V.
6. Input pullup current measured with V_{IL} = 0.2 V.

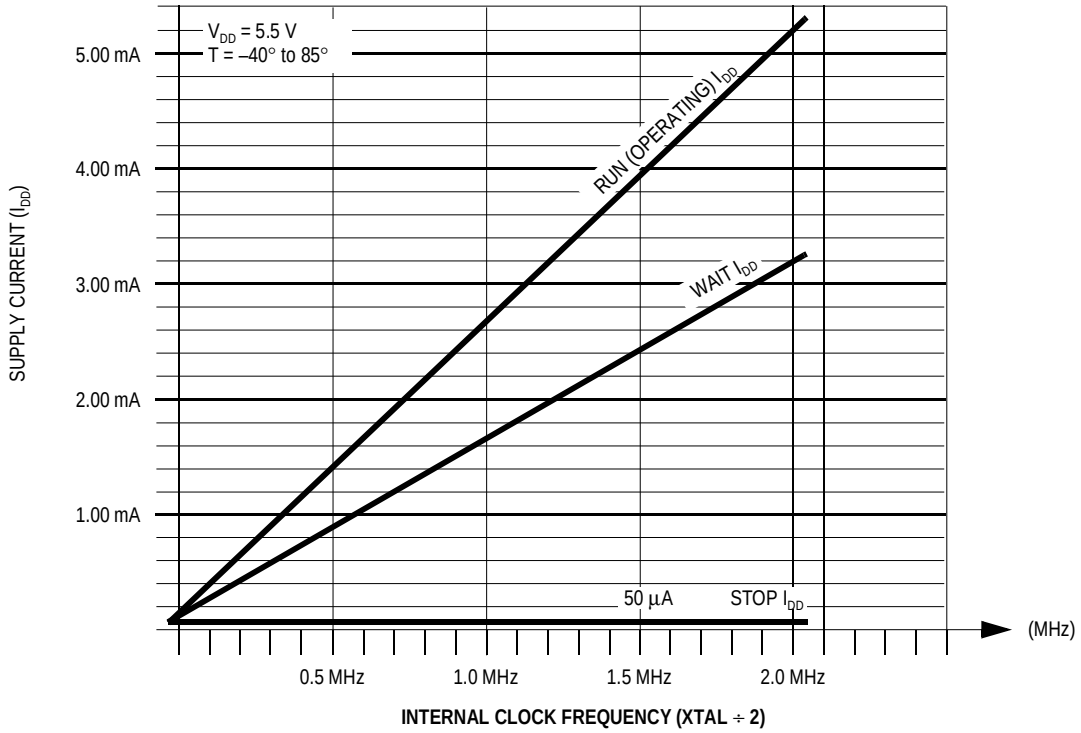


Figure 12-2. Maximum Supply Current vs Internal Clock Frequency, V_{DD} = 5.5 V

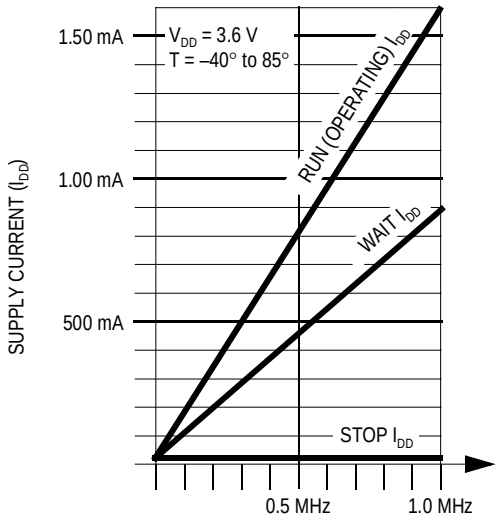


Figure 12-3. Maximum Supply Current vs Internal Clock Frequency, V_{DD} = 3.6 V

12.8 5.0-Vdc Control Timing

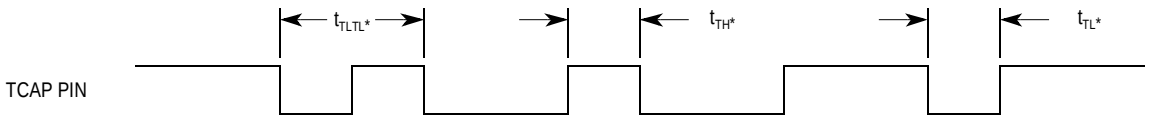
Characteristic[1]	Symbol	Min	Max	Unit
Frequency of operation Crystal External clock	f_{OSC}	 — DC	 4.2 4.2	MHz
Internal pperating frequency ($f_{OSC} \div 2$) Crystal External clock	f_{OP}	 — DC	 2.1 2.1	MHz
Cycle time	t_{CYC}	480	—	ns
Crystal oscillator startup time	t_{OXOV}	—	100	ms
Stop recovery startup time (crystal oscillator)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ pulse width	t_{RL}	1.5	—	t_{CYC}
Timer Resolution[2] Input capture pulse width Input capture pulse period	 t_{RESL} t_{TH}, t_{TL} t_{TLTL}	 4.0 125 [3]	 — — —	 t_{CYC} ns t_{CYC}
Interrupt pulse width low (edge-triggered)	t_{ILIH}	125	—	ns
Interrupt pulse period	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 = –40 to +85 °C, unless otherwise noted
2. Because 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 t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 19 t_{CYC}.

12.9 3.3-Vdc Control Timing

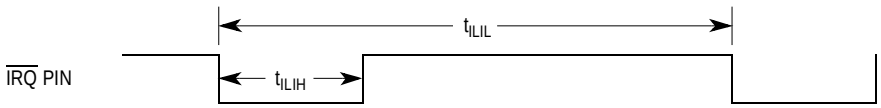
Characteristic[1]	Symbol	Min	Max	Unit
Frequency of operation Crystal External clock	f_{OSC}	 — DC	 2.0 2.0	MHz
Internal operating frequency ($f_{OSC} \div 2$) Crystal External clock	f_{OP}	 — DC	 1.0 1.0	MHz
Cycle time	t_{CYC}	1000	—	ns
Crystal oscillator startup time	t_{OXOV}	—	100	ms
Stop recovery startup time (crystal oscillator)	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ pulse width	t_{RL}	1.5	—	t_{CYC}
Timer Resolution[2] Input capture pulse width Input capture pulse period	 t_{RESL} t_{TH}, t_{TL} t_{TLTL}	 4.0 125 [3]	 — — —	 t_{CYC} ns t_{CYC}
Interrupt pulse width low (edge-triggered)	t_{ILIH}	250	—	ns
Interrupt pulse period	t_{ILIL}	[4]	—	t_{CYC}
OSC1 pulse width	t_{OH},t_{OL}	200	—	ns

1. V_{DD} = 3.3Vdc ± 0.3 Vdc, V_{SS} = 0 Vdc, T_A = –40 to +85°C, unless otherwise noted
2. Because 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 t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 19 t_{CYC}.

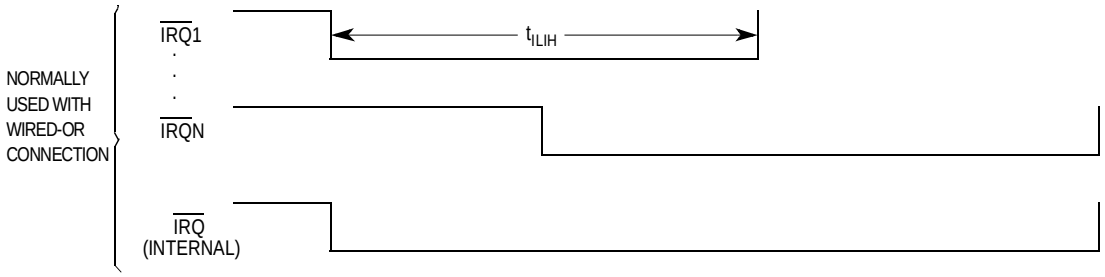


* Refer to timer resolution data in **12.8 5.0-Vdc Control Timing** and **12.9 3.3-Vdc Control Timing**.

Figure 12-4. TCAP Timing Relationships

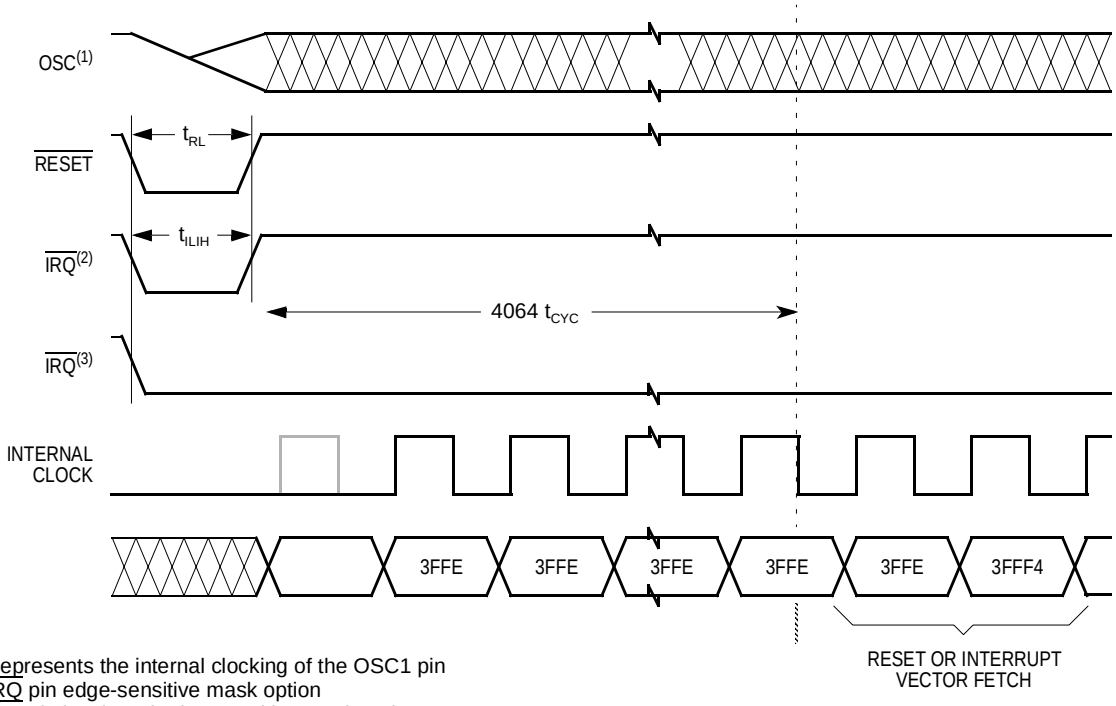


a. **Edge-Sensitive Trigger Condition**. The minimum pulse width (t_{ILIH}) is either 125 ns (f_{OP} = 2.1 MHz) or 250 ns (f_{OP} = 1 MHz). The period t_{ILIL} should not be less than the number of t_{CYC} cycles it takes to execute the interrupt service routine plus 19 t_{CYC} cycles.



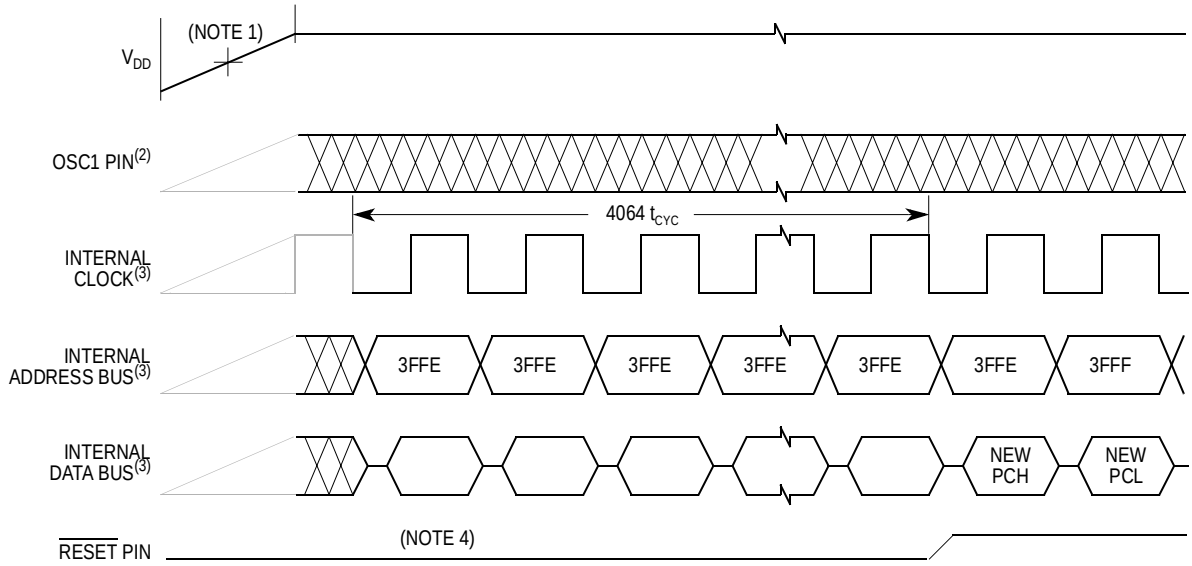
b. **Level-Sensitive Trigger Condition**. If after servicing an interrupt the $\overline{IRQ}$ remains low, the next interrupt is recognized

Figure 12-5. External Interrupt Timing



Notes:
1. Represents the internal clocking of the OSC1 pin
2. IRQ pin edge-sensitive mask option
3. IRQ pin level- and edge-sensitive mask option
4. RESET vector address shown for timing example

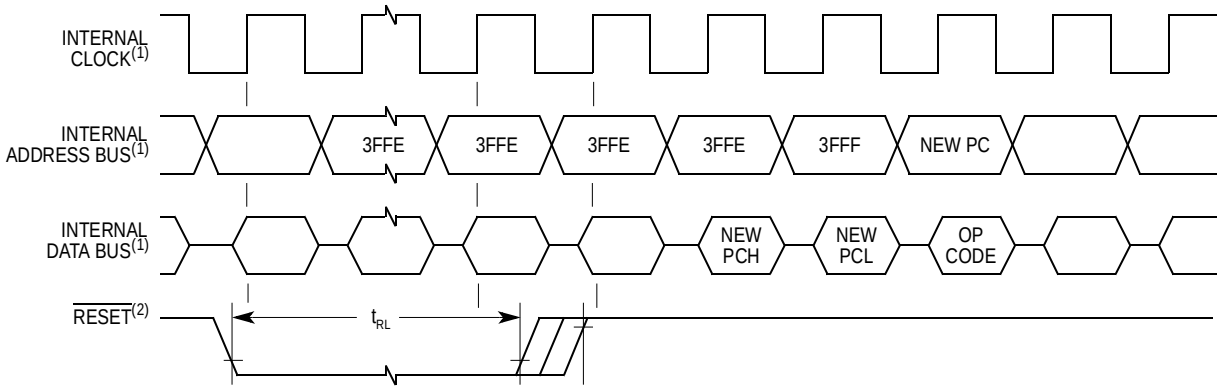
Figure 12-6. STOP Recovery Timing Diagram



Notes:

1. Power-on reset threshold is typically between 1 V and 2 V.
2. OSC1 line is meant to represent time only, not frequency.
3. Internal clock, internal address bus, and internal data bus are not available externally.
4. RESET outputs V_{OL} during 4064 POR cycles.

Figure 12-7. Power-On Reset Timing Diagram



Notes:

1. Internal clock, internal address bus, and internal data bus are not available externally.
2. The next rising edge of the internal clock after the rising edge of RESET initiates the reset sequence.

Figure 12-8. External Reset Timing

12.10 5.0-Vdc Serial Peripheral Interface Timing

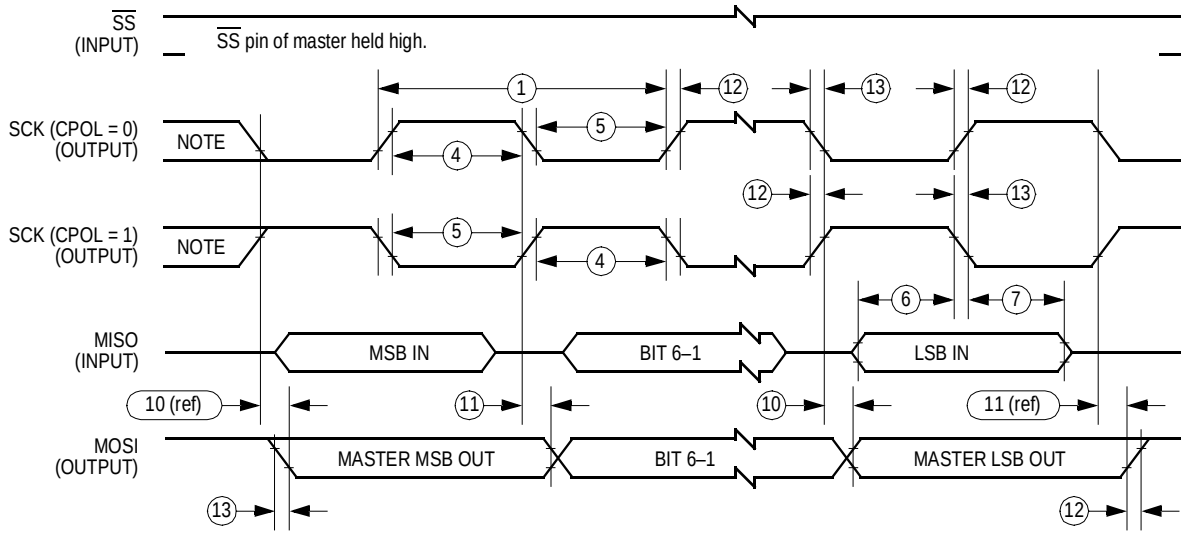
No.	Characteristic[1]	Symbol	Min	Max	Unit
	Operating frequency Master Slave	$f_{OP(M)}$ $f_{OP(S)}$	dc dc	0.5 2.1	f_{OP} MHz
1	Cycle time Master Slave	$t_{CYC(M)}$ $t_{CYC(S)}$	2.0 480	— —	t_{CYC} ns
2	Enable lead time Master Slave	$t_{LEAD(M)}$ $t_{LEAD(S)}$	(2) 240	— —	ns
3	Enable lag time Master Slave	$t_{LAG(M)}$ $t_{LAG(S)}$	(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	Slave access time (time to data active from high-impedance state)	t_A	0	120	ns
9	Slave disable time (hold time to high-impedance state)	t_{DIS}	—	240	ns
10	Data valid Master (before capture edge) Slave (after enable edge)[3]	$t_{V(M)}$ $t_{V(S)}$	0.25 —	— 240	$t_{CYC(M)}$ ns
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 (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI outputs (SCK, MOSI, and MISO) SPI inputs (SCK, MOSI, MISO, and $\overline{SS}$)	t_{RM} t_{RS}	— —	100 2.0	ns µs
13	Fall time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI outputs (SCK, MOSI, and MISO) SPI inputs (SCK, MOSI, MISO, and $\overline{SS}$)	t_{FM} t_{FS}	— —	100 2.0	ns µs

1. V_{DD} = 5.0 Vdc ± 10%; V_{SS} = 0 Vdc, T_A = –40 to +85°C, unless otherwise noted. Refer to **Figure 12-9** and **Figure 12-10**.
2. Signal production depends on software.
3. Assumes 200 pF load on all SPI pins.

12.11 3.3- Vdc Serial Peirpheral Interface Timing

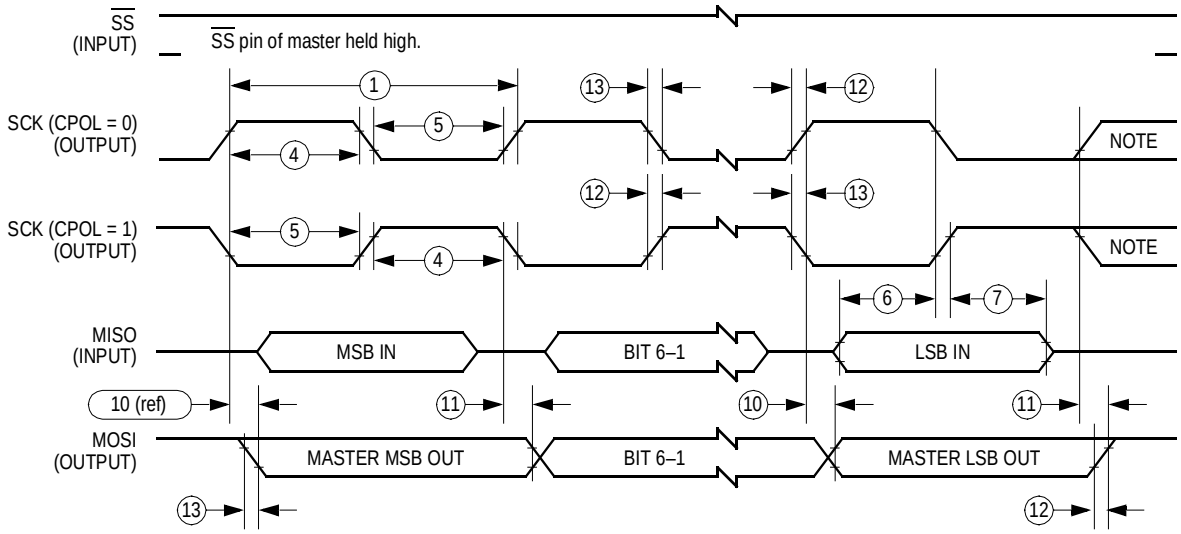
No.	Characteristic[1]	Symbol	Min	Max	Unit
	Operating frequency Master Slave	$f_{OP(M)}$ $f_{OP(S)}$	dc dc	0.5 1.0	f_{OP} MHz
1	Cycle time Master Slave	$t_{CYC(M)}$ $t_{CYC(S)}$	2.0 1.0	— —	t_{CYC} µs
2	Enable lead time Master Slave	$t_{LEAD(M)}$ $t_{LEAD(S)}$	(2) 500	— —	ns
3	Enable lag time Master Slave	$t_{LAG(M)}$ $t_{LAG(S)}$	(2) 1.5	— —	ns µs
4	Clock (SCK) high time Master Slave	$t_{W(SCKH)M}$ $t_{W(SCKH)S}$	720 400	— —	ns
5	Clock (SCK) low time Master Slave	$t_{W(SCKL)M}$ $t_{W(SCKL)S}$	720 400	— —	ns
6	Data setup time (inputs) Master Slave	$t_{SU(M)}$ $t_{SU(S)}$	200 200	— —	ns
7	Data hold time (inputs) Master Slave	$t_{H(M)}$ $t_{H(S)}$	200 200	— —	ns
8	Slave access time (time to data active from high-impedance state)	t_A	0	250	ns
9	Slave disable time (hold time to high-impedance state)	t_{DIS}	—	500	ns
10	Data valid Master (before capture edge) Slave (after enable edge)[3]	$t_{V(M)}$ $t_{V(S)}$	0.25 —	— 500	$t_{CYC(M)}$ ns
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 (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI outputs (SCK, MOSI, and MISO) SPI inputs (SCK, MOSI, MISO, and $\overline{SS}$)	t_{RM} t_{RS}	— —	200 2.0	ns µs
13	Fall time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI outputs (SCK, MOSI, and MISO) SPI inputs (SCK, MOSI, MISO, and $\overline{SS}$)	t_{FM} t_{FS}	— —	200 2.0	ns µs

1. V_{DD} = 3.3 Vdc ± 0.3 Vdc; V_{SS} = 0 Vdc, T_A = –40 to +85 °C, unless otherwise noted. Refer to **Figure 12-9** and **Figure 12-10**.
2. Signal production depends on software.
3. Assumes 200 pF load on all SPI pins.



Note:
This first clock edge is generated internally, but is not seen at the SCK pin.

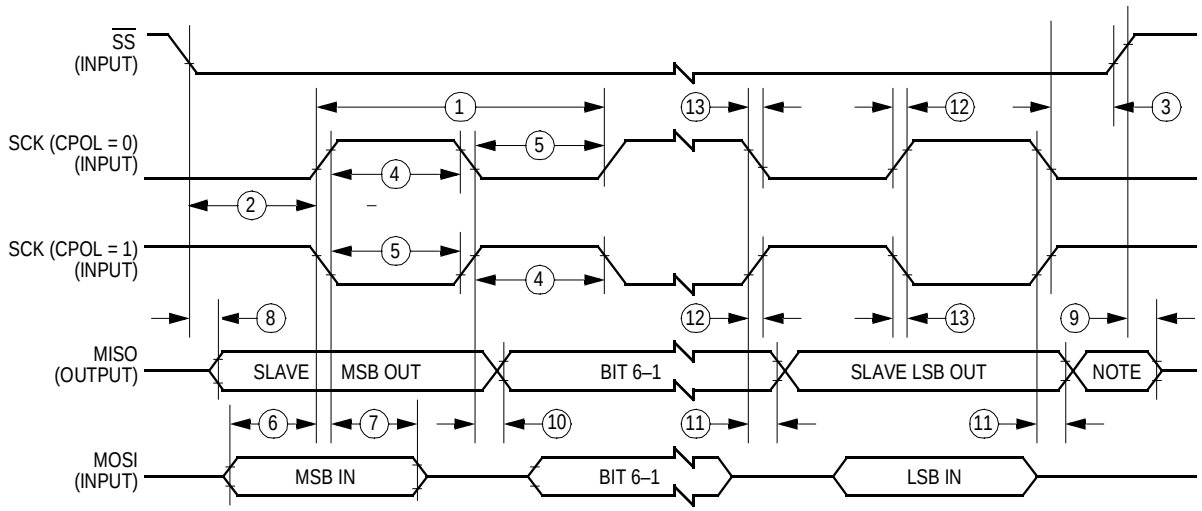
a) SPI Master Timing (CPHA = 0)



Note:
This last clock edge is generated internally, but is not seen at the SCK pin.

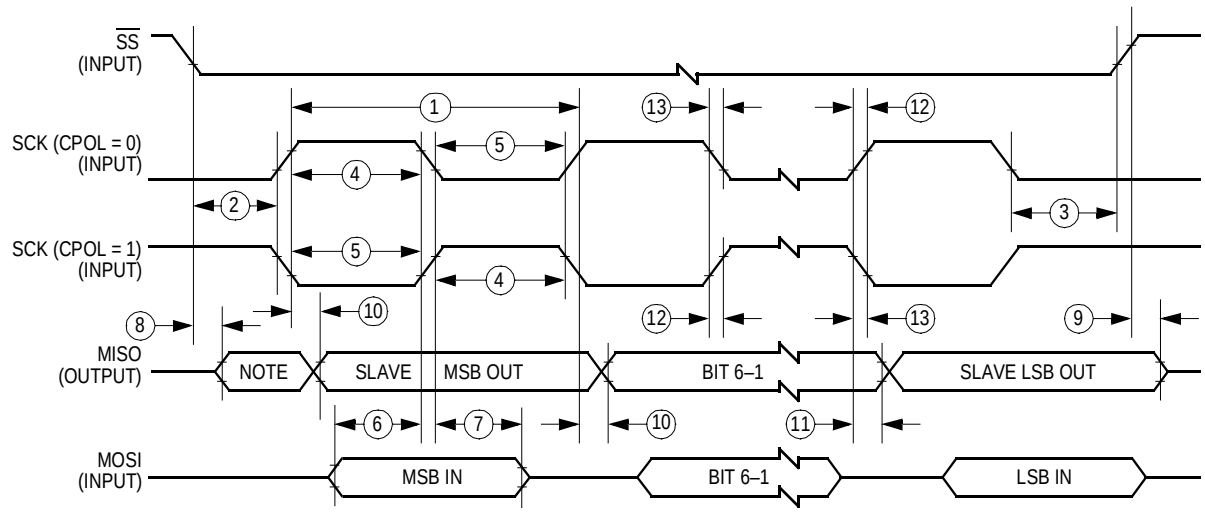
b) SPI Master Timing (CPHA = 1)

Figure 12-9. SPI Master Timing Diagram



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 12-10. SPI Slave Timing Diagram

Electrical Specifications

Section 13. Mechanical Specifications

13.1 Contents

13.2 Introduction

This section describes the dimensions of the plastic dual in-line package (DIP), plastic shrink dual in-line package (SDIP), plastic leaded chip carrier (PLCC), and quad flat pack (QFP) MCU packages.

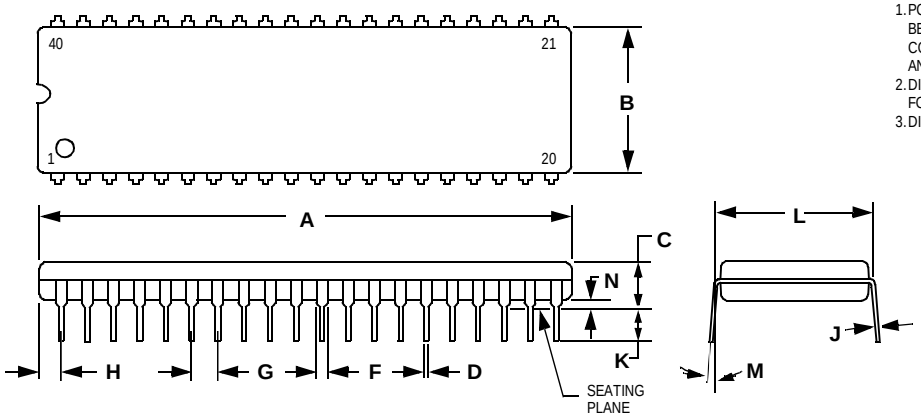
Package dimensions available at the time of this publication are provided in this section.

To make sure that you have the latest case outline specifications, contact one of the following:

- Local Motorola Sales Office
- World Wide Web at http://www.motorola.com/semiconductors

Follow World Wide Web on-line instructions to retrieve the current mechanical specifications.

13.3 40-Pin Plastic Dual In-Line (DIP) Package (Case 711-03)

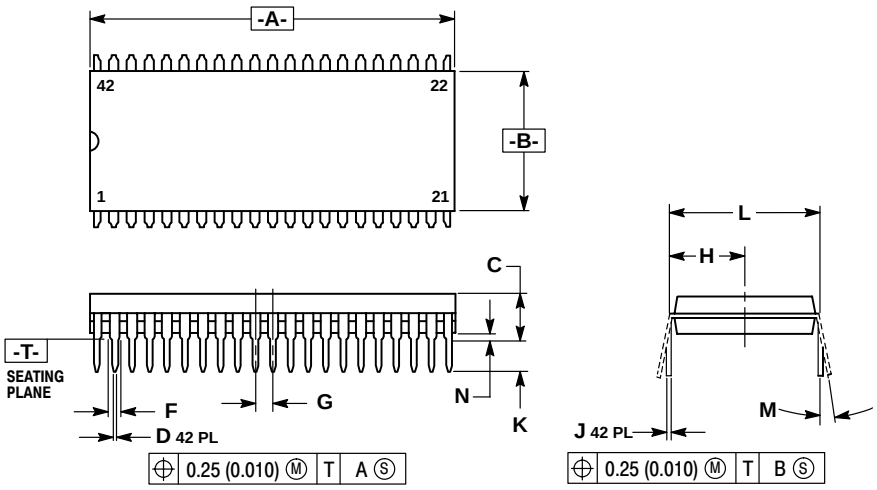


NOTES:
1. POSITION TOLERANCE OF LEADS (D), SHALL BEWITHIN 0.25 (0.010) AT MAXIMUM MATERIAL CONDITIONS, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	51.69	52.45	2.035	2.065
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	1°	0°	1°
N	0.51	1.02	0.020	0.040

Figure 13-1. 40-Pin Plastic DIP Package (Case 711-03)

13.4 42-Pin Plastic Shrink Dual In-Line (SDIP) Package (Case 858-01)

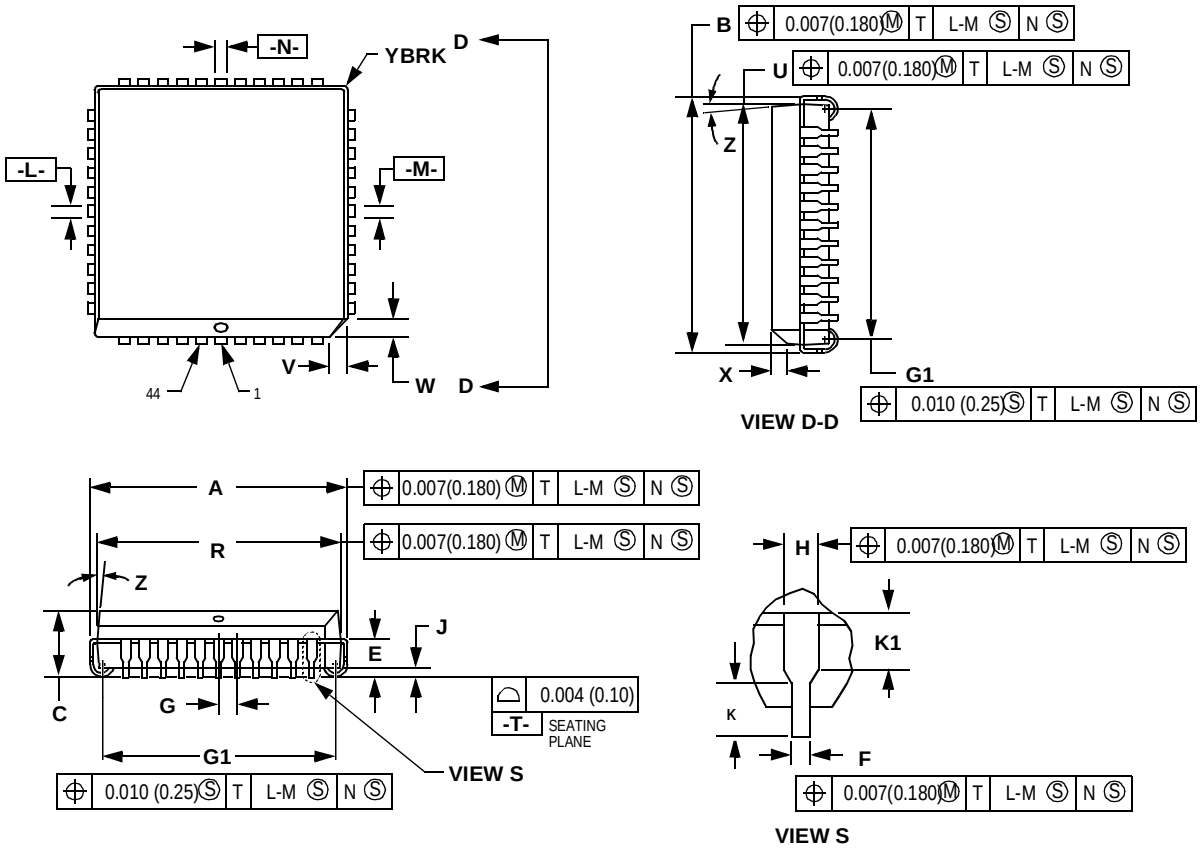


NOTES:
1. DIMENSIONS AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.
4. DIMENSIONS A AND B DO NOT INCLUDE MOLD FLASH. MAXIMUM MOLD FLASH 0.25 (0.010).

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	1.435	1.465	36.45	37.21
B	0.540	0.560	13.72	14.22
C	0.155	0.200	3.94	5.08
D	0.014	0.022	0.36	0.56
F	0.032	0.046	0.81	1.17
G	0.070 BSC		1.778 BSC	
H	0.300 BSC		7.62 BSC	
J	0.008	0.015	0.20	0.38
K	0.115	0.135	2.92	3.43
L	0.600 BSC		15.24 BSC	
M	0°	15°	0°	15°
N	0.020	0.040	0.51	1.02

Figure 13-2. 42-Pin Plastic SDIP Package (Case 858-01)

13.5 44-Lead Plastic Leaded Chip Carrier (PLCC) (Case 777-02)

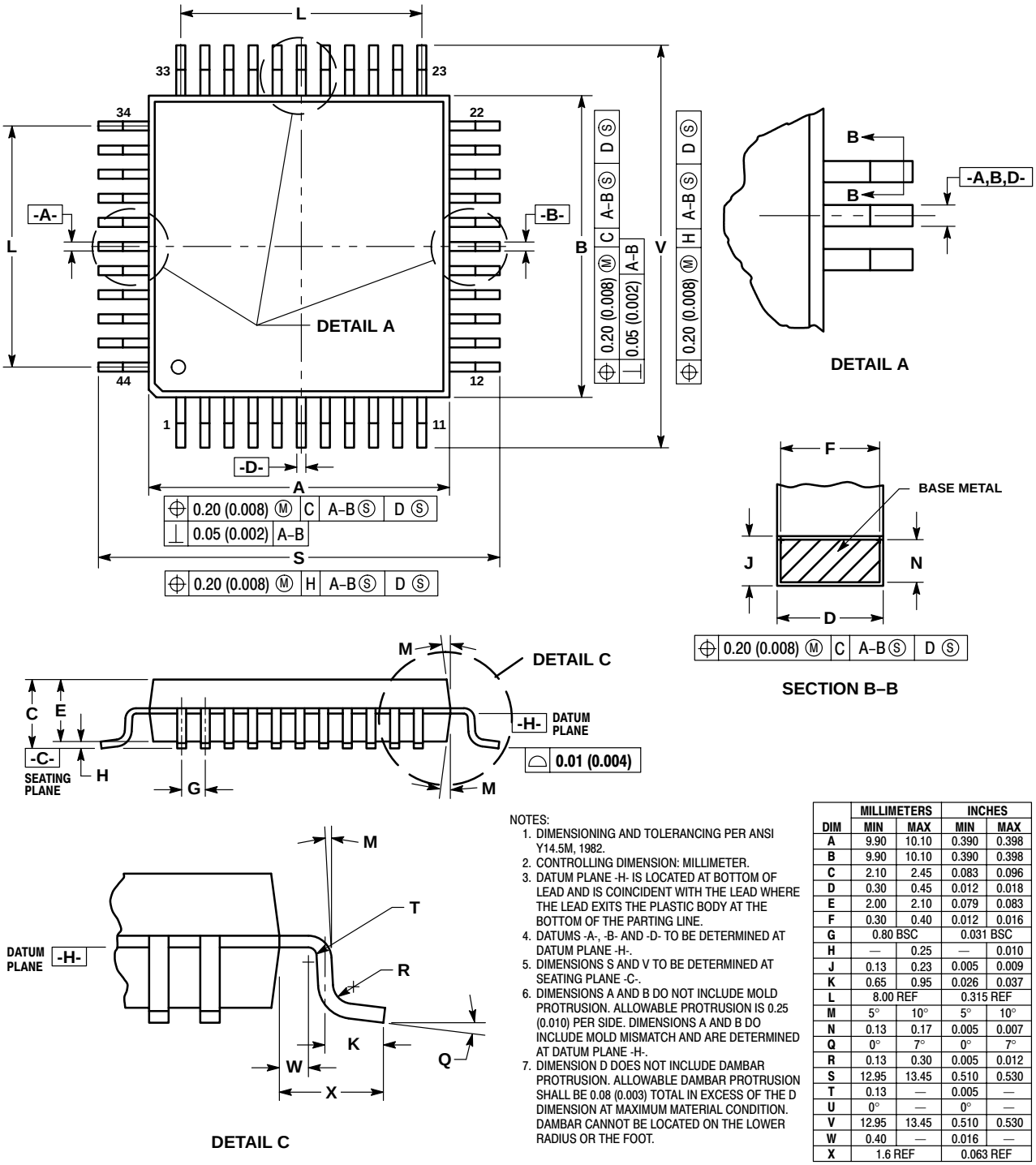


NOTES:
1. DATUMS -L-, -M-, AND -N- ARE DETERMINED WHERE TOP OF LEAD SHOLDERS EXITS PLASTIC BODY AT MOLD PARTING LINE.
2. DIMENSION G1, TRUE POSITION TO BE MEASURED AT DATUM -T-, SEATING PLANE.
3. DIMENSION R AND U DO NOT INCLUDE MOLD FLASH. ALLOWABLE MOLD FLASH IS 0.010 (0.25) PER SIDE.
4. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
5. CONTROLLING DIMENSION: INCH.
6. THE PACKAGE TOP MAY BE SMALLER THAN THE PACKAGE BOTTOM BY UP TO 0.012 (0.300). DIMENSIONS R AND U ARE DETERMINED AT THE OUTERMOST EXTREMES OF THE PLASTIC BODY EXCLUSIVE OF THE MOLD FLASH, TIE BAR BURRS, GATE BURRS AND INTERLEAD FLASH, BUT INCLUDING ANY MISMATCH BETWEEN THE TOP AND BOTTOM OF THE PLASTIC BODY.
7. DIMINSION H DOES NOT INCLUDE DAMBAR PROTRUSION OR INTRUSION. THE DAMBAR PROTUSION(S) SHALL NOT CAUSE THE H DIMINSION TO BE GREATER THAN 0.037 (0.940150). THE DAMBAR INTRUSION(S) SHALL NOT CAUSE THE H DIMINISION TO SMALLER THAN 0.025 (0.635).

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.685	0.695	17.40	17.65
B	0.685	0.695	17.40	17.65
C	0.165	0.180	4.20	4.57
E	0.090	0.110	2.29	2.79
F	0.013	0.019	0.33	0.48
G	0.050 BSC		1.27 BSC	
H	0.026	0.032	0.66	0.81
J	0.020	—	0.51	—
K	0.025	—	0.64	—
R	0.650	0.656	16.51	16.66
U	0.650	0.656	16.51	16.66
V	0.042	0.048	1.07	1.21
W	0.042	0.048	1.07	1.21
X	0.042	0.056	1.07	1.42
Y	—	0.020	—	0.50
Z	2°	10°	2°	10°
G1	0.610	0.630	15.50	16.00
K1	0.040	—	1.02	—

Figure 13-3. 44-Lead PLCC (Case 777-02)

13.6 44-Lead Quad Flat Pack (QFP) (Case 824A-01)



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM PLANE -H- IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS -A-, -B- AND -D- TO BE DETERMINED AT DATUM PLANE -H-.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE -C-.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -H-.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08 (0.003) TOTAL IN EXCESS OF THE D DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	9.90	10.10	0.390	0.398
B	9.90	10.10	0.390	0.398
C	2.10	2.45	0.083	0.096
D	0.30	0.45	0.012	0.018
E	2.00	2.10	0.079	0.083
F	0.30	0.40	0.012	0.016
G	0.80 BSC		0.031 BSC	
H	—	0.25	—	0.010
J	0.13	0.23	0.005	0.009
K	0.65	0.95	0.026	0.037
L	8.00 REF		0.315 REF	
M	5°	10°	5°	10°
N	0.13	0.17	0.005	0.007
Q	0°	7°	0°	7°
R	0.13	0.30	0.005	0.012
S	12.95	13.45	0.510	0.530
T	0.13	—	0.005	—
U	0°	—	0°	—
V	12.95	13.45	0.510	0.530
W	0.40	—	0.016	—
X	1.6 REF		0.063 REF	

Figure 13-4. 44-Lead QFP (Case 824A-01)

Section 14. Ordering Information

14.1 Contents

14.2 Introduction

This section contains ordering information for the available package types.

14.3 MC Order Numbers

Table 14-1 shows the MC order numbers for the available package types.

Table 14-1. MC Order Numbers

Package Type	Temperature Range	Order Number
40-pin plastic dual in-line package (DIP)	–40°C to 85°C	MC68HC705C9ACP
42-pin shrink dual in-line package (SDIP)	–40°C to 85°C	MC68HC705C9ACB
44-lead plastic leaded chip carrier (PLCC)	–40°C to 85°C	MC68HC705C9ACFN
44-pin quad flat pack (QFP)	–40°C to 85°C	MC68HC705C9ACFB

P = Plastic dual in-line package (PDIP)
B = Shrink dual in-line package (SDIP)
FN = Plastic-leaded chip carrier (PLCC)
FB = Quad flat pack (QFP)

Ordering Information

Appendix A. EPROM Programming

A.1 Contents

A.2 Introduction

This section describes programming of the EPROM.

A.3 Bootloader Mode

Table A-1. Operating Modes

$\overline{\text{RESET}}$	$\overline{\text{IRQ}}$	TCAP	Mode
	V_{SS} to V_{DD} V_{TST}	V_{SS} to V_{DD} V_{DD}	User Bootloader

Bootloader mode is entered upon the rising edge of $\overline{\text{RESET}}$ if the $\overline{\text{IRQ}}$ is at V_{TST} and the TCAP pin is at logic one. The bootloader code resides in the ROM from $3F00 to $3FEF. This program handles copying of user code from an external EPROM into the on-chip EPROM. The bootload function does not have to be done from an external EPROM, but it may be done from a host.

The user code must be a one-to-one correspondence with the internal EPROM addresses.

A.4 Bootloader Functions

Three pins are used to select various bootloader functions: PD5, PD4, and PD3. Two other pins, PC6 and PC7, are used to drive the PROG LED and the VERF LED, respectively. The programming modes are shown in **Table A-2**.

Table A-2. Bootloader Functions

PD5	PD4	PD3	Mode
0	0	0	Program/verify
0	0	1	Verify only
0	1	0	Load RAM and execute
1	X	X	Secure

A.5 Programming Register (PROG)

This register is used to program the EPROM array. To program a byte of EPROM, set LATCH, write data to the desired address, and set EPGM for t_{EPGM}.

$001X	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:						LATCH		EPGM
Reset:	0	0	0	0	0	0	0	0

Figure A-1. EPROM Programming Register

LATCH — EPROM Latch Control Bit

This read/write bit controls the latching of the address and data buses when programming the EPROM.

1 = Address and data buses latched when the following instruction is a write to 1 of the EPROM locations. Normal reading is disabled if LATCH = 1.

0 = EPROM address and data bus configured for normal reading

EPGM — EPROM Program Control Bit

This read/write bit controls whether the programming voltage is applied to the EPROM array. For programming, this bit can be set only if the LATCH bit has been set previously. Both EPGM and LATCH cannot be set in the single write.

1 = Programming voltage applied to EPROM array
0 = Programming voltage not applied to EPROM array

NOTE: *Bits 7–3 and bit 1 MUST be set to 0 when writing to the EPROM programming register.*

EPROM Programming

Appendix B. M68HC05Cx Family Feature Comparisons

Refer to **Table B-1** for a comparison of the features for all the M68HC05C Family members.

Table B-1. M68HC05Cx Feature Comparison

	C4	C4A	705C4A	C8	C8A	705C8	705C8A	C12	C12A	C9	C9A/C9E	705C9	705C9A
USER ROM	4160	4160	—	7744	7744	—	—	12,096	12,096	15,760–15,936	15,760–15,936	—	—
USER EPROM	—	—	4160	—	—	7596–7740	7596–7740	—	—	—	—	15,760–15,936	12,096–15,936
CODE SECURITY	NO	YES	YES	NO	YES	YES	YES	NO	YES	NO	YES	NO	YES
RAM	176	176	176	176	176	176–304	176–304	176	176	176–352	176–352	176–352	176–352
OPTION REGISTER (IRQ/RAM/ SEC)	NO	NO	$1FDF (IRQ/SEC)	NO	NO	$1FDF (IRQ/RAM/ SEC)	$1FDF (IRQ/RAM/SEC)	NO	NO	$3FDF (IRQ/RAM)	$3FDF (IRQ/RAM)	$3FDF (IRQ/RAM)	$3FDF (IRQ/RAM)
MASK OPTION REGISTER(S)	NO	NO	$1FF0–$1FF1	NO	NO	NO	$1FF0–$1FF1	NO	NO	NO	NO	NO	$3FF0–$3FF1
PORTB KEYSCAN (PULLUP/ INTERRUPT)	NO	YES MASK OPTION	YES MOR SELECT-ABLE	NO	YES MASK OPTION	NO	YES MOR SELECTABLE	YES MASK OPTION	YES MASK OPTION	NO	YES MASK OPTION	NO	YES MOR SELECTABLE
PC7 DRIVE	STANDARD	HIGH CURRENT	HIGH CURRENT	STANDARD	HIGH CURRENT	STANDARD	HIGH CURRENT	HIGH CURRENT	HIGH CURRENT	STANDARD	HIGH CURRENT	STANDARD	HIGH CURRENT
PORT D	PD7, 5–0 INPUT ONLY	PD7, 5–0 INPUT ONLY	PD7, 5–0 INPUT ONLY	PD7, 5–0 INPUT ONLY	PD7, 5–0 INPUT ONLY	PD7, 5–0 INPUT ONLY	PD7, 5–0 INPUT ONLY	PD7, 5–0 INPUT ONLY	PD7, 5–0 INPUT ONLY	PD7, 5–0 BIDIREC-TIONAL	PD7, 5–0 BIDIREC-TIONAL	PD7, 5–0 BIDIREC-TIONAL	PD7, 5–0 BIDIRECTIONAL
COP	NO	YES	YES	NO	YES	YES	TWO TYPES	YES	YES	YES	YES	YES	TWO TYPES
COP ENABLE	—	MASK OPTION	MOR	—	MASK OPTION	SOFTWARE	SOFTWARE+ MOR	MASK OPTION	MASK OPTION	SOFTWARE	SOFTWARE	SOFTWARE	SOFTWARE+ MOR
COP TIMEOUT	—	64 ms (@4 MHz OSC)	64 ms (@4 MHz OSC)	—	64 ms (@4 MHz OSC)	SOFTWARE SELECTABLE	SOFTWARE+ MOR SELECTABLE	64 ms (@4 MHz OSC)	64 ms (@4MHz OSC)	SOFTWARE SELECTABLE	SOFTWARE SELECTABLE	SOFTWARE SELECTABLE	SOFTWARE+ MOR SELECTABLE
COP CLEAR	—	CLR $1FF0	CLR $1FF0	—	CLR $1FF0	WRITE $55/$AA TO $001D	WRITE $55/$AA TO $001D OR CLR $1FF0	CLR $3FF0	CLR $3FF0	WRITE $55/$AA TO $001D	WRITE $55/$AA TO $001D	WRITE $55/$AA TO $001D	WRITE $55/$AA TO $001D OR CLR $3FF0
CLOCK MONITOR	NO	NO	NO	NO	NO	YES	YES	NO	NO	YES	YES	YES	YES (C9A MODE)
ACTIVE RESET	NO	NO	NO	NO	NO	COP/CLOCK MONITOR	PROGRAM-MABLE COP/CLOCK MONITOR	NO	NO	POR/COP/ CLOCK MONITOR	POR/COP/ CLOCK MONITOR	POR/COP/ CLOCK MONITOR	POR/C9A COP/ CLOCK MONITOR
STOP DISABLE	NO	MASK OPTION	NO	NO	MASK OPTION	NO	NO	MASK OPTION	MASK OPTION	NO	NO	NO	MOR SELECTABLE (C12A MODE)

Notes:

1. The expanded RAM map (from $30–$4F and $100–$15F) available on the OTP devices MC68HC705C8 and MC68HC705C8A is not available on the ROM devices MC68HC05C8 and MC68HC05C8A.
2. The programmable COP available on the MC68HC705C8 and MC68HC705C8A is not available on the MC68HC05C8A. For ROM compatibility, use the non-programmable COP.

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217
1-303-675-2140 or 1-800-441-2447

JAPAN:

Motorola Japan Ltd.; SPS, Technical Information Center,
3-20-1, Minami-Azabu Minato-ku, Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.;
Silicon Harbour Centre, 2 Dai King Street,
Tai Po Industrial Estate, Tai Po, N.T., Hong Kong
852-26668334

TECHNICAL INFORMATION CENTER:

1-800-521-6274

HOME PAGE:

http://www.motorola.com/semiconductors

MC68HC705C9A/D

68HC705C9A : Microcontroller

The MC68HC705C9A HCMOS microcomputer is a member if the M68HC05 Family. The MC68HC705C9A is the EPROM version of the MC68HC05C9A and can also be configuerd as the EPROM version of the MC68HC05C12A. The MC68HC705C9A memory map consists of 12,092 bytes of user EPROM and 176 bytes of RAM when configured as an MC68HC05C9A. The MC68HC705C9A includes a serial communications interface, a serial peripheral interface, and a 16-bit capture/compare timer.

68HC705C9A Features

- Programmable Mask Option Register (MOR) for C9A/C12A Configuration
- Programmable MOR for Port B Pullups and Interrupts
- Popular HC05 CPU
- 15,932 Bytes of EPROM (12,092 Bytes for C12A Configuration)
- 352 Bytes of RAM (176 for C12A Configuration)
- Memory Mapped Input/Output (I/O)
- 31 Bidirectional I/O lines (24 I/O+6 Input Only for C12A Configuration) with High Current Sink and Source on PC7
- Asynchronous Serial Communications Interface (SCI)
- Synchronous Serial Peripheral Interface (SPI)
- 16-bit Capture/Compare Timer

[top]

68HC705C9A Parametrics

RAM (Bytes)	EPROM/OTP (Bytes)	Timer	I/O	Serial	Operating Voltage (V)	Bus Frequency (Max) (MHz)	Availability
352	16K	16-Bit, 1 I/C, 1 O/C	31	SCI SPI	3.3, 5.0	2.1	Now

[top]

68HC705C9A Documentation

Application Note

ID	Name	Format	Size K	Rev #	Date Last Modified	Order Availability
AN-HK-22/D	MC68HC05SR3 and MC68HC705SR3 Design Notes	pdf	3241	0	1/01/1994	-
AN-HK-23/D	MC6805R3 and MC68HC05SR3 Technical Comparison	pdf	544	0	1/01/1994	-
AN1050/D	Designing for Electromagnetic Compatibility (EMC) with HCMOS Microcontrollers	pdf	82	0	1/01/2000	
AN1055/D	M6805 16-Bit Support Macros	pdf	1048	0	1/01/1990	
AN1060SW	Software Files for AN1060 zipped	zip	176	0	1/01/1995	-
AN1066/D	Interfacing the MC68HC05C5 SIOP to an I2C Peripheral	pdf	196	1	11/01/2001	
AN1067/D	Pulse Generation and Detection with Microcontroller Units	pdf	242	1	5/31/2002	
AN1212/D	J1850 Multiplex Bus Communication Using the MC68HC705C8 and the SC371016 J1850 Communications Interface (JCI)	pdf	377	1	11/13/2001	
AN1212SW	Software files for AN1212 zipped (exe file included)	zip	3329	0	1/01/1992	-
AN1222/D	Arithmetic Waveform Synthesis with the HC05/08 MCUs	pdf	24	0	1/01/1993	
AN1222SW	Software Files for AN1222 zipped	zip	20	0	1/01/1995	-
AN1226/D	Use of the 68HC705C8A in Place of a 68HC705C8	pdf	90	4	1/01/1996	
AN1227/D	Using 9346 Series Serial EEPROMs with 6805 Series Microcontrollers	pdf	401	1	1/01/1997	
AN1227SW	Software files zipped for AN1227	zip	151	1	1/01/1997	-
AN1228/D	Interfacing the M68HC05 MCU to the MC145051 A/D Converter	pdf	160	2	1/01/1997	
AN1228SW	Software for AN1228 zippedr	zip	27	2	1/01/1997	-
AN1256/D	Interfacing the HC05 MCU to a Multichannel Digital-to-Analog Converter Using the MC68HC705C8A and the MC68HC705J1A	pdf	89	1.1	1/01/1995	
AN1256SW	Software files for AN1256 zipped	zip	80	1.1	1/01/1995	-
AN1259/D	System Design and Layout Techniques for Noise Reduction in MCU-Based Systems	pdf	78	0	1/01/1995	
AN1262/D	Simple Real-Time Kernels for M68HC05 Microcontrollers	pdf	84	0	1/01/1995	
AN1262SW	Software files for AN1262	zip	11	0	1/01/1995	-
AN1263/D	Designing for Electromagnetic Compatibility with Single-Chip Microcontrollers	pdf	104	0	1/01/1995	
AN1292/D	Adding a Voice User Interface to M68HC05 Applications	pdf	155	0	1/01/1996	
AN1292SW	Software files for AN1292 zipped	zip	215	0	1/01/1996	-
AN1298/D	Variations in the Motorola MC68HC(7)05Cx Family	pdf	200	0	1/01/1996	

AN1667/D	Software SCI Implementation to the MISC Communication Protocol	pdf	112	0	7/10/2002	
AN1667SW	Software for AN1667, zip format	zip	93	1.0	7/31/2002	-
AN1688/D	MISC Bus Slave Switch Node	pdf	243	0	7/11/2002	
AN1705/D	Noise Reduction Techniques for Microcontroller-Based Systems	pdf	67	0	1/01/1999	
AN1723/D	Interfacing MC68HC05 Microcontrollers to the IBM AT Keyboard Interface	pdf	274	0	1/01/1997	
AN1734/D	Pulse Width Modulation Using the 16-Bit Timer	pdf	102	0	1/01/1998	
AN1734SW	Software files for AN1734 zipped	zip	2	0	1/01/1997	-
AN1744/D	Resetting Microcontrollers During Power Transitions	pdf	80	0	1/01/1998	
AN1745/D	Interfacing the HC705C8A to an LCD Module	pdf	157	0	1/01/1998	
AN1745SW	Software files for AN1745 zipped	zip	3	0	1/01/1998	-
AN1752/D	Data Structures for 8-Bit Microcontrollers	pdf	213	1	5/07/2001	
AN1755/D	Interfacing the MC68HC705C8A to the DS2430A 256-Bit 1-Wire EEPROM	pdf	129	0	1/01/1998	
AN1757/D	Add a Unique Silicon Serial Number to the HC05	pdf	105	0	1/01/1998	
AN1758/D	Add Addressable Switches to the HC05	pdf	111	0	1/01/1998	
AN1761/D	Interfacing the MC68HC705C8A to the X76F041 PASS SecureFlash	pdf	179	0	1/01/1998	
AN1771/D	Precision Sine-Wave Tone Synthesis Using 8-Bit MCUs	pdf	250	0	1/01/1998	
AN1775/D	Expanding Digital Input with an A/D Converter	pdf	86	1	1/01/1998	
AN1818/D	Software SCI Routines with the 16-Bit Timer Module	pdf	84	0	1/01/1999	
AN1820/D	Software I2C Communications	pdf	55	0	1/01/1999	
AN1820SW	Software files for AN1820 zipped	zip	2	0	1/01/1998	-
AN2103/D	Local Interconnect Network (LIN) Demonstration	pdf	953	0	12/01/2000	
AN2159/D	Digital Direct Current Ignition System Using HC08 Microcontrollers	pdf	129	0	11/20/2001	
AN2159SW	AN2159SW	zip	182	1	3/08/2002	-
AN442/D	Driving LCDs with M6805 Microprocessors	pdf	1134	0	1/01/1991	
AN463/D	68HC05K0 Infra-red Remote Control	pdf	111	0	1/01/1992	
AN464/D	Software Driver Routines for the Motorola MC68HC05 CAN Module	pdf	2859	0	1/01/1993	
AN477/D	Simple A/D for MCUs without Built-In A/D Converters	pdf	224	0	1/01/1993	
AN499/D	Let the MC68HC705 Program Itself	pdf	154	0	7/01/1996	
AN991/D	Using the Serial Peripheral Interface to Communicate Between Multiple Microcomputers	pdf	251	1	1/28/2002	
ANE416/D	MC68HC05B4 Radio Synthesizer	pdf	1958	0	1/01/1988	

Brochure

ID	Name	Format	Size K	Rev #	Date Last Modified	Order	Availability
FLYREMBEDFLASH/D	Embedded Flash: Changing the Technology World for the Better	pdf	621	1	4/03/2002		

Data Sheets

ID	Name	Format	Size K	Rev #	Date Last Modified	Order	Availability
MC68HC705C9A/D	68HC705C9A Technical Data Book	pdf	1693	4.0	2/13/2002		

Engineering Bulletin

ID	Name	Format	Size K	Rev #	Date Last Modified	Order	Availability
EB166/D	System Design Considerations: Converting from the MC68HC805B6 to the MC68HC705B16 Microcontroller	pdf	757	0	1/01/1993		
EB180/D	Differences between the MC68HC705B16 and the MC68HC705B16N	pdf	20	0	1/01/1996		
EB181/D	Frequently Asked Questions and Answers for the M68HC05 Family MCAN Module	pdf	181	0	1/01/1997		
EB349/D	RAM Data Retention Considerations for Motorola Microcontrollers	pdf	45	1	6/22/2000		
EB396/D	Use of OSC2/XTAL as a Clock Output on Motorola Microcontrollers	pdf	49	0	6/19/2002		
EB413/D	Resetting MCUs	pdf	62	0	1/01/2000		
EB421/D	The Motorola MCAN Module	pdf	78	0	2/23/2000		

Errata

ID	Name	Format	Size K	Rev #	Date Last Modified	Order	Availability
68HC705C9AMSE1/D	68HC705C9A Device Information Sheet: 2F63J/3F63J Mask Sets	pdf	13	0	1/01/1996		-

Product Change Notices

ID	Name	Format	Size K	Rev #	Date Last Modified	Order	Availability
PCN7701	QFP 10X10 ASSY MOVE FROM SHC TO BAT3	htm	16	-	7/09/2002		-
PCN7899	44/52/68 PLCC ASSY MOVE FROM SDI TO KLM	htm	31	0	8/14/2002		-

Reference Manual

ID	Name	Format	Size K	Rev #	Date Last Modified	Order	Availability
M68HC05AG/AD	M68HC05 Applications Guide	pdf	3272	4	3/18/2002		-

M68HC05TB/D	HC05 Family - Understanding Small Microcontrollers	pdf	2866	2	1/01/1998	
MC68HC05CXRG/D	MC68HC05C4, C8, C9, MC68HC705C8, MC68HC805C4, MC68HCL05C4, C8, MC68HSC05C4, C8 Programming Reference	pdf	3150	1	2/23/2000	-

Roadmap

ID	Name	Format	Size K	Rev #	Date Last Modified	Order Availability
8BITMCURDMAP	8-Bit MCU Family Roadmap	pdf	35	-	-	-

Selector Guide

ID	Name	Format	Size K	Rev #	Date Last Modified	Order Availability
SG1006/D	Microcontrollers SPS Sales Guide	pdf	600	0	9/26/2002	
SG1011/D	Software and Development Tools Sales Guide	pdf	259	1	9/26/2002	
SG2000CR/D	Application Selector Guide Index and Cross-Reference.	pdf	62	0	6/24/2002	

[top]

68HC705C9A Development Tools/Boards

ID	Name	Vendor ID	Order Availability
M68ICS05C	In-Circuit Simulator for the 68HC05C and 68HC705C	MOTOROLA	
M68EM05C9A	Emulation Module	MOTOROLA	
KITMMDS05C	Modular Development System (MMDS) Kits	MOTOROLA	
KITMMEVS05C	Modular Evaluation System (MMEVS)	MOTOROLA	
CWHC05	CodeWarrior Development Tools for HC05	METROWERKS	

[top]

Design Tools

Software

ID	Name	Vendor ID	Format	Size K	Rev #
HC705C9AH	C header file for 68HC705C9A	MOTOROLA	zip	10	-

MATH16ACOD	General Math routines	asm	4	-
MATHAGBCOD	General Math routines	asm	6	-

Software Tools/Assemblers

ID	Name	Vendor ID	Format	Size K	Rev #
ASHC5ASM	DOS based freeware assembler	MOTOROLA	arc	55	0

Software Tools/Emulation

ID	Name	Vendor ID	Format	Size K	Rev #
M68HC05C9EM	M68HC05C9EM Self extracting PC file. Emulation module configuration/help file for MMDS and MMEVS.	MOTOROLA	exe	52	1

Software/Application Software/Code Examples

ID	Name	Vendor ID	Format	Size K	Rev #
C8THERMSW	HC05 Software Example: Home Thermostat example using the 705C8 with indoor/outdoor temperature and time of day	MOTOROLA	zip	11	-
FLOAT05COD	Floating Point routines	MOTOROLA	zip	13	-
HC05DELAYSW	HC05 Software Example: Subroutine that delays for a whole number of milliseconds	MOTOROLA	zip	2	-
HC05EXSW	Library containing software examples in assembly for 68HC05	MOTOROLA	zip	45	-
HC05KEYINTSW	HC05 Software Examples: Using keyboard interrupts and decoding a matrix keypad	MOTOROLA	zip	7	-
HC05KEYPADSW	HC05 Software Example: Keypad debounce and decode. When a key is found, it is changed to ASCII and displayed on an LCD	MOTOROLA	zip	2	-
HC05LCDSW	HC05 Software Example: Initializes an LCD and displays ABCDEF...S	MOTOROLA	zip	1	-
HC05SCISW	HC05 Software Example: Serial Communications Interface example	MOTOROLA	zip	1	-
HC05SPISW	HC05 Software Example: Serial Peripheral Interface example	MOTOROLA	zip	1	-
HC05SWITCHSW	HC05 Software Example: Simple program that reads the state of a switch on a general-purpose I/O pin and lights an LED based on the state of the switch	MOTOROLA	zip	1	-
HC05TIMERSW	HC05 Software Example: Using the 68HC05 16-bit Timer	MOTOROLA	zip	1	-
J1APWMSW	HC05 Software Example: Low frequency PWM example using the 68HC705J1A real-time interrupt and timer overflow interrupt	MOTOROLA	zip	1	-
J1AUARTSW	HC05 Software example: Software UART example that transmits and receives data on the 68HC705J1A	MOTOROLA	zip	2	-
K1THERMSW	HC05 Software Example: Thermometer project using the 68HC705K1	MOTOROLA	zip	5	-

SAMPPROGCOD	Example routines	MOTOROLA	exe	35	-
THERM-CCOD	Thermometer example in C	MOTOROLA	zip	11	-

Software/Operating Systems

ID	Name	Vendor ID	Format	Size K	Rev #
PE68HC05SIM	Windows upgrades for P&E's simulator software for 68HC05	PEMICRO	html	0	-

[top]

Orderable Parts Information

PartNumber	Package Info	Life Cycle Description (code)	Remarks	Budgetary Price QTY 1000+ ($US)	Order Availability
KMC705C9ACFB	10x10 mm Quad Flat Pack (QFP)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$4.95	
KMC705C9ACFN	Plastic Leaded Chip Carrier (PLCC)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$4.95	
KMC705C9ACP	Plastic Dual-in-Line (PDIP)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$4.95	
KMC705C9ACS	Shrink Plastic Dual-in-Line-window (SDIP)	PROD PHASE OUT/SEE LAST ORD DT(6) 30 Nov 2002	-40 to +85 C	$35.00	-
KMC705C9AFS	Ceramic Leaded - window (CLCC)	PROD PHASE OUT/SEE LAST ORD DT(6) 30 Nov 2002	0 to +70 C	$35.00	-
MC68HC705C9ACB	42-Pin Shrink Dual In-Line Package (SDIP)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$4.95	

MC68HC705C9ACFB	44-Pin Quad Flat Pack (QFP)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$4.95	
MC68HC705C9ACFN	44-Lead Plastic Leaded Chip Carrier (PLCC)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$4.95	
MC68HC705C9ACFS	Ceramic Leaded - window (CLCC)	PROD PHASE OUT/SEE LAST ORD DT(6) 30 Nov 2002	-40 to +85 C	-	-
MC68HC705C9ACP	40-Pin Plastic Dual In-Line Package (DIP)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$4.95	
MC68HC705C9ACS	Shrink Plastic Dual-in-Line-window (SDIP)	PROD PHASE OUT/SEE LAST ORD DT(6) 30 Nov 2002	-40 to +85 C	$35.00	
MC68HC705C9AFB	10x10 mm Quad Flat Pack (QFP)	PRODUCT MATURITY/SATURATION(4)	0 to +70 C	$4.95	
MC68HC705C9AFN	Plastic Leaded Chip Carrier (PLCC)	PRODUCT MATURITY/SATURATION(4)	0 to +70 C	$4.95	
MC68HC705C9AFS	Ceramic Leaded - window (CLCC)	PROD PHASE OUT/SEE LAST ORD DT(6) 30 Nov 2002	0 to +70 C	-	-
MC68LNC705C9ACFN	Plastic Leaded Chip Carrier (PLCC)	PRODUCT MATURITY/SATURATION(4)	-40 to +85 C	$5.44	
KMC705C9ACB	-	PRODUCT MATURITY/SATURATION(4)	-	-	-
MC68HC705C9AVFN	-	PRODUCT MATURITY/SATURATION(4)	-	$5.19	

68HC705C9A : Microcontroller

The MC68HC705C9A HCMOS microcomputer is a member if the M68HC05 Family. The MC68HC705C9A is the EPROM version of the MC68HC05C9A and can also be configuerd as the EPROM version of the MC68HC05C12A. The MC68HC705C9A memory map consists of 12,092 bytes of user EPROM and 176 bytes of RAM when configured as an MC68HC05C9A. The MC68HC705C9A includes a serial communications interface, a serial peripheral interface, and a 16-bit capture/compare timer.

Page Contents:
- Features
- Documentation
- Tools
- Orderable Parts
- Related Links

Other Info:
- FAQs
- 3rd Party Design Help
- 3rd Party Tool Vendors
- 3rd Party Trainers

Block Diagram

68HC705C9A Features

- Programmable Mask Option Register (MOR) for C9A/C12A Configuration
- Programmable MOR for Port B Pullups and Interrupts
- Popular HC05 CPU
- 15,932 Bytes of EPROM (12,092 Bytes for C12A Configuration)
- 352 Bytes of RAM (176 for C12A Configuration)
- Memory Mapped Input/Output (I/O)
- 31 Bidirectional I/O lines (24 I/O+6 Input Only for C12A Configuration) with High Current Sink and Source on PC7
- Asynchronous Serial Communications Interface (SCI)
- Synchronous Serial Peripheral Interface (SPI)
- 16-bit Capture/Compare Timer

Return to Top

68HC705C9A Documentation

Documentation

Application Note

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
AN-HK-22	MC68HC05SR3 and MC68HC705SR3 Design Notes	MOTOROLA	pdf	0	0	1/01/1994	ORDER
AN-HK-23	MC6805R3 and MC68HC05SR3 Technical Comparison	MOTOROLA	pdf	0	0	1/01/1994	ORDER
AN1050_D	Designing for Electromagnetic Compatibility (EMC) with HCMOS Microcontrollers	MOTOROLA	pdf	82	0	1/01/2000	-
AN1055/D	M6805 16-Bit Support Macros	MOTOROLA	pdf	1048	0	1/01/1990	ORDER
AN1060SW	Software Files for AN1060 zipped	MOTOROLA	zip	176	0	1/01/1995	-
AN1066/D	Interfacing the MC68HC05C5 SIOP to an I2C Peripheral	MOTOROLA	pdf	196	1	11/01/2001	ORDER
AN1067/D	Pulse Generation and Detection with Microcontroller Units	MOTOROLA	pdf	242	1	5/31/2002	ORDER

AN1212/D	J1850 Multiplex Bus Communication Using the MC68HC705C8 and the SC371016 J1850 Communications Interface (JCI)	MOTOROLA	pdf	377	1	11/13/2001	ORDER
AN1212SW	Software files for AN1212 zipped (exe file included)	MOTOROLA	zip	3329	0	1/01/1992	-
AN1222/D	Arithmetic Waveform Synthesis with the HC05/08 MCUs	MOTOROLA	pdf	24	0	1/01/1993	ORDER
AN1222SW	Software Files for AN1222 zipped	MOTOROLA	zip	20	0	1/01/1995	-
AN1226/D	Use of the 68HC705C8A in Place of a 68HC705C8	MOTOROLA	pdf	90	4	1/01/1996	ORDER
AN1227/D	Using 9346 Series Serial EEPROMs with 6805 Series Microcontrollers	MOTOROLA	pdf	401	1	1/01/1997	ORDER
AN1227SW	Software files zipped for AN1227	MOTOROLA	zip	151	1	1/01/1997	-
AN1228/D	Interfacing the M68HC05 MCU to the MC145051 A/D Converter	MOTOROLA	pdf	160	2	1/01/1997	ORDER
AN1228SW	Software for AN1228 zippedr	MOTOROLA	zip	27	2	1/01/1997	-
AN1256/D	Interfacing the HC05 MCU to a Multichannel Digital-to-Analog Converter Using the MC68HC705C8A and the MC68HC705J1A	MOTOROLA	pdf	89	1.1	1/01/1995	ORDER
AN1256SW	Software files for AN1256 zipped	MOTOROLA	zip	80	1.1	1/01/1995	-
AN1259/D	System Design and Layout Techniques for Noise Reduction in MCU-Based Systems	MOTOROLA	pdf	78	0	1/01/1995	ORDER
AN1262/D	Simple Real-Time Kernels for M68HC05 Microcontrollers	MOTOROLA	pdf	84	0	1/01/1995	ORDER
AN1262SW	Software files for AN1262	MOTOROLA	zip	11	0	1/01/1995	-
AN1263/D	Designing for Electromagnetic Compatibility with Single-Chip Microcontrollers	MOTOROLA	pdf	104	0	1/01/1995	ORDER
AN1292/D	Adding a Voice User Interface to M68HC05 Applications	MOTOROLA	pdf	155	0	1/01/1996	ORDER
AN1292SW	Software files for AN1292 zipped	MOTOROLA	zip	215	0	1/01/1996	-
AN1298/D	Variations in the Motorola MC68HC(7)05Cx Family	MOTOROLA	pdf	200	0	1/01/1996	ORDER
AN1516/D	Liquid Level Control Using a Motorola Pressure Sensor	MOTOROLA	pdf	77	2	1/24/2003	ORDER
AN1667/D	Software SCI Implementation to the MISC Communication Protocol	MOTOROLA	pdf	112	0	7/10/2002	ORDER
AN1667SW	Software for AN1667, zip format	MOTOROLA	zip	93	1.0	7/31/2002	-
AN1688/D	MISC Bus Slave Switch Node	MOTOROLA	pdf	243	0	7/11/2002	ORDER
AN1705/D	Noise Reduction Techniques for Microcontroller-Based Systems	MOTOROLA	pdf	67	0	1/01/1999	ORDER
AN1723/D	Interfacing MC68HC05 Microcontrollers to the IBM AT Keyboard Interface	MOTOROLA	pdf	274	0	1/01/1997	ORDER
AN1734/D	Pulse Width Modulation Using the 16-Bit Timer	MOTOROLA	pdf	102	0	1/01/1998	ORDER
AN1734SW	Software files for AN1734 zipped	MOTOROLA	zip	2	0	1/01/1997	-
AN1744/D	Resetting Microcontrollers During Power Transitions	MOTOROLA	pdf	80	0	1/01/1998	ORDER
AN1745/D	Interfacing the HC705C8A to an LCD Module	MOTOROLA	pdf	157	0	1/01/1998	ORDER

AN1745SW	Software files for AN1745 zipped	MOTOROLA	zip	3	0	1/01/1998	-
AN1752/D	Data Structures for 8-Bit Microcontrollers	MOTOROLA	pdf	213	1	5/07/2001	ORDER
AN1755/D	Interfacing the MC68HC705C8A to the DS2430A 256-Bit 1-Wire EEPROM	MOTOROLA	pdf	129	0	1/01/1998	ORDER
AN1757/D	Add a Unique Silicon Serial Number to the HC05	MOTOROLA	pdf	105	0	1/01/1998	ORDER
AN1758/D	Add Addressable Switches to the HC05	MOTOROLA	pdf	111	0	1/01/1998	ORDER
AN1761/D	Interfacing the MC68HC705C8A to the X76F041 PASS SecureFlash	MOTOROLA	pdf	179	0	1/01/1998	ORDER
AN1771/D	Precision Sine-Wave Tone Synthesis Using 8-Bit MCUs	MOTOROLA	pdf	250	0	1/01/1998	ORDER
AN1775/D	Expanding Digital Input with an A/D Converter	MOTOROLA	pdf	86	1	1/01/1998	ORDER
AN1818/D	Software SCI Routines with the 16-Bit Timer Module	MOTOROLA	pdf	84	0	1/01/1999	ORDER
AN1820/D	Software I2C Communications	MOTOROLA	pdf	55	0	1/01/1999	ORDER
AN1820SW	Software files for AN1820 zipped	MOTOROLA	zip	2	0	1/01/1998	-
AN2103/D	Local Interconnect Network (LIN) Demonstration	MOTOROLA	pdf	953	0	12/01/2000	ORDER
AN2159/D	Digital Direct Current Ignition System Using HC08 Microcontrollers	MOTOROLA	pdf	129	0	11/20/2001	ORDER
AN2159SW	AN2159SW	MOTOROLA	zip	182	1	3/08/2002	-
AN442/D	Driving LCDs with M6805 Microprocessors	MOTOROLA	pdf	1134	0	1/01/1991	ORDER
AN463/D	68HC05K0 Infra-red Remote Control	MOTOROLA	pdf	111	0	1/01/1992	ORDER
AN464/D	Software Driver Routines for the Motorola MC68HC05 CAN Module	MOTOROLA	pdf	2859	0	1/01/1993	ORDER
AN477/D	Simple A/D for MCUs without Built-In A/D Converters	MOTOROLA	pdf	224	0	1/01/1993	ORDER
AN499/D	Let the MC68HC705 Program Itself	MOTOROLA	pdf	154	0	7/01/1996	ORDER
AN991/D	Using the Serial Peripheral Interface to Communicate Between Multiple Microcomputers	MOTOROLA	pdf	251	1	1/28/2002	ORDER
ANE416/D	MC68HC05B4 Radio Synthesizer	MOTOROLA	pdf	1958	0	1/01/1988	ORDER

Brochure

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
BR68HC08FAMAM/D	68HC08 Family: High Performance and Flexibility	MOTOROLA	pdf	57	2	5/21/2003	ORDER
FLYREMBEDFLASH/D	Embedded Flash: Changing the Technology World for the Better	MOTOROLA	pdf	68	2	5/21/2003	ORDER

Data Sheets

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
MC68HC705C9A/D	68HC705C9A Technical Data Book	MOTOROLA	pdf	1693	4.0	2/13/2002	ORDER

Engineering Bulletin

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
EB166/D	System Design Considerations: Converting from the MC68HC805B6 to the MC68HC705B16 Microcontroller	MOTOROLA	pdf	757	0	1/01/1993	ORDER
EB180/D	Differences between the MC68HC705B16 and the MC68HC705B16N	MOTOROLA	pdf	20	0	1/01/1996	ORDER
EB181/D	Frequently Asked Questions and Answers for the M68HC05 Family MCAN Module	MOTOROLA	pdf	181	0	1/01/1997	ORDER
EB349/D	RAM Data Retention Considerations for Motorola Microcontrollers	MOTOROLA	pdf	45	1	6/22/2000	ORDER
EB396/D	Use of OSC2/XTAL as a Clock Output on Motorola Microcontrollers	MOTOROLA	pdf	49	0	6/19/2002	ORDER
EB413/D	Resetting MCUs	MOTOROLA	pdf	62	0	1/01/2000	ORDER
EB421/D	The Motorola MCAN Module	MOTOROLA	pdf	78	0	2/23/2000	ORDER

Errata - Click here for important errata information

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
68HC705C9AMSE1/D	68HC705C9A Device Information Sheet: 2F63J/3F63J Mask Sets	MOTOROLA	pdf	13	0	1/01/1996	-

Fact Sheets

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
CWDEVSTUDFACTHC08	Development Studio	MOTOROLA	pdf	48	2	5/13/2002	-

Product Change Notices

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
PCN7701	QFP 10X10 ASSY MOVE FROM SHC TO BAT3	MOTOROLA	htm	16	-	7/09/2002	-
PCN7899	44/52/68 PLCC ASSY MOVE FROM SDI TO KLM	MOTOROLA	htm	31	0	8/14/2002	-
PCN8901	BINDING STRAP CHANGE FOR QFP PRODUCTS	MOTOROLA	htm	5	0	5/21/2003	-

Reference Manual

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
M68HC05TB/D	HC05 Family - Understanding Small Microcontrollers	MOTOROLA	pdf	2866	2	1/01/1998	ORDER
MC68HC05CXRG/D	MC68HC05C4, C8, C9, MC68HC705C8, MC68HC805C4, MC68HCL05C4, C8, MC68HSC05C4, C8 Programming Reference	MOTOROLA	pdf	3150	1	2/23/2000	-

Roadmap

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
8BITMCURD	8-Bit MCU Family Roadmap	MOTOROLA	pdf	30	0	9/01/2002	-

Selector Guide

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
SG1002	Analog Selector Guide - Quarter 4, 2003	MOTOROLA	pdf	579	0	10/24/2003	ORDER
SG1006	Microcontrollers Selector Guide - Quarter 4, 2003	MOTOROLA	pdf	826	0	10/24/2003	ORDER
SG1010	Sensors Selector Guide - Quarter 4, 2003	MOTOROLA	pdf	219	0	10/24/2003	ORDER
SG1011	Software and Development Tools Selector Guide - Quarter 4, 2003	MOTOROLA	pdf	287	0	10/24/2003	ORDER
SG2000CR	Application Selector Guide Index and Cross-Reference.	MOTOROLA	pdf	95	3	11/11/2003	ORDER
SG2039	Application Selector Guide - Vacuum Cleaners Vacuum Cleaners	MOTOROLA	pdf	0	0	6/17/2003	ORDER

Return to Top

68HC705C9A Tools

Hardware Tools

Adapters

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
PA44-P(Z)P	Prototyping Adapter	LOGSYS	-	-	-	-
PA705C8-DP(-PP)	Prototyping Adapter	LOGSYS	-	-	-	-
PA705C8-DX-QF	Prototyping Adapter	LOGSYS	-	-	-	-
PA705C8-PD	Programming Adapter	LOGSYS	-	-	-	-
PA705C8-QD-16	Programming Adapter	LOGSYS	-	-	-	-

Emulators/Probes/Wigglers

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
AX-6811	AX-6811	HITEX	-	-	-	-
IC10000	iC1000 PowerEmulator	ISYS	-	-	-	-
IC20000	iC2000 PowerEmulator	ISYS	-	-	-	-
IC40000	iC4000 ActiveEmulator	ISYS	-	-	-	-

Evaluation/Development Boards and Systems

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
KITMMDS05C	Modular Development System (MMDS) Kits	MOTOROLA	-	-	-	BUY
KITMMEVS05C	Modular Evaluation System (MMEVS)	MOTOROLA	-	-	-	BUY
M68CBL05B	Low-noise Flex Cable	MOTOROLA	-	-	-	BUY
M68CBL05C	Low-noise Flex Cable	MOTOROLA	-	-	-	BUY
M68EM05C9A	Emulation Module	MOTOROLA	-	-	-	BUY
M68ICS05C	In-Circuit Simulator for the 68HC05C and 68HC705C	MOTOROLA	-	-	-	BUY

Programmers

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
MP8011A	Gang Programmer Base Unit	SOFTEC	-	-	-	-
POWERLAB	Universal Programmer	SYSGEN	-	-	-	-

Software

Application Software

Application Development Framework

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
HC705C9AH	C header file for 68HC705C9A	MOTOROLA	zip	10	-	-

Code Examples

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
C8THERMSW	HC05 Software Example: Home Thermostat example using the 705C8 with indoor/outdoor temperature and time of day	MOTOROLA	zip	11	-	-
FLOAT05COD	Floating Point routines	MOTOROLA	zip	13	-	-
HC05DELAYSW	HC05 Software Example: Subroutine that delays for a whole number of milliseconds	MOTOROLA	zip	2	-	-
HC05EXSW	Library containing software examples in assembly for 68HC05	MOTOROLA	zip	45	-	-
HC05KEYINTSW	HC05 Software Examples: Using keyboard interrupts and decoding a matrix keypad	MOTOROLA	zip	7	-	-
HC05KEYPADSW	HC05 Software Example: Keypad debounce and decode. When a key is found, it is changed to ASCII and displayed on an LCD	MOTOROLA	zip	2	-	-
HC05LCDSW	HC05 Software Example: Initializes an LCD and displays ABCDEF...S	MOTOROLA	zip	1	-	-
HC05SCISW	HC05 Software Example: Serial Communications Interface example	MOTOROLA	zip	1	-	-
HC05SPISW	HC05 Software Example: Serial Peripheral Interface example	MOTOROLA	zip	1	-	-
HC05SWITCHSW	HC05 Software Example: Simple program that reads the state of a switch on a general-purpose I/O pin and lights an LED based on the state of the switch	MOTOROLA	zip	1	-	-
HC05TIMERSW	HC05 Software Example: Using the 68HC05 16-bit Timer	MOTOROLA	zip	1	-	-
J1APWMSW	HC05 Software Example: Low frequency PWM example using the 68HC705J1A real-time interrupt and timer overflow interrupt	MOTOROLA	zip	1	-	-
J1AUARTSW	HC05 Software example: Software UART example that transmits and receives data on the 68HC705J1A	MOTOROLA	zip	2	-	-
K1THERMSW	HC05 Software Example: Thermometer project using the 68HC705K1	MOTOROLA	zip	5	-	-
MATH16ACOD	General Math routines	MOTOROLA	asm	4	-	-
MATHAGBCOD	General Math routines	MOTOROLA	asm	6	-	-
SAMPPROGCOD	Example routines	MOTOROLA	exe	35	-	-
THERM-CCOD	Thermometer example in C	MOTOROLA	zip	11	-	-

Software Tools

Assemblers

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
ASHC5ASM	DOS based freeware assembler	MOTOROLA	arc	55	0	-
AX6805	AX6805 relocatable/absolute macro assembler for HC05	COSMIC	-	-	-	-

Compilers

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
CWHC05	CodeWarrior Development Tools for HC05	METROWERKS	-	-	-	BUY
CX6805	CX6805 C Cross Compiler for HC05	COSMIC	-	-	-	-

Debuggers

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
CWHC05	CodeWarrior Development Tools for HC05	METROWERKS	-	-	-	BUY
ZAP 6805 MMDS	ZAP 6805 MMDS Debugger	COSMIC	-	-	-	-
ZAP 6805 SIM	ZAP 6805 Simulator Debugger	COSMIC	-	-	-	-
AX-6811	AX-6811	HITEX	-	-	-	-

Emulation

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
M68HC05C9EM	M68HC05C9EM Self extracting PC file. Emulation module configuration/help file for MMDS and MMEVS.	MOTOROLA	exe	52	1	-

IDE (Integrated Development Environment)

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
CWHC05	CodeWarrior Development Tools for HC05	METROWERKS	-	-	-	BUY
IDEA05	IDEA05 integrated development environment for HC05	COSMIC	-	-	-	-
IC-SW-OPR	winIDEA	ISYS	-	-	-	-

Models

Instruction Set Simulator

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
PE68HC05SIM	Windows upgrades for P&E's simulator software for 68HC05	PEMICRO	html	0	-	-

Performance and Testing

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
AX-6811	AX-6811	HITEX	-	-	-	-

Return to Top

Orderable Parts Information

PartNumber	Package Info	Tape and Reel	Life Cycle Description (code)	Budgetary Price QTY 1000+ ($US)	Additional Info	Order Availability

KMC705C9ACB	PSDIP 42	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	-
KMC705C9ACFB	QFP 44 10*10*2.0P0.8	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	BUY
KMC705C9ACFN	PLCC 44	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	BUY
KMC705C9ACP	PDIP 40	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	BUY
MC68HC705C9ACB	PSDIP 42	No	PRODUCT MATURITY/SATURATION(4)	$4.95	more	BUY
MC68HC705C9ACFB	QFP 44 10*10*2.0P0.8	No	PRODUCT MATURITY/SATURATION(4)	$4.95	more	BUY
MC68HC705C9ACFN	PLCC 44	No	PRODUCT MATURITY/SATURATION(4)	$4.95	more	BUY
MC68HC705C9ACFS	CQUAD-J 44 WD	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	-
MC68HC705C9ACP	PDIP 40	No	PRODUCT MATURITY/SATURATION(4)	$4.95	more	BUY
MC68HC705C9ACS	CDIP 40 WINDOW	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	-
MC68HC705C9AFB	QFP 44 10*10*2.0P0.8	No	PRODUCT MATURITY/SATURATION(4)	$4.95	more	BUY
MC68HC705C9AFN	PLCC 44	No	PRODUCT MATURITY/SATURATION(4)	$4.95	more	BUY
MC68HC705C9AVFN	PLCC 44	No	PRODUCT MATURITY/SATURATION(4)	$5.19	more	BUY
MC68LNC705C9ACFN	PLCC 44	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	BUY

NOTE: Are you looking for an obsolete orderable part? Click **HERE** to check our distributors' inventory.

Return to Top

Related Links

- Automotive
- Microcontrollers
- Motor Control

Return to Top