68HC805P18

SPECIFICATION
(General Release)

© December 7, 1995

CSIC System Design Group
Austin, Texas

TABLE OF CONTENTS

Paragraph   **Title**   **Page**

SECTION 1
INTRODUCTION

SECTION 2
MEMORY

SECTION 3
CENTRAL PROCESSING UNIT

TABLE OF CONTENTS

Paragraph	Title	Page

SECTION 4
INTERRUPTS

SECTION 5
RESETS

SECTION 6
OPERATING MODES

TABLE OF CONTENTS

Paragraph	Title	Page

SECTION 7
INPUT/OUTPUT PORTS

SECTION 8
EEPROM

SECTION 9
ANALOG-TO-DIGITAL CONVERTER

TABLE OF CONTENTS

Paragraph   **Title**   **Page**

SECTION 10
16-BIT TIMER

SECTION 11
SERIAL INPUT/OUTPUT PORT

SECTION 12
INSTRUCTION SET

TABLE OF CONTENTS

Paragraph	Title	Page

SECTION 13
ELECTRICAL SPECIFICATIONS

SECTION 14
MECHANICAL SPECIFICATIONS

SECTION 15
ORDERING INFORMATION

APPENDIX A
EMULATION

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

SECTION 1
INTRODUCTION

1.1 Introduction

The Motorola MC68HC805P18 microcontroller is a member of the M68HC05 microcontroller family with a 4-channel, 8-bit analog-to-digital (A/D) converter, a 16-bit timer with output compare and input capture, a serial communications port (SIOP), a computer operating properly (COP) watchdog timer, and 21 input/output (I/O) pins (20 bidirectional, one input-only). The memory map contains 192 bytes of RAM, 8064 bytes of program EEPROM (for user code), 512 bytes of boot ROM, and128 bytes of EEPROM (for data storage). This device is available in a 28-pin dual in-line package (DIP) or a small outline (SOIC) package. A functional block diagram of the MC68HC805P18 is shown in Figure 1-1.

1.2 Features

- Low-cost HC05 core running at 2 MHz bus speed

- 28-pin DIP or SOIC package

- 4 MHz on-chip crystal/ceramic resonator oscillator

- 8064 bytes of user EEPROM including 48 bytes of page zero EEPROM and 16 bytes of user vectors

- 192 bytes of on-chip RAM

- 128 bytes of EEPROM

- Low-voltage reset

- 4-channel, 8-bit A/D converter

- SIOP serial communications port

- COP watchdog timer with active pulldown on $\overline{\text{RESET}}$

- 16-bit timer with output compare and input capture

- 20 bidirectional I/O lines and one input-only line

- High current sink and source on two I/O pins (PC0 and PC1)
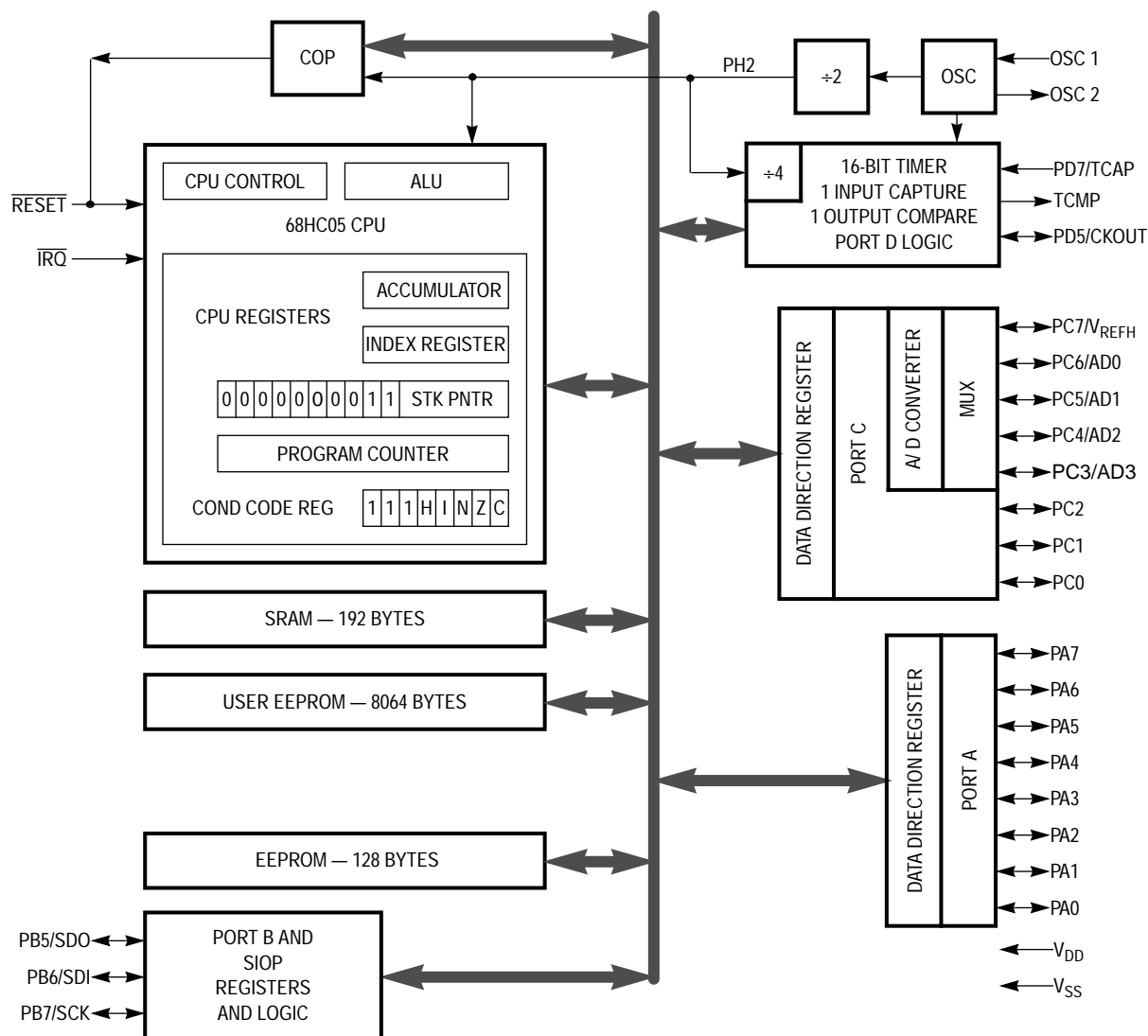
Figure 1-1. Block Diagram

INTRODUCTION

1.3 Mask Options

EEPROM mask option register (MOR) selectable options include the following. For additional information, refer to **8.4 Mask Option Registers (MOR)**.

- $\overline{\text{IRQ}}$ is edge- and level-sensitive or edge-sensitive only.

- SIOP most significant bit (MSB) first or least significant bit (LSB) first

- SIOP clock rate set to oscillator divided by 2, 4, 8, or 16

- COP watchdog timer enabled or disabled

- Stop instruction enabled or converted to halt mode

- Option to enable clock output pin to replace PD5

- Option to individually enable pullups/interrupts on each of the eight port A pins

- LVR reset enabled or disabled

NOTE

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

Any reference to voltage, current, or frequency specified in the following sections will refer to the nominal values. The exact values and their tolerance or limits are specified in **SECTION 13 ELECTRICAL SPECIFICATIONS.**

1.4 Functional Pin Description

The following paragraphs describe the functionality of each pin on the MC68HC805P18 package. Pins connected to subsystems described in other sections provide a reference to the section instead of a detailed functional description.The pinout is shown in Figure 1-2.

```
     RESET ▭ 1 ○         28 ▭ V_DD
       IRQ ▭ 2           27 ▭ OSC1
       PA7 ▭ 3           26 ▭ OSC2
       PA6 ▭ 4           25 ▭ PD7/TCAP
       PA5 ▭ 5           24 ▭ TCMP
       PA4 ▭ 6           23 ▭ PD5/CKOUT
       PA3 ▭ 7           22 ▭ PC0
       PA2 ▭ 8           21 ▭ PC1
       PA1 ▭ 9           20 ▭ PC2
       PA0 ▭ 10          19 ▭ PC3/AD3
   SDO/PB5 ▭ 11          18 ▭ PC4/AD2
   SDI/PB6 ▭ 12          17 ▭ PC5/AD1
   SCK/PB7 ▭ 13          16 ▭ PC6/AD0
      V_SS ▭ 14          15 ▭ PC7/V_REFH
```
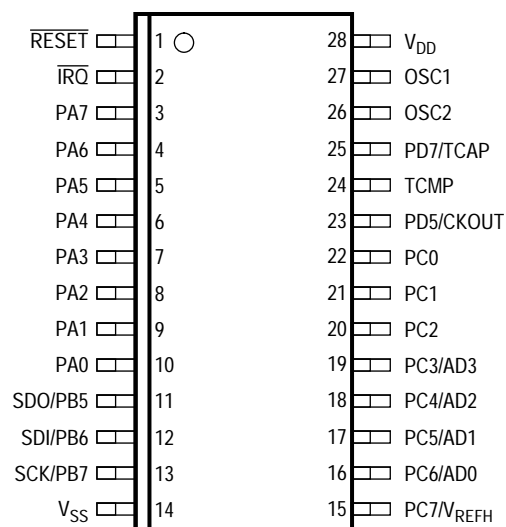
Figure 1-2. User Mode Pinout

1.4.1 V_{DD} and V_{SS}

Power is supplied to the MCU through V_{DD} and V_{SS}. V_{DD} is connected to a regulated positive supply and V_{SS} is connected to ground.

Very fast signal transitions occur on the MCU pins. The short rise and fall times place very high short-duration current demands on the power supply. To prevent noise problems, take special care to provide good power supply bypassing at the MCU. Use bypass capacitors with good high-frequency characteristics, and position them as close to the MCU as possible. Bypassing requirements vary, depending on how heavily the MCU pins are loaded.

1.4.2 OSC1 and OSC2

The OSC1 and OSC2 pins are the control connections for the on-chip oscillator. The OSC1 and OSC2 pins can accept the following:

1. A crystal as shown in Figure 1-3(a)

2. A ceramic resonator as shown in Figure 1-3(a)

3. An external clock signal as shown in Figure 1-3(b)

The frequency, f_{osc}, of the oscillator or external clock source is divided by two to produce the internal PH2 bus clock operating frequency, f_{op}. The oscillator cannot be turned off by software if the stop-to-halt conversion is enabled via mask option register 1. Refer to **8.4 Mask Option Registers (MOR)**.

1.4.3 Crystal

The circuit in Figure 1-3(a) shows a typical oscillator circuit for an AT-cut, parallel resonant crystal. Follow the crystal manufacturer's recommendations, as the crystal parameters determine the external component values required to provide maximum stability and reliable startup. The load capacitance values used in the oscillator circuit design should include all stray capacitances. Mount the crystal and components as close as possible to the pins for startup stabilization and to minimize output distortion.

Ceramic Resonator

In cost-sensitive applications, use a ceramic resonator instead of a crystal. Use the circuit in Figure 1-3(a) for a ceramic resonator and follow the resonator manufacturer's recommendations, as the resonator parameters determine the external component values required for maximum stability and reliable starting. The load capacitance values used in the oscillator circuit design should include all stray capacitances. Mount the resonator and components as close as possible to the pins for startup stabilization and to minimize output distortion.

External Clock

An external clock from another CMOS-compatible device can be connected to the OSC1 input, with the OSC2 input not connected, as shown in Figure 1-3(b)**.**
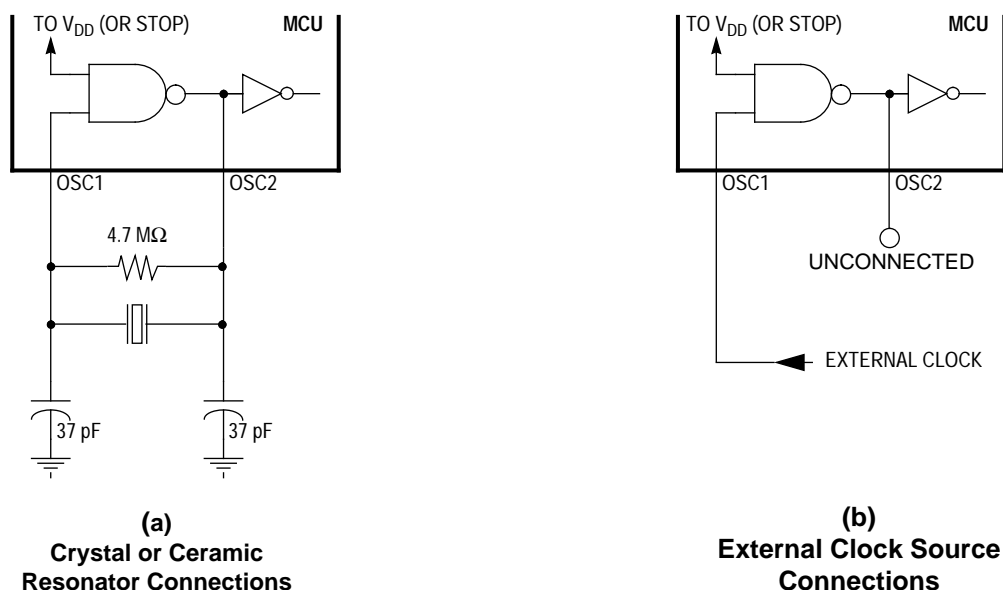
Figure 1-3. Oscillator Connections

1.4.4 Reset ($\overline{\text{RESET}}$)

Driving this input low will reset the MCU to a known startup state.   As an output, the $\overline{\text{RESET}}$ pin indicates that an internal MCU reset has occurred. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger to improve its noise immunity. Refer to **SECTION 5 RESETS**.

1.4.5 Port A (PA0 through PA7)

These eight I/O pins comprise port A. The state of any pin is software programmable and all port A lines are configured as inputs during power-on or reset. The pullups and interrupt options (active low) on the port A pins can be individually programmed in the mask option register 2 (MOR2). For further information, refer to **SECTION 4 INTERRUPTS** and **SECTION 7 INPUT/OUTPUT PORTS**.

1.4.6 Port B (PB5/SDO, PB6/SDI, and PB7/SCK)

These three I/O pins comprise port B and are shared with the SIOP communications subsystem. The state of any pin is software programmable and all port B lines are configured as inputs during power-on or reset. For further information, refer to **SECTION 7 INPUT/OUTPUT PORTS** and **SECTION 11 SERIAL INPUT/OUTPUT PORT**.

1.4.7 Port C (PC0–PC2, PC3/AD3, PC4/AD2, PC5/AD1, PC6/AD0, and PC7/V$_{REFH}$)

These eight I/O pins comprise port C and are shared with the A/D converter subsystem. The state of any pin is software programmable and all port C lines are configured as inputs during power-on or reset. Port pins PC0 and PC1 are capable of sourcing and sinking high currents. For further information, refer to **SECTION 7 INPUT/OUTPUT PORTS** and **SECTION 9 ANALOG-TO-DIGITAL CONVERTER**.

1.4.8 Port D (PD5/CKOUT and PD7/TCAP)

These two I/O pins comprise port D, and one of them is shared with the 16-bit timer subsystem. The state of PD5/CKOUT is software programmable and is configured as an input during power-on or reset (unless clock output has been selected). PD7 is always an input; it may be read at any time, regardless of the mode of operation the 16-bit timer may be in. For further information, refer to **SECTION 7 INPUT/OUTPUT PORTS** and **SECTION 10 16-BIT TIMER**. The PD5/CKOUT pin can be turned into a clock output pin by programming mask option register 1. Clock output is a buffered OSC2 signal with a CMOS output driver.

1.4.9 TCMP

This pin is the output from the 16-bit timer's output compare function. It is low after reset. For further information, refer to **SECTION 10 16-BIT TIMER**.

1.4.10 Maskable Interrupt Request ($\overline{\text{IRQ}}$)

This input pin drives the asynchronous interrupt function of the MCU. The MCU will complete the current instruction being executed before it responds to the $\overline{\text{IRQ}}$ interrupt request. When $\overline{\text{IRQ}}$ is driven low, the event is latched internally to signify an interrupt has been requested. When the MCU completes its current instruction, the interrupt latch is tested. If the interrupt latch is set and the interrupt mask bit (I bit) in the condition code register is clear, the MCU will begin the interrupt sequence.

Depending on the programming option selected in the mask option register 1 (MOR1), the $\overline{\text{IRQ}}$ pin will trigger this interrupt on either a negative-going edge at the $\overline{\text{IRQ}}$ pin and/or while the $\overline{\text{IRQ}}$ pin is held in the low state. In either case, the $\overline{\text{IRQ}}$ pin must be held low for at least one t_{ILIH} time period. If the edge- and level-sensitive edge is programmed in the MOR1, the $\overline{\text{IRQ}}$ input requires an external resistor connected to V_{DD} for wired-OR operation. If the $\overline{\text{IRQ}}$ pin is not used, it must be tied to the V_{DD} supply. The $\overline{\text{IRQ}}$ pin contains an internal Schmitt trigger as part of its input circuitry to improve noise immunity. For further information, refer to **SECTION 4 INTERRUPTS**.

NOTE

If the voltage level applied to the $\overline{\text{IRQ}}$ pin exceeds V_{DD}, it may affect the MCU's mode of operation. See **SECTION 6 OPERATING MODES**.

SECTION 2
MEMORY

2.1 Introduction

The MC68HC805P18 utilizes 14 address lines to access an internal memory space covering 16 Kbytes. This memory space is divided into I/O, RAM, EEPROM, and boot ROM areas.

2.2 User Mode Memory Map

When the MC68HC805P18 is in the user mode, the 32 bytes of I/O, 192 bytes of RAM, 128 bytes of EEPROM, 8000 bytes of program EEPROM, 48 bytes of user page zero EEPROM, and 16 bytes of user vectors EEPROM are all active as shown in Figure 2-1.

2.3 I/O and Control Registers

Figure 2-2 through Figure 2-4 briefly describe the I/O and control registers at locations $0000–$001F. Reading unimplemented bits will return unknown states, and writing unimplemented bits will be ignored.

2.4 RAM

The user RAM consists of 192 bytes (including the stack) at locations $0050 through $010F. The stack begins at address $00FF. The stack pointer can access 64 bytes of RAM from $00FF 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.

2.5 EEPROM

The EEPROM is located at address $0140 and consists of 128 bytes. Programming the EEPROM can be done by the user on a single byte basis by manipulating the programming register, located at address $001C. Refer to **SECTION 8 EEPROM** for a discussion of the EEPROM.

2.6 User EEPROM

There are 8064 bytes of user EEPROM available, consisting of 8000 bytes at locations $1FC0 through $3EFF, 48 bytes in page zero locations $0020 through $004F, and 16 additional bytes for user vectors at locations $3FF0 through $3FFF.

This EEPROM can be programmed only in bootloader mode. Refer to **6.2.2 Bootloader Mode** for more details.

Address		Address			Address
$0000	I/O 32 BYTES	0000		I/O REGISTERS SEE FIGURE 2-2	$0000
$001F		0031			
$0020	USER EEPROM 48 BYTES	0032			
$004F		0079			$001F
$0050	INTERNAL RAM 192 BYTES	0080			
$00BF		0191			
$00C0	STACK 64 BYTES	0192			
$00FF		0255			
$010F		0271			
$013F	UNUSED 48 BYTES	0272			
$0140	EEPROM 128 BYTES	0319	COP CONTROL REGISTER	$3FF0
$01BF		0320	UNIMPLEMENTED	$3FF1
$01C0		0447	UNIMPLEMENTED	$3FF2
	UNUSED 7728 BYTES	0448	UNIMPLEMENTED	$3FF3
			UNIMPLEMENTED	$3FF4
$1FBF		8127	UNIMPLEMENTED	$3FF5
$1FC0		8128	UNIMPLEMENTED	$3FF6
	USER EEPROM 8000 BYTES		UNIMPLEMENTED	$3FF7
			TIMER VECTOR (HIGH BYTE)	$3FF8
$3EFF		16127	TIMER VECTOR (LOW BYTE)	$3FF9
$3F00	MASK OPTION REGISTER	16128	IRQ VECTOR (HIGH BYTE)	$3FFA
$3F01			IRQ VECTOR (LOW BYTE)	$3FFB
$3F02	UNUSED 238 BYTES	16130	SWI VECTOR (HIGH BYTE)	$3FFC
			SWI VECTOR (LOW BYTE)	$3FFD
$3FEF		16367	RESET VECTOR (HIGH BYTE)	$3FFE
$3FF0	USER VECTORS EEPROM 16 BYTES	16368	RESET VECTOR (LOW BYTE)	$3FFF
$3FFF		16383		

Figure 2-1. MC68HC805P18 User Mode Memory Map

PORT A DATA REGISTER	$0000
PORT B DATA REGISTER	$0001
PORT C DATA REGISTER	$0002
PORT D DATA REGISTER	$0003
PORT A DATA DIRECTION REGISTER	$0004
PORT B DATA DIRECTION REGISTER	$0005
PORT C DATA DIRECTION REGISTER	$0006
PORT D DATA DIRECTION REGISTER	$0007
UNIMPLEMENTED	$0008
UNIMPLEMENTED	$0009
SIOP CONTROL REGISTER	$000A
SIOP STATUS REGISTER	$000B
SIOP DATA REGISTER	$000C
UNIMPLEMENTED	$000D
UNIMPLEMENTED	$000E
UNIMPLEMENTED	$000F
UNIMPLEMENTED	$0010
RESERVED	$0011
TIMER CONTROL REGISTER	$0012
TIMER STATUS REGISTER	$0013
INPUT CAPTURE MOST SIGNIFICANT BIT	$0014
INPUT CAPTURE LEAST SIGNIFICANT BIT	$0015
OUTPUT COMPARE MOST SIGNIFICANT BIT	$0016
OUTPUT COMPARE LEAST SIGNIFICANT BIT	$0017
TIMER MOST SIGNIFICANT BIT	$0018
TIMER LEAST SIGNIFICANT BIT	$0019
ALTERNATE COUNTER MOST SIGNIFICANT BIT	$001A
ALTERNATE COUNTER LEAST SIGNIFICANT BIT	$001B
EEPROM PROGRAMMING REGISTER	$001C
A/D CONVERTER DATA REGISTER	$001D
A/D CONVERTER CONTROL AND STATUS REGISTER	$001E
RESERVED	$001F

Figure 2-2. MC68HC805P18 I/O and Control Registers Memory Map

ADDR	REGISTER	READ WRITE	7	6	5	4	3	2	1	0
$0000	PORT A DATA PORTA	R	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
		W								
$0001	PORT B DATA PORTB	R	PB7	PB6	PB5	0	0	0	0	0
		W								
$0002	PORT C DATA PORTC	R	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
		W								
$0003	PORT D DATA PORTD	R	PD7	0	PD5	1	0	0	0	0
		W								
$0004	PORT A DATA DIRECTION DDRA	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		W								
$0005	PORT B DATA DIRECTION DDRB	R	DDRB7	DDRB6	DDRB5	1	1	1	1	1
		W								
$0006	PORT C DATA DIRECTION DDRC	R	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		W								
$0007	PORT D DATA DIRECTION DDRD	R	0	0	DDRD5	0	0	0	0	0
		W								
$0008	UNIMPLEMENTED	R								
		W								
$0009	UNIMPLEMENTED	R								
		W								
$000A	SIOP CONTROL REGISTER SCR	R	0	SPE	0	MSTR	0	0	0	0
		W								
$000B	SIOP STATUS REGISTER SSR	R	SPIF	DCOL	0	0	0	0	0	0
		W								
$000C	SIOP DATA REGISTER SDR	R	SDR7	SDR6	SDR5	SDR4	SDR3	SDR2	SDR1	SDR0
		W								
$000D	UNIMPLEMENTED	R								
		W								
$000E	UNIMPLEMENTED	R								
		W								
$000F	UNIMPLEMENTED	R								
		W								

UNIMPLEMENTED ☐   RESERVED ▓

Figure 2-3. MC68HC805P18 I/O and Control Registers $0000–$000F

ADDR	REGISTER	READ WRITE	7	6	5	4	3	2	1	0
$0010	UNIMPLEMENTED	R								
		W								
$0011	RESERVED	R								
		W								
$0012	TIMER CONTROL REGISTER TCR	R	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
		W								
$0013	TIMER STATUS REGISTER TSR	R	ICF	OCF	TOF	0	0	0	0	0
		W								
$0014	INPUT CAPTURE MSB ICRH	R	ICRH7	ICRH6	ICRH5	ICRH4	ICRH3	ICRH2	ICRH1	ICRH0
		W								
$0015	INPUT CAPTURE LSB ICRL	R	ICRL7	ICRL6	ICRL5	ICRL4	ICRL3	ICRL2	ICRL1	ICRL0
		W								
$0016	OUTPUT COMPARE MSB OCRH	R	OCRH7	OCRH6	OCRH5	OCRH4	OCRH3	OCRH2	OCRH1	OCRH0
		W								
$0017	OUTPUT COMPARE LSB OCRL	R	OCRL7	OCRL6	OCRL5	OCRL4	OCRL3	OCRL2	OCRL1	OCRL0
		W								
$0018	TIMER MSB TMRH	R	TMRH7	TMRH6	TMRH5	TMRH4	TMRH3	TMRH2	TMRH1	TMRH0
		W								
$0019	TIMER LSB TMRL	R	TMRL7	TMRL6	TMRL5	TMRL4	TMRL3	TMRL2	TMRL1	TMRL0
		W								
$001A	ALTERNATE COUNTER MSB ACRH	R	ACRH7	ACRH6	ACRH5	ACRH4	ACRH3	ACRH2	ACRH1	ACRH0
		W								
$001B	ALTERNATE COUNTER LSB ACRL	R	ACRL7	ACRL6	ACRL5	ACRL4	ACRL3	ACRL2	ACRL1	ACRL0
		W								
$001C	EEPROM Programming Register	R	0	CPEN	0	ER1	ER0	LATCH	EERC	EEPGM
		W								
$001D	A/D CONVERSION DATA ADC	R	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
		W								
$001E	A/D STATUS AND CONTROL ADSC	R	CC		ADON	0	0	CH2	CH1	CH0
		W								
$001F	RESERVED	R								
		W								

UNIMPLEMENTED [ ]   RESERVED [ ]

Figure 2-4. MC68HC805P18 I/O and Control Registers $0010-$001F

SECTION 3
CENTRAL PROCESSING UNIT

3.1 Introduction

This section describes the CPU registers.

3.2 CPU Registers

The five CPU registers are shown in Figure 3-1 and the interrupt stacking order are shown in Figure 3-2.
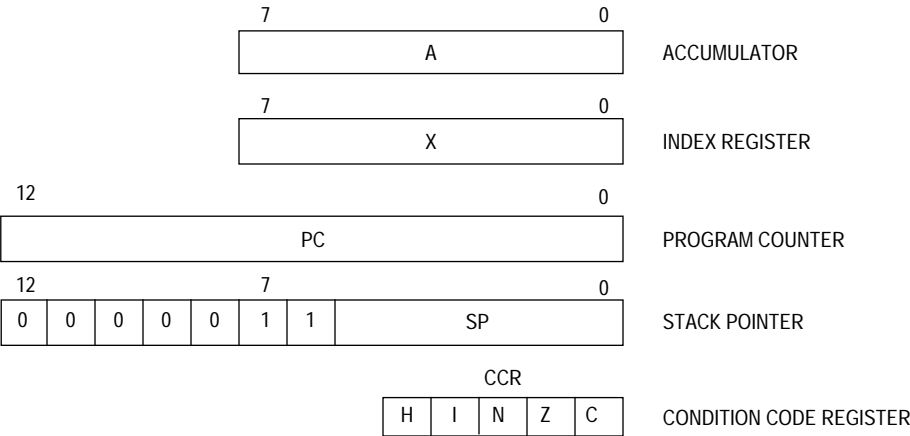
Figure 3-1. Programming Model
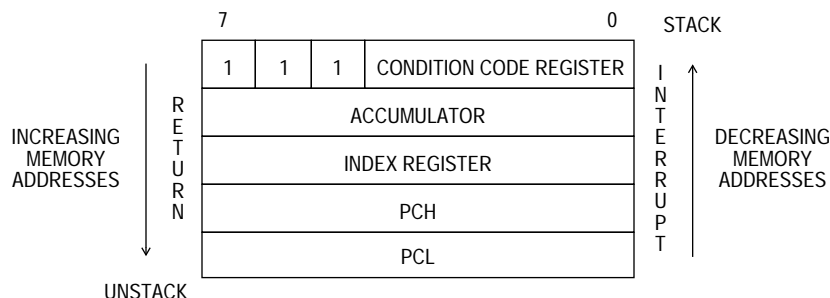
NOTE: Since the stack pointer decrements during pushes, the PCL is stacked first, followed by PCH, etc. Pulling from the stack is in the reverse order.

Figure 3-2. Stacking Order

3.2.1 Accumulator (A)

The accumulator is a general-purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.
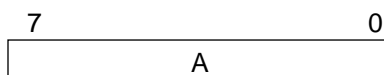
3.2.2 Index Register (X)

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.
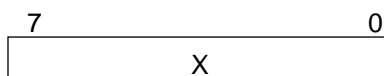
3.2.3 Condition Code Register (CCR)

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.

CCR

H	I	N	Z	C

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 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.

3.2.4 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 $00FF. 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 seven most significant bits are permanently set to 0000011. These seven 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.

```
12                 7                 0
+---+---+---+---+---+---+---+-----------+
| 0 | 0 | 0 | 0 | 0 | 1 | 1 |    SP     |
+---+---+---+---+---+---+---+-----------+
```
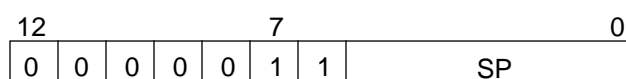
3.2.5 Program Counter (PC)

The program counter is a 13-bit register that contains the address of the next byte to be fetched.

```
12                                     0
+--------------------------------------+
|                 PC                   |
+--------------------------------------+
```
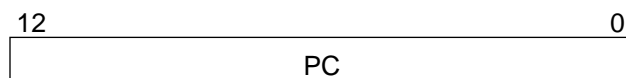
NOTE

The M68HC05 CPU core is capable of addressing a 64-Kbyte memory map. For this implementation, however, the addressing registers are limited to an 8-Kbyte memory map.

SECTION 4
INTERRUPTS

4.1 Introduction

The MCU can be interrupted six different ways:

- Non-maskable software interrupt instruction (SWI)

- External asynchronous interrupt ($\overline{\text{IRQ}}$)

- Input capture interrupt (TIMER)

- Output compare interrupt (TIMER)

- Timer overflow interrupt (TIMER)

- Port A interrupt (if selected via MOR2, bits 0 through 7).

Interrupts cause the processor to save the register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. Unlike reset, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is completed.

When the current instruction is completed, the processor checks all pending hardware interrupts. If interrupts are not masked (I bit in the condition code register is clear) and the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing. Otherwise, the next instruction is fetched and executed. The SWI is executed the same as any other instruction, regardless of the I bit state.

When an interrupt is to be processed, the CPU puts the register contents on the stack, sets the I bit in the CCR, and fetches the address of the corresponding interrupt service routine from the vector table at locations $3FF0 through $3FFF. If more than one interrupt is pending when the interrupt vector is fetched, the interrupt with the highest vector location shown in Table 4-1 will be serviced first.

An RTI instruction is used to signify when the interrupt software service routine is completed. The RTI instruction causes the CPU state to be recovered from the stack and normal processing to resume at the next instruction that was to be executed when the interrupt took place. Figure 4-1 shows the sequence of events that occurs during interrupt processing.

Table 4-1. Vector Addresses for Interrupts and Reset

Register	Flag Name	Interrupts	CPU Interrupt	Vector Address
N/A	N/A	Reset	RESET	$3FF3–$3FFF
N/A	N/A	Software	SWI	$3FFC–$3FFD
N/A	N/A	External Interrupt	IRQ	$3FFA–$3FFB
TSR	ICF	Timer Input Capture	TIMER	$3FF8–$3FF9
TSR	OCF	Timer Output Compare	TIMER	$3FF8–$3FF9
TSR	TOF	Timer Overflow	TIMER	$3FF8–$3FF9
N/A	N/A	Unimplemented	N/A	$3FF6–$3FF7
N/A	N/A	Unimplemented	N/A	$3FF4–$3FF5
N/A	N/A	Unimplemented	N/A	$3FF2–$3FF3
N/A	N/A	Unimplemented	N/A	$3FF0–$3FF1

4.2 Interrupt Types

The interrupts fall into three categories: reset, software, and hardware.

4.2.1 Reset Interrupt Sequence

The reset function is not in the strictest sense an interrupt; however, it is acted upon in a similar manner as shown in Figure 4-1. A low level input on the $\overline{\text{RESET}}$ pin or internally generated RST signal causes the program to vector to its starting address which is specified by the contents of memory locations $3FFE and $3FFF. The I bit in the condition code register is also set. The MCU is configured to a known state during this type of reset as described in **SECTION 5 RESETS**.

4.2.2 Software Interrupt (SWI)

The SWI is an executable instruction. It is also a non-maskable interrupt since it is executed regardless of the state of the I bit in the CCR. As with any instruction, interrupts pending during the previous instruction will be serviced before the SWI opcode is fetched. The interrupt service routine address for the SWI instruction is specified by the contents of memory locations $3FFC and $3FFD.

4.2.3 Hardware Interrupts

All hardware interrupts are maskable by the I bit in the CCR. If the I bit is set, all hardware interrupts (internal and external) are disabled. Clearing the I bit enables the hardware interrupts. Four hardware interrupts are explained in the following paragraphs.
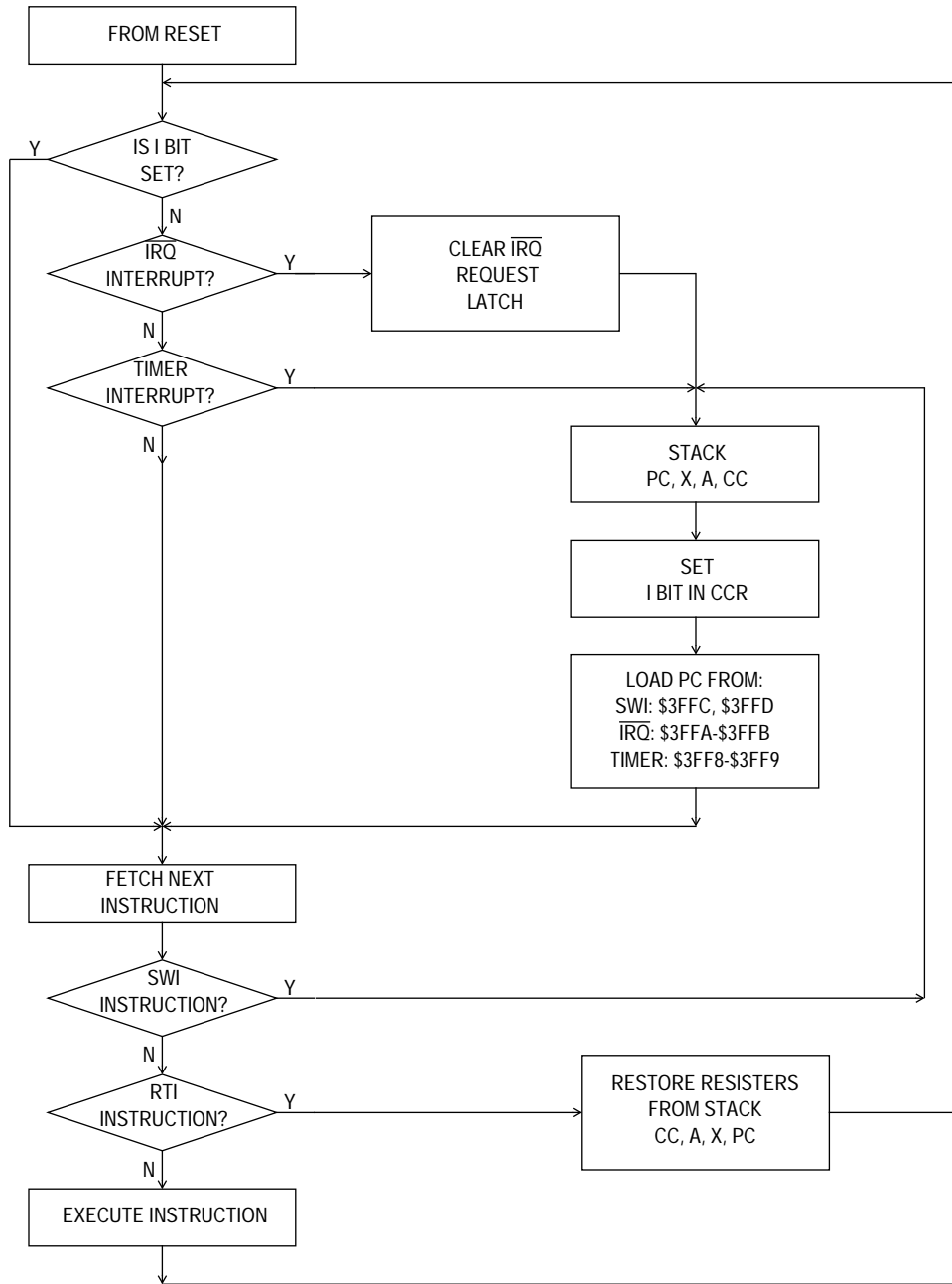
Figure 4-1. Interrupt Processing Flowchart

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

The $\overline{\text{IRQ}}$ pin drives an asynchronous interrupt to the CPU. An edge detector flip-flop is latched on the falling edge of $\overline{\text{IRQ}}$. If either the output from the internal edge detector flip-flop or the level on the $\overline{\text{IRQ}}$ pin is low, a request is synchronized to the CPU to generate the IRQ interrupt. If the edge-sensitive only option is selected, the output of the internal edge detector flip-flop is sampled and the input level on the $\overline{\text{IRQ}}$ pin is ignored. If port A interrupts are

programmed as an option, a port A interrupt will use the same vector. The interrupt service routine address is specified by the contents of memory locations $3FFA and $3FFB.

NOTE

The internal interrupt latch is cleared 9 PH2 clock cycles after the interrupt is recognized (after location $3FFA is read). Therefore, another external interrupt pulse could be latched during the IRQ service routine.

NOTE

When the edge- and level-sensitive option is selected, the voltage applied to the $\overline{IRQ}$ pin must return to the high state before the RTI instruction in the interrupt service routine is executed.

The $\overline{IRQ}$ pin is one source of an IRQ interrupt and a mask option can also enable the port A pins (PA0 through PA7) to act as other IRQ interrupt sources. These sources are all combined into a single ORing function to be latched by the IRQ latch.
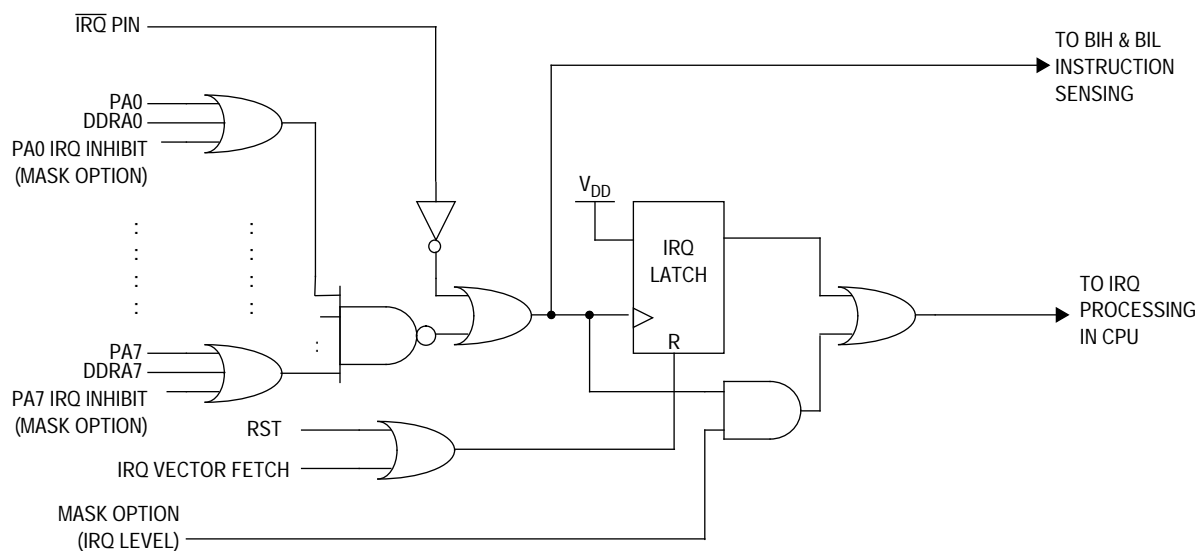
Figure 4-2. IRQ Function Block Diagram

INTERRUPTS

Any enabled IRQ interrupt source sets the IRQ latch on the falling edge of the $\overline{\text{IRQ}}$ pin or a port A pin if port A interrupts have been enabled. If edge-only sensitivity is chosen by a mask option, only the IRQ latch output can activate a request to the CPU to generate the IRQ interrupt sequence. This makes the IRQ interrupt sensitive to the following cases:

1. Falling edge on the $\overline{\text{IRQ}}$ pin with all enabled port A interrupt pins at a high level.

2. Falling edge on any enabled port A interrupt pin with all other enabled port A interrupt pins and the $\overline{\text{IRQ}}$ pin at a high level.

If level sensitivity is chosen, the active high state of the IRQ input can also activate an IRQ request to the CPU to generate the IRQ interrupt sequence. This makes the IRQ interrupt sensitive to the following cases:

1. Low level on the $\overline{\text{IRQ}}$ pin

2. Falling edge on the $\overline{\text{IRQ}}$ pin with all enabled port A interrupt pins at a high level

3. Low level on any enabled port A interrupt pin

4. Falling edge on any enabled port A interrupt pin with all enabled port A interrupt pins and the $\overline{\text{IRQ}}$ pin at a high level

This interrupt is serviced by the interrupt service routine located at the address specified by the contents of $3FFA and $3FFB. The IRQ latch is automatically cleared by entering the interrupt service routine.

Optional External Interrupts (PA0–PA7)

The IRQ interrupt can be triggered by the inputs on the PA0 through PA7 port pins if enabled by individual mask options. With pullup enabled, each port A pin can activate the IRQ interrupt function and the interrupt operation will be the same as for inputs to the $\overline{\text{IRQ}}$ pin. Once enabled by mask option, each individual port A pin can be disabled as an interrupt source if its corresponding DDR bit is configured for output mode.

NOTE

The BIH and BIL instructions apply to the output of the logic OR function of the enabled PA0 through PA7 interrupt pins and the $\overline{\text{IRQ}}$ pin. The BIH and BIL instructions do not test only the state of the $\overline{\text{IRQ}}$ pin.

NOTE

If enabled, the PA0 through PA7 pins will cause an IRQ interrupt only
if these individual pins are configured as inputs.

Input Capture Interrupt

The input capture interrupt is generated by the 16-bit timer as described in
SECTION 10 16-BIT TIMER. The input capture interrupt flag is located in
register TSR and its corresponding enable bit can be found in register TCR. The
I bit in the CCR must be clear in order for the input capture interrupt to be
enabled. The interrupt service routine address is specified by the contents of
memory locations $3FF8 and $3FF9.

Output Compare Interrupt

The output compare interrupt is generated by the 16-bit timer as described in
SECTION 10 16-BIT TIMER. The output compare interrupt flag is located in
register TSR and its corresponding enable bit can be found in register TCR. The
I bit in the CCR must be clear in order for the output compare interrupt to be
enabled. The interrupt service routine address is specified by the contents of
memory locations $3FF8 and $3FF9.

Timer Overflow Interrupt

The timer overflow interrupt is generated by the 16-bit timer as described in
SECTION 10 16-BIT TIMER. The timer overflow interrupt flag is located in
register TSR and its corresponding enable bit can be found in register TCR. The
I bit in the CCR must be clear in order for the timer overflow interrupt to be
enabled. This internal interrupt will vector to the interrupt service routine located
at the address specified by the contents of memory locations $3FF8 and $3FF9.

SECTION 5
RESETS

5.1 Introduction

The MCU can be reset from four sources: one external input and three internal reset conditions. The $\overline{RESET}$ pin is an input with a Schmitt trigger as shown in Figure 5-1. The CPU and all peripheral modules will be reset by the RST signal which is the logical OR of internal reset functions and is clocked by PH2.

5.2 External Reset ($\overline{RESET}$)

The $\overline{RESET}$ input is the only external reset and is connected to an internal Schmitt trigger. The external reset occurs whenever the $\overline{RESET}$ input is driven below the lower threshold and remains in reset until the $\overline{RESET}$ pin rises above the upper threshold. The upper and lower thresholds are given in **SECTION 13 ELECTRICAL SPECIFICATIONS**.
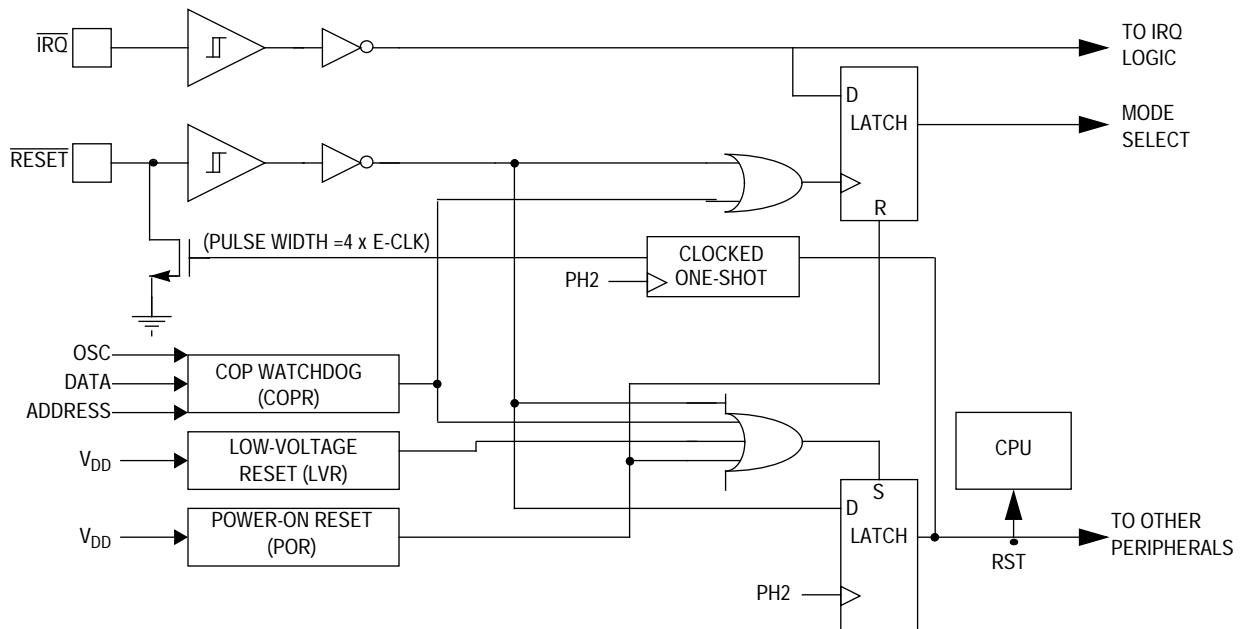
Figure 5-1. Reset Block Diagram

5.3 Internal Resets

The three internally generated resets are the initial power-on reset (POR), the COP watchdog timer, and low-voltage reset (LVR) functions.

5.3.1 Power-On Reset (POR)

The internal POR is generated at power-up to allow the clock oscillator to stabilize. The POR 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 PH2 clock cycle oscillator stabilization delay after the oscillator becomes active.

The POR will generate the RST signal and reset the MCU. The POR will also pull the $\overline{\text{RESET}}$ pin low at the same time, allowing external devices to be reset with the MCU. If any other reset function is active at the end of this 4064 PH2 clock cycle delay, the RST signal will remain active until the other reset condition(s) end.

5.3.2 Computer Operating Properly (COP) Reset

When the COP watchdog timer is enabled (by MOR1, bit 0), the internal COP reset is generated automatically by a timeout of the COP watchdog timer. This timer is implemented with an 18-stage ripple counter that provides a time-out period of 65.5 ms when a 4-MHz oscillator is used. The COP watchdog counter is cleared by writing a logical zero to bit zero at location $3FF0.

The COP register is shared with the most significant bit (MSB) of an unimplemented user interrupt vector as shown in Figure 5-2. Reading this location will return the MSB of the unimplemented user interrupt vector. Writing to this location will clear the COP watchdog timer.

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	0	0
Write:	R							COPR
Reset:	—	—	—	—	—	—	—	—

$3FF0 (to left of Read/Write rows)

  = Unimplemented   R = Reserved

**Figure 5-2. Unimplemented Vector and
COP Watchdog Timer Register**

5.3.3 Low-Voltage Reset (LVR)

If the LVR has been enabled via MOR1, the internal LVR reset is generated when the supply voltage to the V_{DD} pin falls below a nominal 3.80 Vdc. The LVR threshold is not intended to be an accurate and stable trip point, but is intended to ensure that the CPU will be held in reset when the V_{DD} supply voltage is below reasonable operating limits. If the LVR is tripped for a short time, the LVR reset signal will last at least two cycles of the CPU bus clock, PH2.

The LVR will generate the RST signal which will reset the CPU and other peripherals. Also, the LVR will establish the mode of operation based on the state of the $\overline{IRQ}$ pin at the time the LVR signal ends. If any other reset function is active at the end of the LVR reset signal, the RST signal will remain in the reset condition until the other reset condition(s) end.

NOTE

The voltage of the $\overline{IRQ}$ pin must be between 0–V_{DD} volts to stay in the normal operation mode.

SECTION 6
OPERATING MODES

6.1 Introduction

This section describes the user, bootloader, and low-power modes. In addition the computer operating properly (COP) timer considerations are discussed.

6.2 User Modes

The MC68HC805P18 has two modes of operation available to the user:

- User mode
- Bootloader mode

The mode of operation is determined by the voltages on the $\overline{\text{IRQ}}$ and PD7/TCAP pins on the rising edge of the external $\overline{\text{RESET}}$ pin. Table 6-1 shows the condition required to go into each mode.

Table 6-1. Operating Mode Conditions

$\overline{\text{RESET}}$	$\overline{\text{IRQ}}$	TCAP	Mode
⌐⌐⌐	0–5 V	0–5 V	User
	2 x V_{DD}	5 V	Bootloader

6.2.1 User Mode

The user mode allows the MCU to function as a self-contained microcontroller with maximum use of the pins for on-chip peripheral functions. All address and data activity occurs within the MCU and is not available externally. User mode is entered on the rising edge of $\overline{\text{RESET}}$, if the $\overline{\text{IRQ}}$ pin is within the normal operating voltage range. In the user mode, there is an 8-bit I/O port, a second 8-bit I/O port shared with the analog-to-digital (A/D) subsystem, one 3-bit I/O port shared with the serial input/output port (SIOP), and a 2-bit I/O port shared with the16-bit timer subsystem.

6.2.2 Bootloader Mode

Bootloader mode is entered upon the rising edge of $\overline{\text{RESET}}$ if the $\overline{\text{IRQ}}$ pin is twice the V_{DD} voltage and the TCAP/PD7 pin is at logic one. In bootloader mode, the user EEPROM and mask option register (MOR) bytes can be erased and programmed. Figure 6-1 shows the bootloader circuit. PTC4 determines whether erasing or programming will occur as shown in Table 6-2.

Table 6-2. Bootloader Functions

PTC4	Function
0	Bulk Erase/Blank Verify
1	Bulk Erase/Program/Verify

Bulk Erase/Blank Verify

To use the bootloader circuit to bulk erase the user EEPROM, follow this sequence:

1. Close $\overline{\text{RESET}}$ switch and PTC4 switch so these pins are held low.

2. Apply 12 V power to $\overline{\text{IRQ}}$.

3. Release $\overline{\text{RESET}}$.

4. Programming LED will turn on while bulk erase is occurring.

5. When bulk erase is finished, programming LED will turn off.

6. When blank verify is finished, verify LED will turn on.

7. Close $\overline{\text{RESET}}$ switch.

8. Remove 12 V from $\overline{\text{IRQ}}$, then remove power.
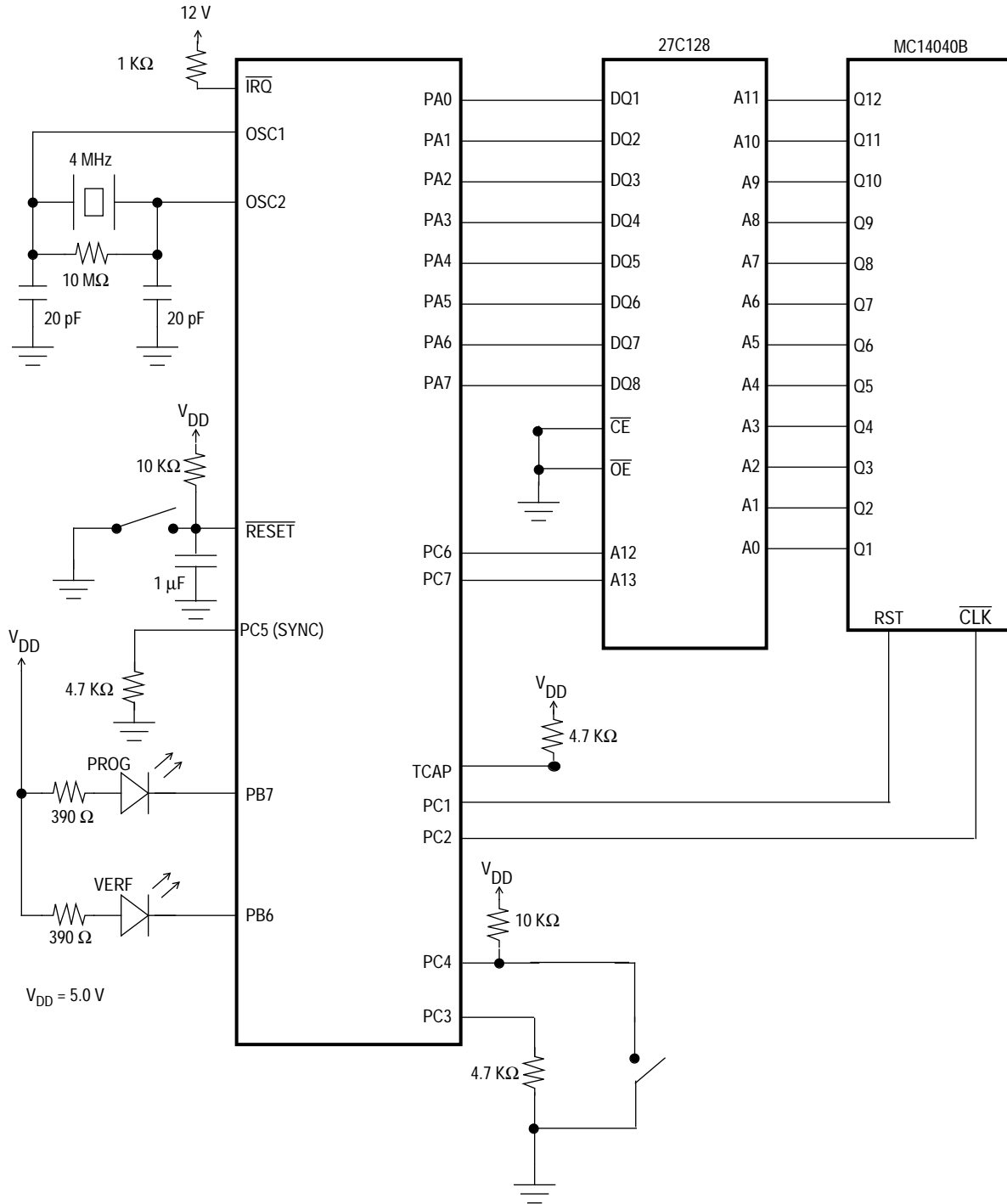
Figure 6-1. Bootloader Circuit

Bulk Erase/Program Verify

To use the bootloader circuit to bulk erase, program, and verify the user EEPROM, follow this sequence:

1. Close $\overline{\text{RESET}}$ switch so $\overline{\text{RESET}}$ is low upon power up.

2. Open PTC4 switch so PTC4 remains high during reset sequence.

3. Make sure code to be loaded into user EEPROM is in the external EPROM (shown as 27C128).

4. Apply 12 V power to $\overline{\text{IRQ}}$.

5. Release $\overline{\text{RESET}}$.

6. Programming LED will be on during bulk erase and programming. (The code in the 27C128 will be loaded into the user EEPROM and MOR.)

7. When programming is finished, the programming LED will turn off.

8. When the verify is finished, verify LED will turn on.

9. Close $\overline{\text{RESET}}$ switch.

10. Remove 12 V from $\overline{\text{IRQ}}$, then remove power.

NOTE

Bootloader mode is the only mode in which the user can program the 8K user EEPROM and MOR. The 128-byte EEPROM can be programmed in user mode.

6.3 Low-Power Modes

The MC68HC805P18 is capable of running in a low-power mode in each of its configurations. The WAIT and STOP instructions provide modes that reduce the power required for the MCU by stopping various internal clocks and/or the on-chip oscillator. The STOP and WAIT instructions are not normally used if the COP watchdog timer is enabled (MOR1, bit 0). The stop conversion to halt option (MOR1, bit 5) is used to modify the behavior of the STOP instruction from stop mode to halt mode. The flow of the stop, halt, and wait modes is shown in Figure 6-2.

6.4 STOP Instruction

The STOP instruction can result in one of two modes of operation depending on the option programmed in the mask option register 1. If the stop conversion to halt option (MOR1,bit 5) is not chosen, the STOP instruction will behave like a normal STOP instruction in the M68HC805 Family and place the MCU in the stop mode. If the stop conversion to halt option is chosen, the STOP instruction will behave like a WAIT instruction (with the exception of a brief delay at startup) and place the MCU in the halt mode.

6.4.1 Stop Mode

Execution of the STOP instruction (without conversion to halt) places the MCU in its lowest-power consumption mode. In the stop mode, the internal oscillator is turned off stopping *all* internal processing including the COP watchdog timer. The RC oscillator that feeds the EEPROM and the A/D converter is also stopped. Execution of the STOP instruction automatically clears the I bit in the condition code register so that the $\overline{IRQ}$ external interrupt is enabled. All other registers and memory remain unaltered. All input/output lines remain unchanged.

The MCU can be brought out of the stop mode only by an IRQ external interrupt (or port A, if selected as an option in the MOR2) or an externally generated reset. When exiting the stop mode, the internal oscillator will resume after a 4064 PH2 clock cycle oscillator stabilization delay.
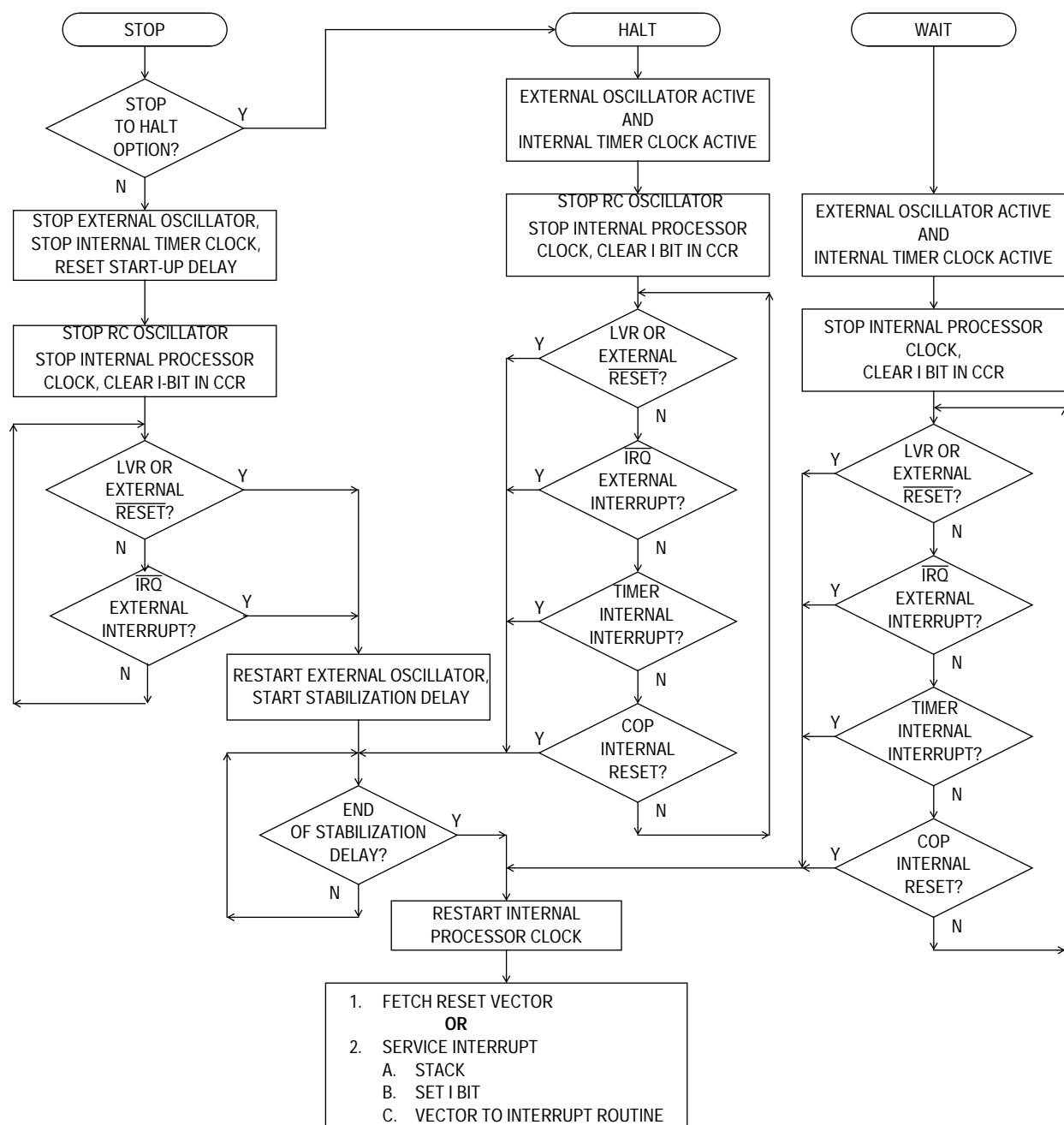
Figure 6-2. STOP/WAIT Flowcharts

NOTE

Execution of the STOP instruction without conversion to halt (via MOR1) will cause the oscillator to stop, and therefore disable the COP watchdog timer. If the COP watchdog timer is to be used, the stop mode should be changed to the halt mode by programming the appropriate option in MOR1.

6.4.2 Halt Mode

Execution of the STOP instruction with the conversion to halt places the MCU in this low-power mode. Halt mode consumes the same amount of power as wait mode (both halt and wait modes consume more power than stop mode).

In halt mode the PH2 clock is halted, suspending all processor and internal bus activity. Internal timer clocks remain active, permitting interrupts to be generated from the 16-bit timer or a reset to be generated from the COP watchdog timer. Execution of the STOP instruction automatically clears the I bit in the condition code register enabling the $\overline{\text{IRQ}}$ external interrupt. All other registers, memory, and input/output lines remain in their previous states.

If the 16-bit timer interrupt is enabled, it will cause the processor to exit the halt mode and resume normal operation. The halt mode also can be exited when an $\overline{\text{IRQ}}$ external interrupt (or port A, if selected as an option in the MOR2) or external $\overline{\text{RESET}}$ occurs. When exiting the halt mode, the PH2 clock will resume after a delay of one to 4064 PH2 clock cycles. This varied delay time is the result of the halt mode exit circuitry testing the oscillator stabilization delay timer (a feature of the stop mode), which has been free-running (a feature of the wait mode).

NOTE

The halt mode is not intended for normal use. This feature is provided to keep the COP watchdog timer active in the event a STOP instruction is executed inadvertently.

6.4.3 WAIT Instruction

The WAIT instruction places the MCU in a low-power mode which consumes more power than the stop mode. In wait mode the PH2 clock is halted, suspending all processor and internal bus activity. Internal timer clocks remain active, permitting interrupts to be generated from the 16-bit timer and reset to be generated from the COP watchdog timer. Execution of the WAIT instruction automatically clears the I bit in the condition code register enabling the $\overline{IRQ}$ external interrupt. All other registers, memory, and input/output lines remain in their previous state.

If the 16-bit timer interrupt is enabled it will cause the processor to exit the wait mode and resume normal operation. The 16-bit timer may be used to generate a periodic exit from the wait mode. The wait mode may also be exited when an $\overline{IRQ}$ or $\overline{RESET}$ occurs. Note that if port A interrupts (if programmed as an option in the mask option register 1) will also exit wait mode. However, when exiting the wait mode, the internal oscillator will not need to wait for 4064 PH2 clock cycles to stabilize as in the stop and halt modes.

6.5 COP Watchdog Timer Considerations

The COP watchdog timer is active in user mode of operation when programmed as an option in MOR1. Executing the STOP instruction without conversion to halt (via mask option register1) will cause the COP to be disabled. Therefore, it is recommended that the STOP instruction be modified to produce halt mode (via MOR1) if the COP watchdog timer will be enabled.

Furthermore, it is recommended that the COP watchdog timer be disabled for applications that will use the halt or wait modes for time periods that will exceed the COP time-out period.

COP watchdog timer interactions are summarized in Table 6-3.

Table 6-3. COP Watchdog Timer Recommendations

IF the following conditions exist:		THEN the COP Watchdog Timer should be:
STOP Instruction Mode	Wait Period	
Halt Mode Selected via MOR1, Bit 5	WAIT Period *Less than* COP Time Out	Enable or Disable COP via MOR1, Bit 0
Halt Mode Selected via MOR1, Bit 5	WAIT Period *More Than* COP Time Out	Disable COP via MOR1, Bit 0
Stop Mode Selected via MOR1, Bit 5	Any Length Wait Period	Disable COP via MOR1, Bit 0

SECTION 7
INPUT/OUTPUT PORTS

7.1 Introduction

In user mode, 20 bidirectional input/output (I/O) lines are arranged as two 8-bit I/O ports (ports A and C), one 3-bit I/O port (port B), and one 1-bit I/O port (port D). These ports are programmable as either inputs or outputs under software control of the data direction registers (DDRs). An input-only pin is associated with port D.

7.2 Port A

Port A is an 8-bit bidirectional port which can share its pins with the IRQ interrupt system as shown in Figure 7-1. Each port A pin is controlled by the corresponding bits in a data direction register and a data register. The port A data register is located at address $0000. The port A data direction register (DDRA) is located at address $0004. Reset clears the DDRA, thereby initializing port A as an input port. The port A data register is unaffected by reset.
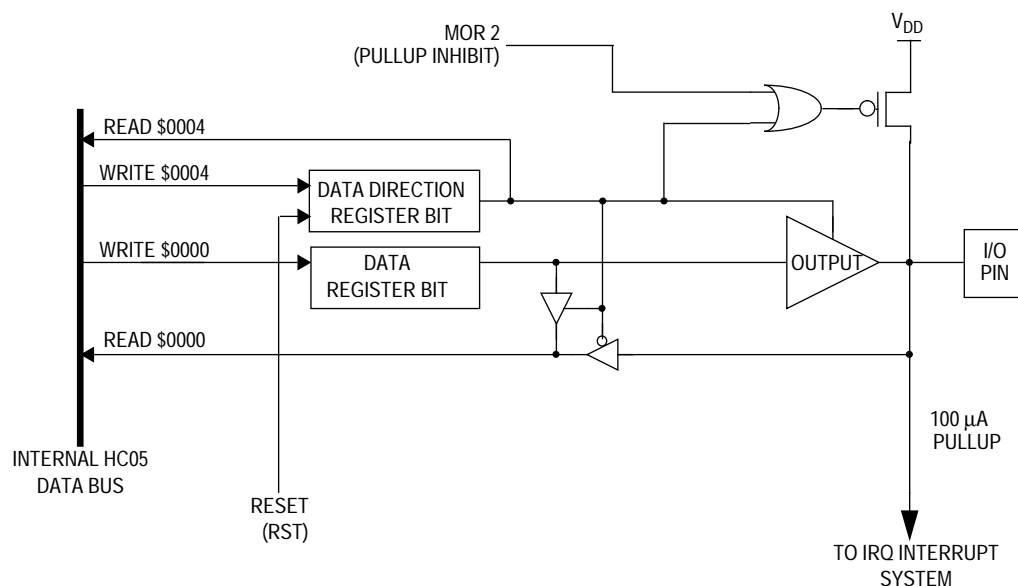

Figure 7-1. Port A I/O Circuitry

7.3 Port B

Port B is a 3-bit bidirectional port which can share pins PB5–PB7 with the SIOP communications subsystem. The port B data register is located at address $0001 and its data direction register (DDR) is located at address $0005. Reset does not affect the data registers, but clears the DDRs, thereby setting all of the port pins to input mode. Writing a logic one to a DDR bit sets the corresponding port pin to output mode (see Figure 7-2).

Port B may be used for general I/O applications when the SIOP subsystem is disabled. The SPE bit in register SPCR is used to enable/disable the SIOP subsystem. When the SIOP subsystem is enabled, port B registers are still accessible to software. Writing to either of the port B registers while a data transfer is under way could corrupt the data. See **SECTION 11 SERIAL INPUT/OUTPUT PORT** for a discussion of the SIOP subsystem.
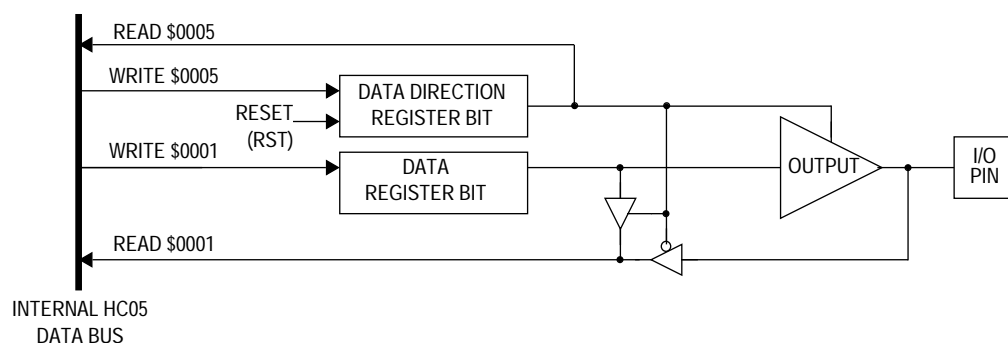
Figure 7-2. Port B I/O Circuitry

7.4 Port C

Port C is an 8-bit bidirectional port which can share pins PC3–PC7 with the A/D subsystem. The port C data register is located at address $0002 and its data direction register (DDR) is located at address $0006. Reset does not affect the data registers, but clears the DDRs, thereby setting all of the port pins to input mode. Writing a logic one to a DDR bit sets the corresponding port pin to output mode (see Figure 7-3). Two port C pins, PC0 and PC1, can source and sink a higher current than a typical I/O pin. See **SECTION 13 ELECTRICAL SPECIFICATIONS** regarding current specifications.

Port C may be used for general I/O applications when the A/D subsystem is disabled. The ADON bit in register ADSC is used to enable/disable the A/D subsystem. Care must be exercised when using pins PC0–PC2 while the A/D subsystem is enabled. Accidental changes to bits that affect pins PC3–PC7 in the data or DDR registers will produce unpredictable results in the A/D subsystem. See **SECTION 9 ANALOG-TO-DIGITAL CONVERTER**.
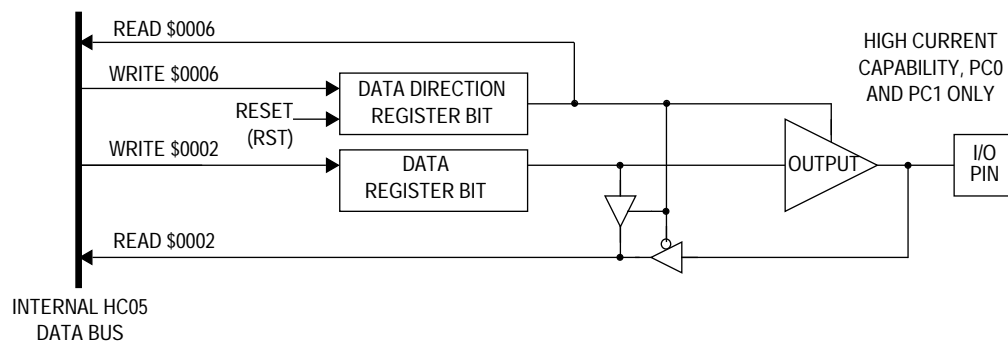

Figure 7-3. Port C I/O Circuitry

7.5 Port D

Port D is a 2-bit port with one bidirectional pin (PD5/CKOUT) and one input-only pin (PD7). Pin PD7 is shared with the 16-bit timer. PD5 can be replaced with a buffered OSC2 clock output via MOR1. The port D data register is located at address $0003 and its data direction register (DDR) is located at address $0007. Reset does not affect the data registers, but clears the DDRs, thereby setting PD5/CKOUT to input mode. Writing a one to DDR bit 5 sets PD5/CKOUT to output mode (see Figure 7-4).

Port D may be used for general I/O applications regardless of the state of the 16-bit timer. Since PD7 is an input-only line, its state can be read from the port D data register at any time.
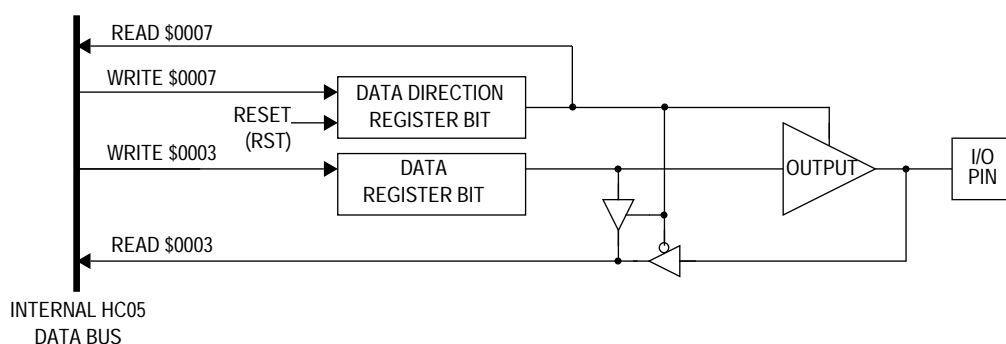

Figure 7-4. Port D I/O Circuitry

INPUT/OUTPUT PORTS

7.6 I/O Port Programming

Each pin on ports A through port D (except pin 7 of port D) may be programmed as an input or an output under software control as shown in Table 7-1, Table 7-2, Table 7-3, and Table 7-4. The direction of a pin is determined by the state of its corresponding bit in the associated port data direction register (DDR). A pin is configured as an output if its corresponding DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared to a logic zero.

Table 7-1. Port A I/O Functions

DDRA	I/O Pin Mode	Access to DDRA @ $0004	Access to Data Register @ $0000	
		Read/Write	Read	Write
0	Input, High Impedance	DDRA0–DDRA7	I/O Pin	*
1	Output	DDRA0–DDRA7	PA0–PA7	PA0–PA7

*Does not affect input, but stored to data register

Table 7-2. Port B I/O Functions

DDRB	I/O Pin Mode	Access to DDRA @ $0005	Access to Data Register @ $0001	
		Read/Write	Read	Write
0	Input, High Impedance	DDRB5–DDRB7	I/O Pin	*
1	Output	DDRB5–DDRB7	PB5–PB7	PB5–PB7

*Does not affect input, but stored to data register

Table 7-3. Port C I/O Functions

DDRA	I/O Pin Mode	Access to DDRA @ $0006	Accesses to Data Register @ $0002	
		Read/Write	Read	Write
0	Input, High Impedance	DDRC0–DDRC7	I/O Pin	*
1	Output	DDRC0–DDRC7	PC0–PC7	PC0–PC7

*Does not affect input, but stored to data register

Table 7-4. Port D I/O Functions

DDRA	I/O Pin Mode	Access to DDRA @ $0007	Accesses to Data Register @ $0003	
		Read/Write	Read	Write
0	Input, High Impedance	DDRD5	I/O Pin	*
1	Output	DDRD5	PD5/CKOUT	PD5/CKOUT

*Does not affect input, but stored to data register
**PD7 is input-only

NOTE

To avoid generating a glitch on an I/O port pin, data should be written
to the I/O port data register before writing a logical one to the
corresponding data direction register.

SECTION 8
EEPROM

8.1 Introduction

This section describes the EEPROM which is located at address $0140 and consists of 128 bytes. Programming the EEPROM can be done by the user on a single byte basis by manipulating the programming register located at address $001C.

Also, the mask option register (MOR), which consists of two additional EEPROM bytes, is discussed.

8.2 EEPROM Programming Register (EEPROG)

The contents and use of the programming register are discussed here.

		Bit 7	6	5	4	3	2	1	Bit 0
EEPROG $001C	Read:	0	CPEN	0	ER1	ER0	LATCH	EERC	EEPGM
	Write:								
	Reset:	0	0	0	0	0	0	0	0

☐ = Unimplemented

Figure 8-1. EEPROM Programming Register

CPEN — Charge Pump Enable

When set, CPEN enables the charge pump which produces the internal EEPROM programming voltage. This bit should be set concurrently with the LATCH bit. The programming voltage will not be available until EEPGM is set. The charge pump should be disabled when not in use. CPEN is readable and writable and is cleared by reset.

ER1 and ER0 — Erase Select Bits

ER1 and ER0 form a 2-bit field which is used to select one of three erase modes: byte, block, or bulk. Table 8-1 shows the modes selected for each bit configuration. These bits are readable and writable and are cleared by reset.

In byte erase mode, only the selected byte is erased. In block mode, a 32-byte block of EEPROM is erased. The EEPROM memory space is divided into four 32-byte blocks ($140–$15F, $160–$17F, $180–$19F, $1A0–$1BF), and doing a block erase to any address within a block will erase the entire block. In bulk erase mode, the entire 128-byte EEPROM section is erased.

Table 8-1. Erase Mode Select

ER1	ER0	Mode
0	0	Program (no Erase)
0	1	Byte Erase
1	0	Block Erase
1	1	Bulk Erase

LATCH — Latch Bit

When set, LATCH configures the EEPROM address and data bus for programming. Writes to the EEPROM array cause the data bus and the address bus to be latched. This bit is readable and writable, but reads from the array are inhibited if the LATCH bit is set and a write to the EEPROM space has taken place.

When clear, address and data buses are configured for normal operation. Reset clears this bit.

EERC — EEPROM RC Oscillator Control

When this bit is set, the EEPROM section uses the internal RC oscillator instead of the CPU clock. The RC oscillator is shared with the A/D converter, so this bit should be set by the user when the internal bus frequency is below 1.5 MHz to guarantee reliable operation of the EEPROM or A/D converter. After setting the EERC bit, delay a time, t_{RCON}, to allow the RC oscillator to stabilize. This bit is readable and writable. The EERC bit is cleared by reset. The RC oscillator is disabled while the MCU is in stop mode.

EEPGM — EEPROM Programming Power Enable

EEPGM must be written to enable (or disable) the EEPGM function. When set, EEPGM turns on the charge pump and enables the programming (or erasing) power to the EEPROM array. When clear, this power is switched off. This will enable pulsing of the programming voltage to be controlled internally. This bit can be read at any time, but can only be written to if LATCH = 1. If LATCH is not set, then EEPGM cannot be set. LATCH and EEPGM cannot both be set with one write if LATCH is cleared. EEPGM is cleared automatically when LATCH is cleared. Reset clears this bit.

8.3 Programming/Erasing Procedures

To program a byte of EEPROM, set LATCH = CPEN = 1, set ER1 = ER0 = 0, write data to the desired address and then set EEPGM for a time, t_{EPGM}.

NOTE

Any bit should be erased before it is programmed. However, if write/erase cycling is a concern, the following procedure will minimize the cycling of each bit in each EEPROM byte.

If PB • $\overline{EB}$ = 0, then program the new data over the existing data without erasing it first. If PB • $\overline{EB}$ ≠ 0, then erase the byte before programming where PB = byte data to be programmed and EB = existing EEPROM byte data.

To erase a **byte** of EEPROM, set LATCH = 1, CPEN = 1, ER1 = 0 and ER0 = 1, write to the address to be erased and set EEPGM for a time, t_{EBYT}.

To erase a **block** of EEPROM, set LATCH = 1, CPEN = 1, ER1 = 1 and ER0 = 0, write to any address in the block, and set EEPGM for a time, t_{EBLOCK}.

For a **bulk** erase, set LATCH = 1, CPEN = 1, ER1 = 1, and ER0 = 1, write to any address in the array, and set EEPGM for a time, t_{EBULK}.

To terminate the programming or erase sequence, clear EEPGM, delay for a time t_{FPV} to allow the program voltage to fall, and then clear LATCH and CPEN to free up the buses. Following each erase or programming sequence, clear all programming control bits.

NOTE

Erased/programmed state of the programming EEPROM (128 bytes) and the user EEPROM (8064 bytes) is opposite. An erased EEPROM memory location is a logic **zero** for user EEPROM, while it is a logic **one** for programming EEPROM.

8.4 Mask Option Registers (MOR)

The MOR consists of two EEPROM bytes located at $3F00 and $3F01. The MOR holds the 16 option bits for:

- The SIOP data format, interrupt sensitivity
- COP enable/disable
- SIOP clock rate
- LVR enable/disable
- Stop conversion to halt, pullup/interrupt enable on port A
- Clock output option to replace PD5

When in the erased state, the EEPROM cells will read as logic **zeros.** These registers are refreshed every 256 μs during power-on reset and every 16 ms after the part is out of reset (assuming $f_{OSC} = 4$ MHz).
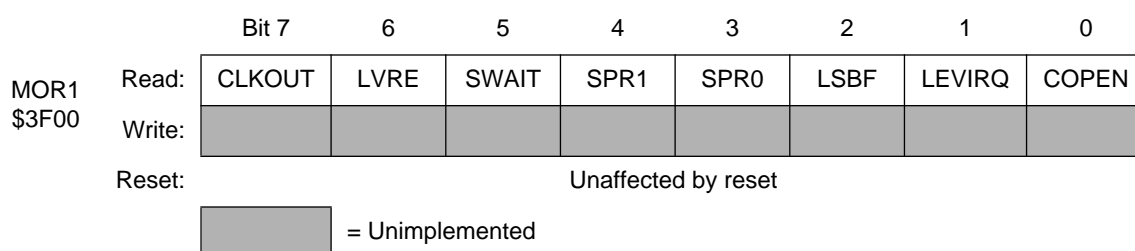
		Bit 7	6	5	4	3	2	1	0
MOR1 $3F00	Read:	CLKOUT	LVRE	SWAIT	SPR1	SPR0	LSBF	LEVIRQ	COPEN
	Write:								
	Reset:				Unaffected by reset				

☐ = Unimplemented

Figure 8-2. Mask Option Register 1

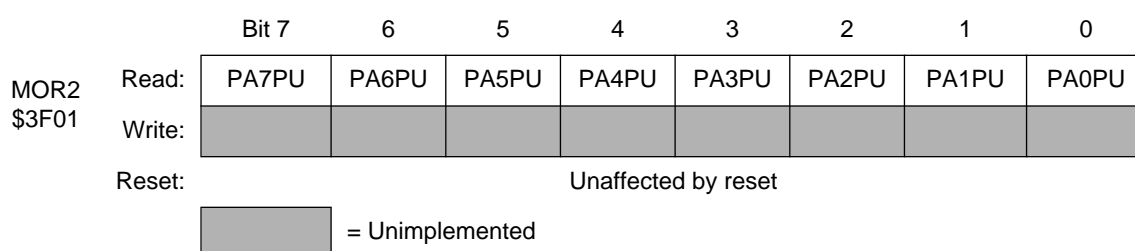|		Bit 7	6	5	4	3	2	1	0
MOR2 $3F01	Read:	PA7PU	PA6PU	PA5PU	PA4PU	PA3PU	PA2PU	PA1PU	PA0PU
	Write:								
	Reset:				Unaffected by reset				

☐ = Unimplemented

Figure 8-3. Mask Option Register 2

COPEN — COP enable/disable

COPEN may be read at any time. In user mode, writing has no effect. It has to be programmed in bootloader mode.
0 = The COP is disabled (erased state).
1 = The COP is enabled.

LEVIRQ — Interrupt request option

LEVIRQ may be read at any time. In user mode, writing has no effect. It has to be programmed in bootloader mode.

0 = The $\overline{\text{IRQ}}$ pin is edge-sensitive (erased state).
1 = The $\overline{\text{IRQ}}$ pin is edge- and level-sensitive.

LSBF — SIOP MSB or LSB first

LSBF may be read at any time. In user mode, writing has no effect. It has to be programmed in bootloader mode.

0 = The SIOP sends/receives MSB (bit 7) first (erased state).
1 = The SIOP sends/receives LSB (bit 0) first.

SPR1 and SPR0 — SIOP Rate Select Bits

These bits may be read at any time. In user mode, writing has no effect. It has to be programmed in bootloader mode.

Table 8-2. SIOP Clock Rate Selection

SPR1	SPR0	Frequency
0	0	f_{OSC} divided by 16
0	1	f_{OSC} divided by 8
1	0	f_{OSC} divided by 4
1	1	f_{OSC} divided by 2

SWAIT — STOP conversion to WAIT

SWAIT may be read at any time. In user mode, writing has no effect. It has to be programmed in bootloader mode.

0 = STOP instruction puts MCU in stop mode.
1 = STOP instruction puts MCU in halt mode.

LVRE — LVR enable/disable

LVRE may be read at any time. In user mode, writing has no effect. It has to be programmed in bootloader mode.

0 = The LVR is disabled (erased state).
1 = The LVR is enabled.

CLKOUT — CLKOUT enable/disable

CLKOUT may be read at any time. In user mode, writing has no effect. It has to be programmed in bootloader mode.

0 = The CLKOUT is disabled (erased state).
1 = The CLKOUT is enabled.

PA7PU through PA0PU — Port A pullups/interrupt enable/disable

These bits may be read at any time. In user mode, writing has no effect. It has to be programmed in bootloader mode.

0 = Port A (bits 0 through 7) pullups/interrupt is disabled (erased state).
1 = Port A (bits 0 through 7) pullups/interrupt is enabled.

SECTION 9
ANALOG-TO-DIGITAL CONVERTER

9.1 Introduction

The MC68HC805P18 includes a 4-channel, multiplexed input, 8-bit successive approximation analog-to-digital (A/D) converter. The A/D subsystem shares its inputs with port C pins PC3 through PC7.

9.2 Analog Section

The following paragraphs describe the operation and performance of analog modules within the analog subsystem.

9.2.1 Ratiometric Conversion

The A/D converter is ratiometric, with pin V_{REFH} supplying the high reference voltage. Applying an input voltage equal to V_{REFH} produces a conversion result of $FF (full scale). Applying an input voltage equal to V_{SS} produces a conversion result of $00. An input voltage greater than V_{REFH} will convert to $FF with no overflow indication. For ratiometric conversions, V_{REFH} should be at the same potential as the supply voltage being used by the analog signal being measured and referenced to V_{SS}.

9.2.2 V_{REFH}

The reference supply for the A/D converter shares pin PC7 with port C. The low reference is tied to the V_{SS} pin internally. V_{REFH} can be any voltage between V_{SS} and V_{DD}; however, the accuracy of conversions is tested and guaranteed only for $V_{REFH} = V_{DD}$.

9.2.3 Accuracy and Precision

The 8-bit conversion result is accurate to within ± 1 1/2 LSB, including quantization; however, the accuracy of conversions is tested and guaranteed only with external oscillator operation.

9.2.4 Conversion Process

The A/D reference inputs are applied to a precision digital-to-analog converter. Control logic drives the D/A and the analog output is successively compared to the selected analog input which was sampled at the beginning of the conversion cycle. The conversion process is monotonic and has no missing codes.

9.3 Digital Section

The following paragraphs describe the operation and performance of digital modules within the analog subsystem.

9.3.1 Conversion Times

Each input conversion requires 32 PH2 clock cycles, which must be at a frequency equal to or greater than 1 MHz.

9.3.2 Internal versus External Oscillator

If the MCU PH2 clock frequency is less than 1 MHz (2 MHz external oscillator), the internal RC oscillator (approximately 1.5 MHz) must be used for the A/D converter clock. The internal RC clock is selected by setting the EERC bit in the EEPROG register.

NOTE

The RC oscillator is shared with the EEPROM module. The RC oscillator is disabled while the MCU is in stop mode.

When the internal RC oscillator is being used, these limitations apply:

1. Since the internal RC oscillator is running asynchronously with respect to the PH2 clock, the conversion complete bit (CC) in the A/D status and control register must be used to determine when a conversion sequence has been completed.

2. Electrical noise will slightly degrade the accuracy of the A/D converter. The A/D converter is synchronized to read voltages during the quiet period of the clock driving it. Since the internal and external clocks are not synchronized the A/D converter occasionally will measure an input when the external clock is making a transition.

3. If the PH2 clock is 1 MHz or greater (for instance, external oscillator 2 MHz or greater), the internal RC oscillator should be turned off and the external oscillator used as the conversion clock.

9.3.3 Multi-Channel Operation

An input multiplexer allows the A/D converter to select from one of four external analog signals. Port C pins PC3 through PC6 are shared with the inputs to the multiplexer.

9.4 A/D Status and Control Register (ADSC)

The ADSC register reports the completion of A/D conversion and provides control over oscillator selection, analog subsystem power, and input channel selection. See Figure 9-1.
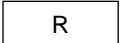

Figure 9-1. A/D Status and Control Register

CC — Conversion Complete

This read-only status bit is set when a conversion sequence has completed and data is ready to be read from the ADC register. CC is cleared when a channel is selected for conversion, when data is read from the ADC register, or when the A/D subsystem is turned off. Once a conversion has been started, conversions of the selected channel will continue every 32 PH2 clock cycles until the ADSC register is written to again. During continuous conversion operation, the ADC register will be updated with new data and the CC bit set every 32 PH2 clock cycles. Also, data from the previous conversion will be overwritten regardless of the state of the CC bit.

Reserved

This bit is not used currently. It can be read or written, but does not control anything.

ADON — A/D Subsystem On

When the A/D subsystem is turned on (ADON = 1), it requires a time, t_{ADON}, to stabilize before accurate conversion results can be attained.

CH2-CH0 — Channel Select Bits

CH2, CH1, and CH0 form a 3-bit field which is used to select an input to the A/D converter. Channels 0 through 3 correspond to port C input pins PC6 through PC3. Channels 4 through 6 are used for reference measurements. In user mode channel 7 is reserved. If a conversion is attempted with channel 7 selected the result will be $00. Table 9-1 lists the inputs selected by bits CH0 through CH3.

If the ADON bit is set and an input from channels 0 through 4 is selected, the corresponding port C pin's DDR bit will be cleared (making that port C pin an input). If the port C data register is read while the A/D is on and one of the shared input channels is selected using bit CH0 through CH2, the corresponding port C pin will read as a logic zero. The remaining port C pins will read normally. To digitally read a port C pin, the A/D subsystem must be disabled (ADON = 0) or input channel 5 through 7 must be selected.

Table 9-1. A/D Multiplexer Input
Channel Assignments

Channel	Signal
0	AD0 — Port C, Bit 6
1	AD1 — Port C, Bit 5
2	AD2 — Port C, Bit 4
3	AD3 — Port C, Bit 3
4	V_{REFH} — Port C, Bit 7
5	$(V_{REFH} + V_{SS})/2$
6	V_{SS}
7	Reserved

9.5 A/D Conversion Data Register (ADC)

This register contains the output of the A/D converter. See Figure 9-1.

		Bit 7	6	5	4	3	2	1	Bit 0
ADC	Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
$001D	Write:								
	Reset:	X	X	X	X	X	X	X	X

= Unimplemented

Figure 9-2. A/D Conversion Data Register

9.6 A/D Subsystem During Wait/Halt Modes

The A/D subsystem continues normal operation during wait and halt modes. To decrease power consumption during wait or halt, the ADON bit in the ADSC register and the EERC bit in the EEPROG register should be cleared if the A/D subsystem is not being used.

9.7 A/D Subsystem Operation During Stop Mode

When the stop mode is enabled, execution of the STOP instruction will terminate all A/D subsystem functions. Any pending conversion is aborted. When the oscillator resumes operation upon leaving the stop mode, a finite amount of time passes before the A/D subsystem stabilizes sufficiently to provide conversions at its rated accuracy. The delays built into the MC68HC805P18 when coming out of stop mode are sufficient for this purpose. No explicit delays need to be added to the application software.

SECTION 10
16-BIT TIMER

10.1 Introduction

The MC68HC805P18 MCU contains a single 16-bit programmable timer with an input capture function and an output compare function. The 16-bit timer is driven by the output of a fixed divide-by-four prescaler operating from the PH2 clock. The 16-bit timer may be used for many applications including input waveform measurement, while simultaneously generating an output waveform. Pulse widths can vary from microseconds to seconds depending on the oscillator frequency selected. The 16-bit timer is also capable of generating periodic interrupts. See Figure 10-1.

Because the timer has a 16-bit architecture, each function is represented by two registers. Each register pair contains the high and low byte of that function. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

NOTE

The I bit in the condition code register (CCR) should be set while manipulating both the high and low byte registers of a specific timer function. This prevents interrupts from occurring between the time the high and low bytes are accessed.
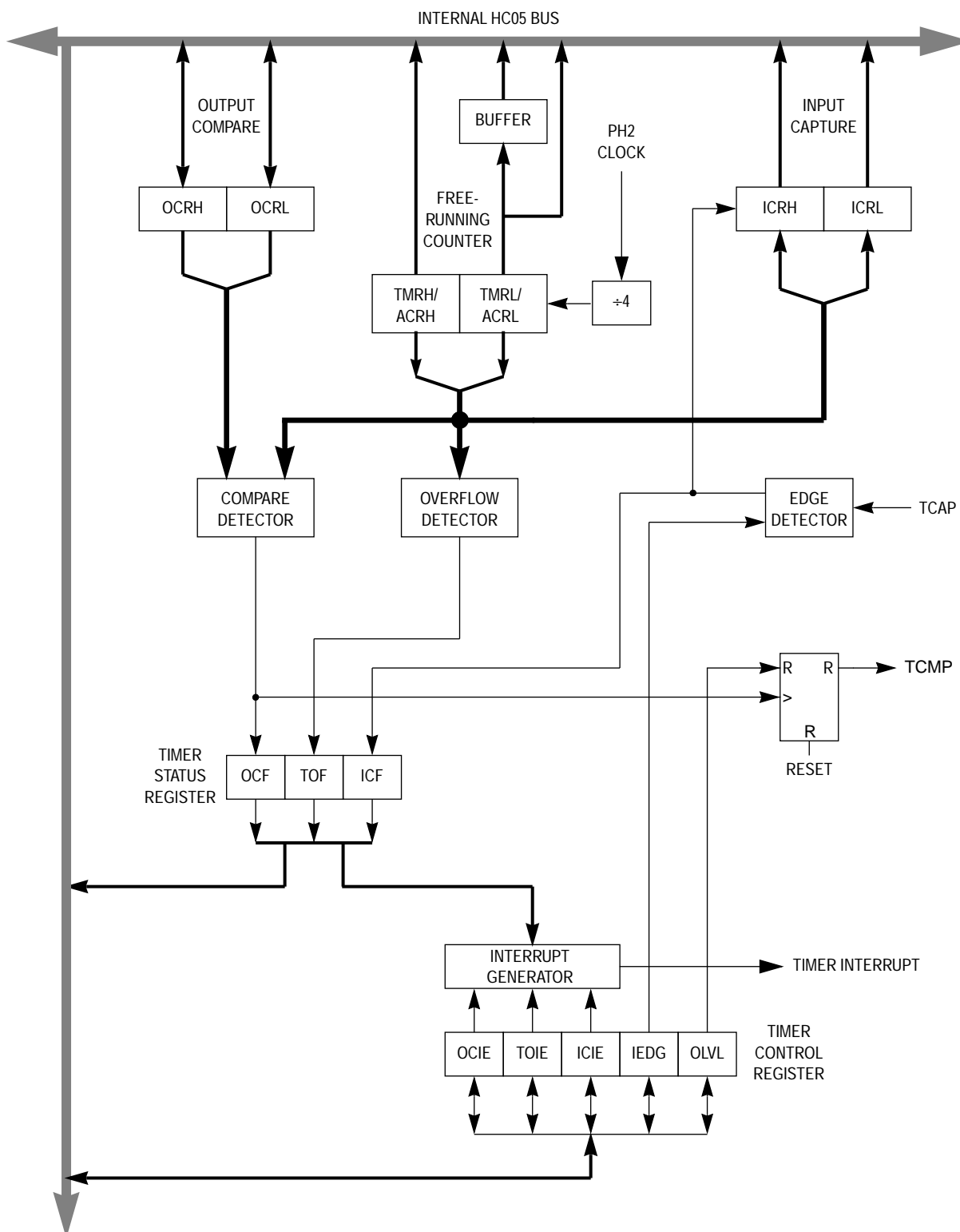
Figure 10-1. 16-Bit Timer Block Diagram

16-BIT TIMER

10.2 Timer

The key element of the programmable timer is a 16-bit free-running counter, or timer registers, preceded by a prescaler which divides the PH2 clock by four. The prescaler gives the timer a resolution of 2.0 microseconds when a 4-MHz crystal is used. The counter is incremented to increasing values during the low portion of the PH2 clock cycle.

The double byte free-running counter can be read from either of two locations: the timer registers (TMRH and TMRL) or the alternate counter registers (ACRH and ACRL). Both locations will contain identical values. A read sequence containing only a read of the LSB of the counter (TMRL/ACRL) will return the count value at the time of the read. If a read of the counter accesses the MSB first (TMRH/ACRH), it causes the LSB (TMRL/ACRL) to be transferred to a buffer. This buffer value remains fixed after the first MSB byte read, even if the MSB is read several times. The buffer is accessed when reading the counter LSB (TMRL/ACRL), and thus completes a read sequence of the total counter value. When reading either the timer or alternate counter registers, if the MSB is read, the LSB must also be read to complete the read sequence. See Figure 10-2 and Figure 10-3.

		Bit 7	6	5	4	3	2	1	Bit 0
TMRH $0018	Read:	TMRH7	TMRH6	TMRH5	TMRH4	TMRH3	TMRH2	TMRH1	TMRH0
	Write:								
	Reset:	1	1	1	1	1	1	1	1

|		Bit 7	6	5	4	3	2	1	Bit 0
TMRL $0019	Read:	TMRL7	TMRL6	TMRL5	TMRL4	TMRL3	TMRL2	TMRL1	TMRL0
	Write:								
	Reset:	1	1	1	1	1	1	0	0

= Unimplemented

Figure 10-2. Timer Registers (TMRH/TMRL)

		Bit 7	6	5	4	3	2	1	Bit 0
ACRH $001A	Read:	ACRH7	ACRH6	ACRH5	ACRH4	ACRH3	ACRH2	ACRH1	ACRH0
	Write:								
	Reset:	1	1	1	1	1	1	1	1

		Bit 7	6	5	4	3	2	1	Bit 0
ACRL $001B	Read:	ACRL7	ACRL6	ACRL5	ACRL4	ACRL3	ACRL2	ACRL1	ACRL0
	Write:								
	Reset:	1	1	1	1	1	1	0	0

= Unimplemented

Figure 10-3. Alternate Counter Registers (ACRH/ACRL)

The timer registers and alternate counter registers can be read at any time without affecting their value. However, the alternate counter registers differ from the timer registers in one respect: Aread of the timer register MSB can clear the timer overflow flag (TOF). Therefore, the alternate counter registers can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF. See Figure 10-4.

PH2
CLOCK

16-BIT
FREE-RUNNING $FFFE $FFFF $0000 $0001 $0002
COUNTER

TIMER
OVERFLOW
FLAG (TOF)
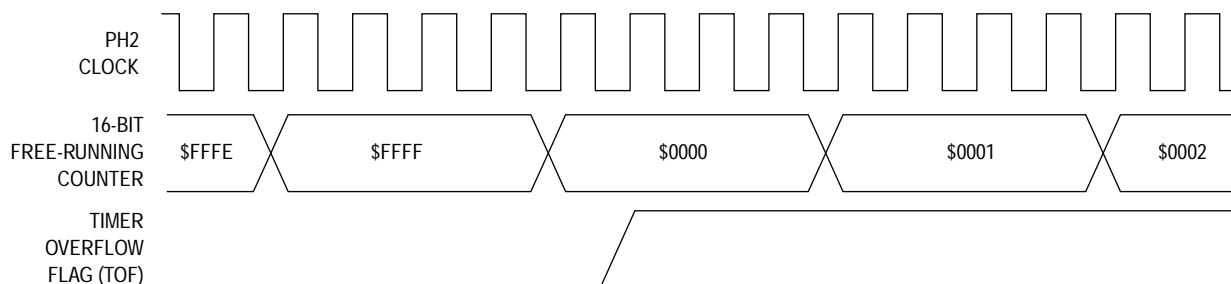
NOTE: The TOF bit is set at timer state T11 (transition of counter from $FFFF to $0000). It is cleared by reading the timer status register (TSR) during the high portion of the PH2 clock followed by reading the LSB of the counter register pair (TCRL).

Figure 10-4. State Timing Diagram for Timer Overflow

The free-running counter is initialized to $FFFC during reset and is a read-only register. During power-on-reset (POR), the counter is initialized to $FFFC and begins counting after the oscillator startup delay. Because the counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the counter repeats every 262,144 PH2 clock cycles (524,288 oscillator cycles). When the free-running counter rolls over from $FFFF to $0000, the timer overflow flag bit (TOF) in register TSR is set. An interrupt can also be enabled when counter rollover occurs by setting the timer overflow interrupt enable bit (TOIE) in register TCR. See Figure 10-5.
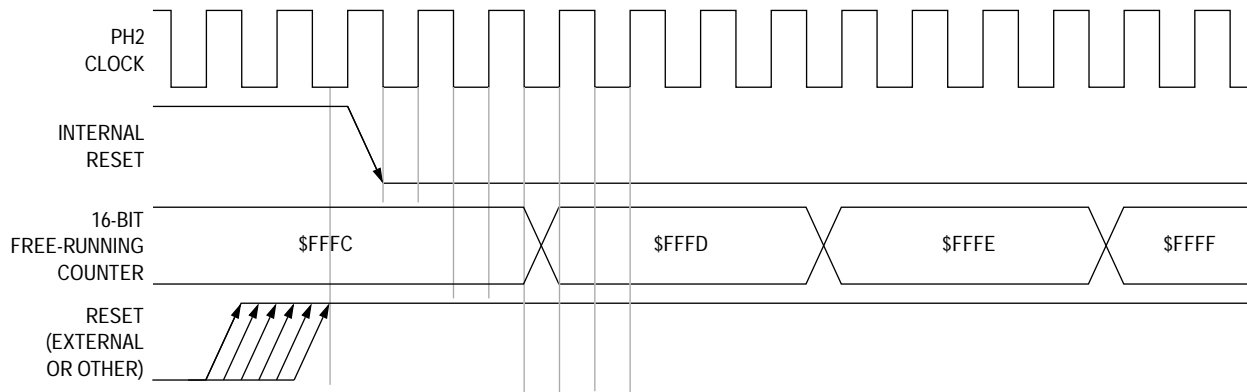
NOTE: The counter and control registers are the only 16-bit timer registers affected by reset.

Figure 10-5. State Timing Diagram for Timer Reset

10.3 Output Compare

The output compare function may be used to generate an output waveform and/or as an elapsed time indicator. All of the bits in the output compare register pair OCRH/OCRL are readable and writable and are not altered by the 16-bit timer's control logic. Reset does not affect the contents of these registers. If the output compare function is not utilized, its registers can be used for data storage. See Figure 10-3.

		Bit 7	6	5	4	3	2	1	Bit 0
OCRH $0016	Read: Write:	OCRH7	OCRH6	OCRH5	OCRH4	OCRH3	OCRH2	OCRH1	OCRH0
	Reset:	X	X	X	X	X	X	X	X

		Bit 7	6	5	4	3	2	1	Bit 0
OCRL $0017	Read: Write:	OCRL7	OCRL6	OCRL5	OCRL4	OCRL3	OCRL2	OCRL1	OCRL0
	Reset:	X	X	X	X	X	X	X	X

Figure 10-6. Output Compare Registers (OCRH/OCRL)

The contents of the output compare registers are compared with the contents of the free-running counter once every four PH2 clock cycles. If a match is found, the output compare flag bit (OCF) is set and the output level bit (OLVL) is clocked to the output latch. The values in the output compare registers and output level bit should be changed after each successful comparison to control an output waveform or to establish a new elapsed timeout. An interrupt can also accompany a successful output compare if the output compare interrupt enable bit (OCIE) is set.

After a CPU write cycle to the MSB of the output compare register pair (OCRH), the output compare function is inhibited until the LSB (OCRL) is written. Both bytes must be written if the MSB is written. A write made only to the LSB will not inhibit the compare function. The free-running counter increments every four PH2 clock cycles. The minimum time required to update the output compare registers is a function of software rather than hardware.

The output compare output level bit (OLVL) will be clocked to its output latch regardless of the state of the output compare flag bit (OCF). A valid output compare must occur before the OLVL bit is clocked to its output latch (TCMP).

Since neither the output compare flag (OCF) nor the output compare registers are affected by reset, care must be exercised when initializing the output compare function. The following procedure is recommended:

1. Block interrupts by setting the I bit in the condition code register (CCR).

2. Write the MSB of the output compare register pair (OCRH) to inhibit further compares until the LSB is written.

3. Read the timer status register (TSR) to arm the output compare flag (OCF).

4. Write the LSB of the output compare register pair (OCRL) to enable the output compare function and to clear its flag (and interrupt).

5. Unblock interrupts by clearing the I bit in the CCR.

This procedure prevents the output compare flag bit (OCF) from being set between the time it is read and the time the output compare registers are updated. A software example is shown in Figure 10-7.

```
9B          SEI                    BLOCK INTERRUPTS
.    .    .        .           .
.    .    .        .           .
B6   XX   LDA      DATAH        HI BYTE FOR COMPARE
BE   XX   LDX      DATAL        LO BYTE FOR COMPARE
B7   16   STA      OCRH         INHIBIT OUTPUT COMPARE
B6   13   LDA      TSR          ARM OCF BIT TO CLEAR
BF   17   STX      OCRL         READY FOR NEXT COMPARE
.    .    .        .           .
```

Figure 10-7. Output Compare Software Initialization Example

10.4 Input Capture

Two 8-bit read-only registers (ICRH and ICRL) make up the 16-bit input capture. They are used to latch the value of the free-running counter after a defined transition is sensed by the input capture edge detector.

NOTE

The input capture edge detector contains a Schmitt trigger to improve noise immunity.

The edge that triggers the counter transfer is defined by the input edge bit (IEDG) in register TCR. Reset does not affect the contents of the input capture registers. See Figure 10-3.

		Bit 7	6	5	4	3	2	1	Bit 0
ICRH $0014	Read: Write:	ICRH7	ICRH6	ICRH5	ICRH4	ICRH3	ICRH2	ICRH1	ICRH0
	Reset:	X	X	X	X	X	X	X	X

		Bit 7	6	5	4	3	2	1	Bit 0
ICRL $0015	Read: Write:	ICRL7	ICRL6	ICRL5	ICRL4	ICRL3	ICRL2	ICRL1	ICRL0
	Reset:	X	X	X	X	X	X	X	X

Figure 10-8. Input Compare Registers (ICRH/ICRL)

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the PH2 clock preceding the external transition (see Figure 10-9). This delay is required for internal synchronization. Resolution is affected by the prescaler, allowing the free-running counter to increment once every four PH2 clock cycles.

The contents of the free-running counter are transferred to the input capture registers on each proper signal transition regardless of the state of the input capture flag bit (ICF) in register TSR. The input capture registers always contain the free-running counter value which corresponds to the most recent input capture.

After a read of the MSB of the input capture register pair (ICRH), counter transfers are inhibited until the LSB of the register pair (ICRL) is also read. This characteristic forces the minimum pulse period attainable to be determined by the time required to execute an input capture software routine in an application.

Reading the LSB of the input capture register pair (ICRL) does not inhibit transfer of the free-running counter. Again, minimum pulse periods are ones which allow software to read the LSB of the register pair (ICRL) and perform needed operations. There is no conflict between reading the LSB (ICRL) and the free-running counter transfer, since they occur on opposite edges of the PH2 clock.

NOTE: If the input edge occurs in the shaded area from one T10 timer state to the other T10 timer state, the input capture flag is set during the next T11 timer state.
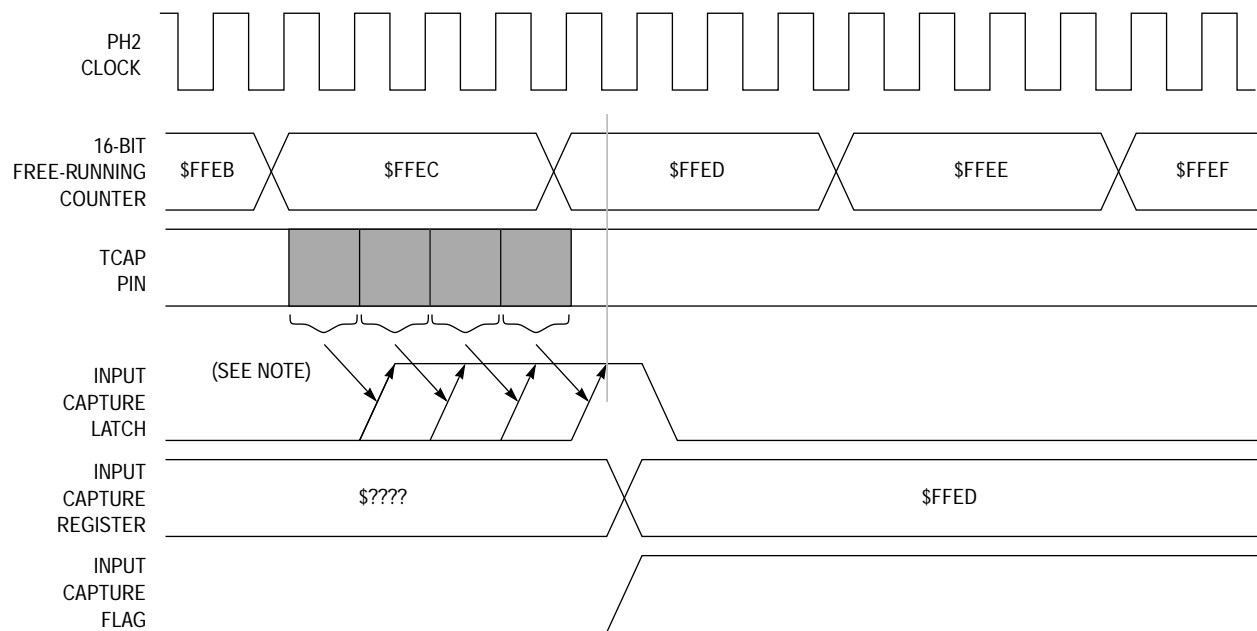
Figure 10-9. State Timing Diagram for Input Capture

10.5 Timer Control Register (TCR)

The timer control (TCR) shown in Figure 10-3 and free-running counter (TMRH, TMRL, ACRH, and ACRL) registers are the only registers of the 16-bit timer affected by reset. The output compare port (TCMP) is forced low after reset and remains low until OLVL is set and a valid output compare occurs.

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

☐ = Unimplemented

Figure 10-10. Timer Control Register (TCR)

ICIE — Input Capture Interrupt Enable

Bit 7, when set, enables input capture interrupts to the CPU. The interrupt will occur at the same time bit 7 (ICF) in the TSR register is set.

OCIE —Output Compare Interrupt Enable

Bit 6, when set, enables output compare interrupts to the CPU. The interrupt will occur at the same time bit 6 (OCF) in the TSR register is set.

TOIE — Timer Overflow Interrupt Enable

Bit 5, when set, enables timer overflow (rollover) interrupts to the CPU. The interrupt will occur at the same time bit 5 (TOF) in the TSR register is set.

IEDG — Input Capture Edge Select

Bit 1 selects which edge of the input capture signal will trigger a transfer of the contents of the free-running counter registers to the input capture registers. Clearing this bit will select the falling edge; setting it selects the rising edge.

OLVL — Output Compare Output Level Select

Bit 0 selects the output level (high or low) that is clocked into the output compare output latch at the next successful output compare.

10.6 Timer Status Register (TSR)

Reading the timer status register (TSR) satisfies the first condition required to clear status flags and interrupts. See Figure 10-3. The only remaining step is to read (or write) the register associated with the active status flag (and/or interrupt). This method does not present any problems for input capture or output compare functions.

However, a problem can occur when using a timer interrupt function and reading the free-running counter at random times to, for example, measure an elapsed time. If the proper precautions are not designed into the application software, a timer interrupt flag (TOF) could unintentionally be cleared if:

1. The TSR is read when bit 5 (TOF) is set, and

2. The LSB of the free-running counter is read, but not for the purpose of servicing the flag or interrupt.

The alternate counter registers (ACRH and ACRL) contain the same values as the timer registers (TMRH and TMRL). Registers ACRH and ACRL can be read at any time without affecting the timer overflow flag (TOF) or interrupt.

		Bit 7	6	5	4	3	2	1	Bit 0
TSR $0013	Read:	ICF	OCR	TOF	0	0	0	0	0
	Write:								
	Reset:	X	X	X	0	0	0	0	0

= Unimplemented

Figure 10-11. Timer Status Register (TSR)

ICF — Input Capture Flag

    Bit 7 is set when the edge specified by IEDG in register TCR has been sensed by the input capture edge detector fed by pin TCAP. This flag and the input capture interrupt can be cleared by reading register TSR followed by reading the LSB of the input capture register pair (ICRL).

OCF — Output Compare Flag

    Bit 6 is set when the contents of the output compare registers match the contents of the free-running counter. This flag and the output compare interrupt can be cleared by reading register TSR followed by writing the LSB of the output compare register pair (OCRL).

TOF — Timer Overflow Flag

Bit 5 is set by a rollover of the free-running counter from $FFFF to $0000. This flag and the timer overflow interrupt can be cleared by reading register TSR followed by reading the LSB of the timer register pair (TMRL).

10.7 Timer Operation During Wait/Halt Modes

During wait and halt modes, the 16-bit timer continues to operate normally and may generate an interrupt to trigger the MCU out of the wait/halt mode.

10.8 Timer Operation During Stop Mode

When the MCU enters the stop mode, the free-running counter stops counting (the PH2 clock is stopped). It remains at that particular count value until the stop mode is exited by applying a low signal to the $\overline{IRQ}$ pin, at which time the counter resumes from its stopped value as if nothing had happened. If stop mode is exited via an external $\overline{RESET}$ (logic low applied to the $\overline{RESET}$ pin), the counter is forced to $FFFC.

If a valid input capture edge occurs at the TCAP pin during stop mode the input capture detect circuitry will be armed. This action does not set any flags or "wake up" the MCU, but when the MCU does "wake up" there will be an active input capture flag (and data) from the first valid edge. If the stop mode is exited by an external $\overline{RESET}$, no input capture flag or data will be present even if a valid input capture edge was detected during stop mode.

SECTION 11
SERIAL INPUT/OUTPUT PORT

11.1 Introduction

The simple synchronous serial input/output port (SIOP) subsystem is designed to provide efficient serial communications between peripheral devices or other MCUs. The SIOP is implemented as a 3-wire master/slave system with serial clock (SCK), serial data Input (SDI), and serial data output (SDO). A block diagram of the SIOP is shown in Figure 11-1.

The SIOP subsystem shares its input/output pins with port B. When the SIOP is enabled (SPE bit set in register SCR), port B data direction registers (DDR) and data registers are modified by the SIOP. Although port B DDR and data registers can be altered by application software, these actions could affect the transmitted or received data.
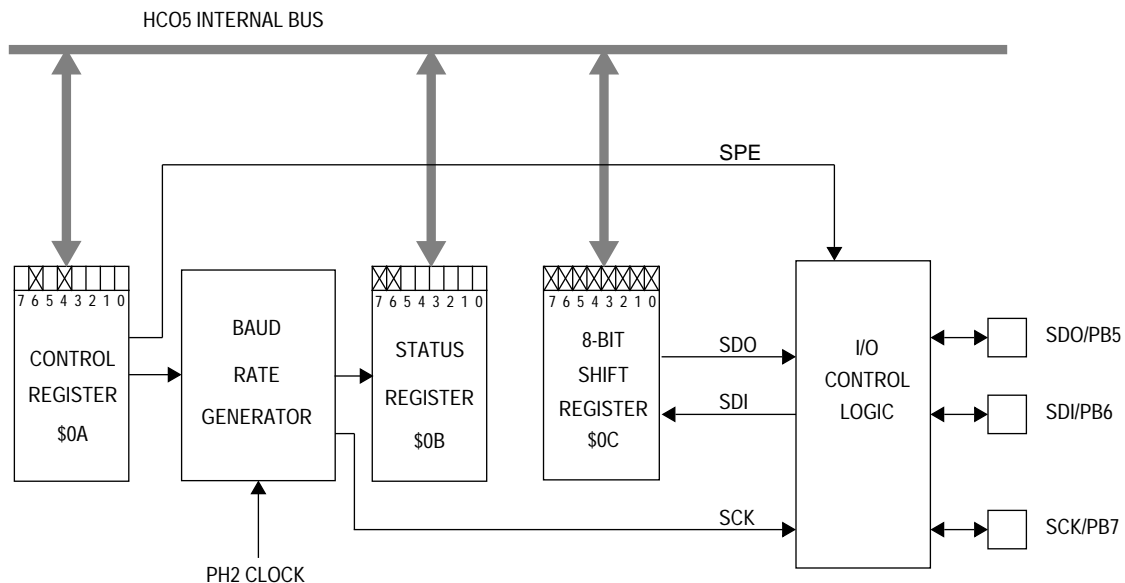
Figure 11-1. SIOP Block Diagram

11.2 SIOP Signal Format

The SIOP subsystem is software configurable for master or slave operation. No external mode selection inputs are available (such as the slave select pin).

11.2.1 Serial Clock (SCK)

The state of the SCK output normally remains a logic one during idle periods between data transfers. The first falling edge of SCK signals the beginning of a data transfer. At this time the first bit of received data is accepted at the SDI pin and the first bit of transmitted data is presented at the SDO pin (see Figure 11-2). Data is captured at the SDI pin on the rising edge of SCK, and the first bit of transmitted data is presented at the SDO pin. The transfer is terminated upon the eighth rising edge of SCK.
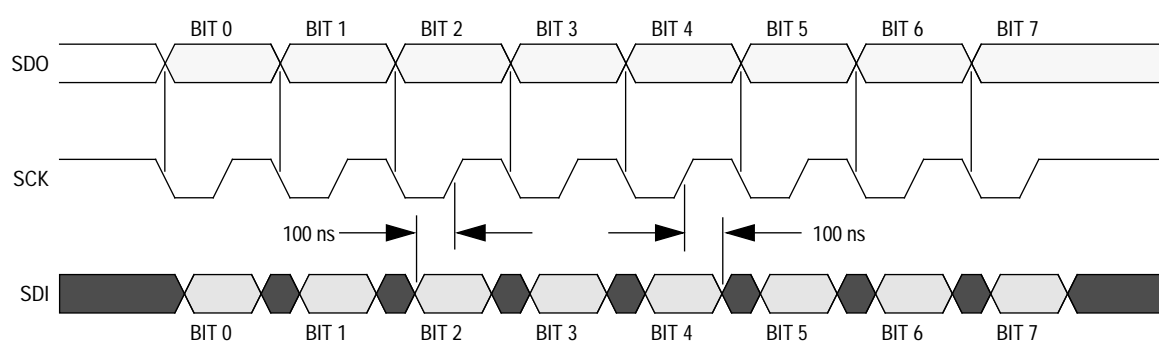
Figure 11-2. SIOP Timing Diagram

The master and slave modes of operation differ only by the sourcing of SCK. In master mode, SCK is driven from an internal source within the MCU. In slave mode, SCK is driven from a source external to the MCU. The SCK frequency is programmable via the mask option register 1 (MOR1). Available rates are OSC divided by 2, 4, 8, or 16.

NOTE

OSC divided by 2 is four times faster than the standard rate available on the 68HC05P6.

Refer to **8.4 Mask Option Registers (MOR)** for a description of available mask option registers.

11.2.2 Serial Data Input (SDI)

The SDI pin becomes an input as soon as the SIOP subsystem is enabled. New data is presented to the SDI pin on the falling edge of SCK. Valid data must be present at least 100 nanoseconds before the rising edge of SCK and remain valid for 100 nanoseconds after the rising edge of SCK. See Figure 11-2.

11.2.3 Serial Data Output (SDO)

The SDO pin becomes an output as soon as the SIOP subsystem is enabled. Prior to enabling the SIOP, PB5 can be initialized to determine the beginning state. While the SIOP is enabled, PB5 cannot be used as a standard output since that pin is connected to the last stage of the SIOP serial shift register. The data can be transmitted in either MSB first format or the LSB format by programming the MOR1.

On the first falling edge of SCK, the first data bit will be shifted out to the SDO pin. The remaining data bits will be shifted out to the SDI pin on subsequent falling edges of SCK. The SDO pin will present valid data at least 100 nanoseconds before the rising edge of the SCK and remain valid for 100 nanoseconds after the rising edge of SCK. See Figure 11-2.

11.3 SIOP Registers

The SIOP is programmed and controlled by the SIOP control register (SCR) located at address $000A, the SIOP status register (SSR) located at address $000B, and the SIOP data register (SDR) located at address $000C.

11.3.1 SIOP Control Register (SCR)

This register is located at address $000A and contains two bits. Figure 11-3 shows the position of each bit in the register and indicates the value of each bit after reset.

		Bit 7	6	5	4	3	2	1	Bit 0
SCR $000A	Read:	0	SPE	0	MSTR	0	0	0	0
	Write:								
	Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 11-3. SIOP Control Register

SPE — Serial Peripheral Enable

When set, the SPE bit enables the SIOP subsystem such that SDO/PB5 is the serial data output, SDI/PB6 is the serial data input, and SCK/PB7 is a serial clock input in the slave mode or a serial clock output in the master mode. Port B DDR and data registers can be manipulated as usual (except for PB5); however, these actions could affect the transmitted or received data.

The SPE bit is readable and writable at any time. Clearing the SPE bit while a transmission is in progress will 1) abort the transmission, 2) reset the serial bit counter, and 3) convert the port B/SIOP port to a general-purpose I/O port. Reset clears the SPE bit.

MSTR — Master Mode Select

When set, the MSTR bit configures the serial I/O port for master mode. A transfer is initiated by writing to the SDR. Also, the SCK pin becomes an output providing a synchronous data clock dependent upon the oscillator frequency. When the device is in slave mode, the SDO and SDI pins do not change function. These pins behave exactly the same in both the master and slave modes.

The MSTR bit is readable and writable at any time regardless of the state of the SPE bit. Clearing the MSTR bit will abort any transfers that may have been in progress. Reset clears the MSTR bit, placing the SIOP subsystem in slave mode.

11.3.2 SIOP Status Register (SSR)

This register is located at address $000B and contains two bits. **Figure 11-3** shows the position of each bit in the register and indicates the value of each bit after reset.

		Bit 7	6	5	4	3	2	1	Bit 0
SSR $000B	Read:	SPIF	DCOL	0	0	0	0	0	0
	Write:								
	Reset:	0	0	0	0	0	0	0	0

☐ = Unimplemented

Figure 11-4. SIOP Status Register

SPIF — Serial Port Interface Flag

SPIF is a read-only status bit that is set on the last rising edge of SCK and indicates that a data transfer has been completed. It has no effect on any future data transfers and can be ignored. The SPIF bit is cleared by reading the SSR followed by a read or write of the SDR. If the SPIF is cleared before the last rising edge of SCK, it will be set again on the last rising edge of SCK. Reset clears the SPIF bit.

DCOL — Data Collision

DCOL is a read-only status bit which indicates that an illegal access of the SDR has occurred. The DCOL bit will be set when reading or writing the SDR after the first falling edge of SCK and before SPIF is set. Reading or writing the SDR during this time will result in invalid data being transmitted or received.

The DCOL bit is cleared by reading the SSR (when the SPIF bit is set) followed by a read or write of the SDR. If the last part of the clearing sequence is done after another transfer has started, the DCOL bit will be set again. Reset clears the DCOL bit.

11.3.3 SIOP Data Register (SDR)

This register is located at address $000C and serves as both the transmit and receive data register. Writing to this register will initiate a message transmission if the SIOP is in master mode. The SIOP subsystem is not double buffered and any write to this register will destroy the previous contents. The SDR can be read at any time; however, if a transfer is in progress, the results may be ambiguous and the DCOL bit will be set. Writing to the SDR while a transfer is in progress can cause invalid data to be transmitted and/or received. Figure 11-3 shows the position of each bit in the register. This register is not affected by reset.

		Bit 7	6	5	4	3	2	1	Bit 0
SDR $000C	Read: Write:	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
	Reset:				Unaffected by reset				
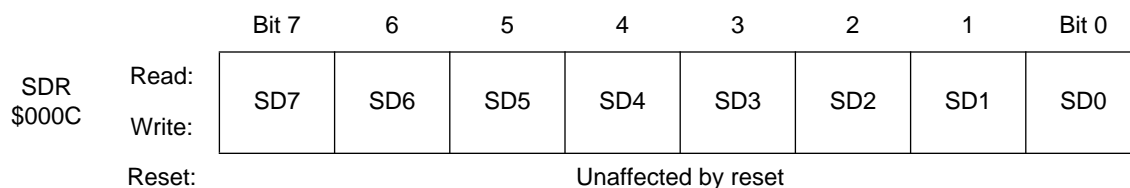
Figure 11-5. SIOP Data Register

SECTION 12
INSTRUCTION SET

12.1 Introduction

This section describes the addressing modes and instruction types.

12.2 Addressing Modes

The CPU uses eight addressing modes for flexibility in accessing data. The addressing modes define the manner in which the CPU finds 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

12.2.1 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 memory address and are one byte long.

12.2.2 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 memory address and are two bytes long. The opcode is the first byte, and the immediate data value is the second byte.

12.2.3 Direct

Direct instructions can access any of the first 256 memory addresses 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. BRSET and BRCLR are 3-byte instructions that use direct addressing to access the operand and relative addressing to specify a branch destination.

12.2.4 Extended

Extended instructions use only three bytes to 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.

12.2.5 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 conditional 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.

12.2.6 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 conditional 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.

12.2.7 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 conditional 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. These instructions can address any location in memory.

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.

12.2.8 Relative

Relative addressing is only for branch instructions. If the branch condition is true, the CPU finds the conditional 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.

12.3 Instruction Types

The MCU instructions fall into five categories:

- Register/Memory instructions
- Read-Modify-Write instructions
- Jump/Branch instructions
- Bit Manipulation instructions
- Control instructions

12.3.1 Register/Memory Instructions

Most of these instructions use two operands. One operand is in either the accumulator or the index register. The CPU finds the other operand in memory. Table 12-1 lists the register/memory instructions.

Table 12-1. 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

12.3.2 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. The test for negative or zero instruction (TST) is an exception to the read-modify-write sequence because it does not write a replacement value. Table 12-2 lists the read-modify-write instructions.

Table 12-2. Read-Modify-Write Instructions

Instruction	Mnemonic
Arithmetic Shift Left	ASL
Arithmetic Shift Right	ASR
Clear Bit in Memory	BCLR
Set Bit in Memory	BSET
Clear	CLR
Complement (One's Complement)	COM
Decrement	DEC
Increment	INC
Logical Shift Left	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

12.3.3 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. All branch instructions use relative addressing.

Bit test and branch 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 conditional 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 12-3 lists the jump and branch instructions.

Table 12-3. 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 $\overline{IRQ}$ Pin High	BIH
Branch if $\overline{IRQ}$ 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

12.3.4 Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory. Port registers, port data direction registers, timer registers, and on-chip RAM locations are in the first 256 bytes of memory. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations. Bit manipulation instructions use direct addressing. Table 12-4 lists these instructions.

Table 12-4. Bit Manipulation Instructions

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

12.3.5 Control Instructions

These register reference instructions control CPU operation during program execution. Control instructions, listed in Table 12-5, use inherent addressing.

Table 12-5. Control Instructions

Instruction	Mnemonic
Clear Carry Bit	CLC
Clear Interrupt Mask	CLI
No Operation	NOP
Reset Stack Pointer	RSP
Return from Interrupt	RTI
Return from Subroutine	RTS
Set Carry Bit	SEC
Set Interrupt Mask	SEI
Stop Oscillator and Enable $\overline{IRQ}$ Pin	STOP
Software Interrupt	SWI
Transfer Accumulator to Index Register	TAX
Transfer Index Register to Accumulator	TXA
Stop CPU Clock and Enable Interrupts	WAIT

12.4 Instruction Set Summary

Table 12-6 is an alphabetical list of all M68HC05 instructions and shows the effect of each instruction on the condition code register.

Table 12-6. Instruction Set Summary
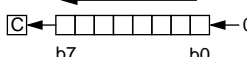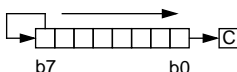
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr* 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* AND *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)	C ◄— [b7 ... b0] ◄— 0	—	—	↕	↕	↕	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	[b7 ... b0] —► C	—	—	↕	↕	↕	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

Table 12-6. Instruction Set Summary (Continued)
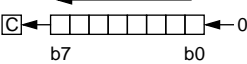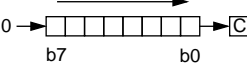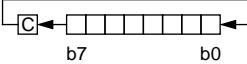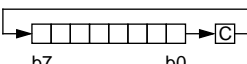
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BHCC *rel*	Branch if Half-Carry Bit Clear	PC ← (PC) + 2 + *rel* ? H = 0	—	—	—	—	—	REL	28	rr	3
BHCS *rel*	Branch if Half-Carry Bit Set	PC ← (PC) + 2 + *rel* ? H = 1	—	—	—	—	—	REL	29	rr	3
BHI *rel*	Branch 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
BIH *rel*	Branch if $\overline{\text{IRQ}}$ Pin High	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 1	—	—	—	—	—	REL	2F	rr	3
BIL *rel*	Branch if $\overline{\text{IRQ}}$ Pin Low	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 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 p	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
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
BRN *rel*	Branch Never	PC ← (PC) + 2 + *rel* ? 1 = 0	—	—	—	—	—	REL	21	rr	3

Table 12-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR H	I	N	Z	C	Address Mode	Opcode	Operand	Cycles
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
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 ← (M̄) = $FF − (M) A ← (Ā) = $FF − (M) X ← (X̄) = $FF − (M) M ← (M̄) = $FF − (M) M ← (M̄) = $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)	—	—	↕	↕	1	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

Table 12-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR H	I	N	Z	C	Address Mode	Opcode	Operand	Cycles
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
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 ← Conditional 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)	C ← [b7 ... b0] ← 0	—	—	↕	↕	↕	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 → [b7 ... b0] → C	—	—	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

Table 12-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
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	ii 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
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		6
RTS	Return from Subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)						INH			
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

Table 12-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
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
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		—	↕	—	—	—	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

Table 12-7. Opcode Map

Cell format: **Mnemonic** (cycles, bytes). Addressing mode is given by the column header.

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

Legend — addressing modes:

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

Cell key:
- MSB of Opcode in Hexadecimal
- LSB of Opcode in Hexadecimal
- Number of Cycles
- Opcode Mnemonic
- Number of Bytes / Addressing Mode

Example: BRSET0 — opcode 0x00, 5 cycles, 3 bytes, DIR

SECTION 13
ELECTRICAL SPECIFICATIONS

13.1 Introduction

This section contains electrical and timing specifications for the MC68HC805P18.

13.2 Maximum Ratings

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{IN}	V_{SS} −0.3 to V_{DD} + 0.3	V
Factory Mode ($\overline{IRQ}$ Pin Only)	V_{IN}	V_{SS} −0.3 to 2 x V_{DD}	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Storage Temperature Range	T_{STG}	−65 to +150	°C

NOTE: Voltages referenced to V_{SS}

13.3 Operating Temperature Range

Rating	Symbol	Value	Unit

Operating Temperature Range
    MC68HC805P18 (Standard)
    MC68HC805P18 (Extended)
    MC68HC805P18 (Automotive)
	T_A	T_L to T_H
		0 to +70
		−40 to +85
		−40 to +125
			°C

NOTE

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

13.4 Thermal Characteristics

Characteristic	Symbol	Value	Unit

Thermal Resistance
   Plastic
   SOIC
	θ_{JA}	60
		60
			°C/W

13.5 Power Considerations

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

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

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 $P_{I/O}$):

$$P_D = K \div (T_J + 273 \, °C) \tag{2}$$

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

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

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.

13.6 DC Electrical Characteristics (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = −40 °C to +125 °C)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage I_{Load} = 10.0 µA	V_{OL} V_{OH}	— V_{DD} −0.1	— —	0.1 —	V
Output High Voltage (I_{Load} = −0.8 mA) PA0–PA7, PB5–PB7, PC0–PC7, PD5/CKOUT	V_{OH}	V_{DD} −0.8	—	—	V
Output Low Voltage (I_{Load} = 1.6 mA) PA0–PA7, PB5–PB7, PC0–PC7, PD5/CKOUT	V_{OL}	—	—	0.4	V
Input High Voltage PA0–PA7, PB5–PB7, PC0–PC7, PD5/CKOUT, TCAP/PD7, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA0–PA7, PB5–PB7, PC0–PC7, PD5, TCAP/PD7, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V

Supply Current
Run
Wait (see Note 3)
Stop (see Note 8)
    25 °C
    0 °C to +70 °C (Standard)
    −40 °C to +85 °C (Extended)
    −40 °C to +125 °C (Automotive
	I_{DD}	—
		—
		
		—
		—
		—
		—
			4.75
			2.75
			
			TBD
			TBD
			TBD
			TBD
				7.50
				5.00
				
				350
				400
				500
				500
					mA
					mA
					
					µA
					µA
					µA
					µA

I/O Ports Hi-Z Leakage Current PA0–PA7, PB5–PB7, PC0–PC7, PD5/CKOUT, TCAP/PD7	I_{IL}	—	—	±10	µA
I/O Ports Switch Resistance (Pullup Enabled PA0–PA7)	R_{PTA}	62	—	102	k
A/D Ports Hi-Z Leakage Current PC3–PC7	I_{IL}	—	—	±1	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, OSC1	I_{IN}	—	—	±1	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$	C_{OUT} C_{IN}	— —	— —	12 8	pF
EEPROM Program/Erase Time (128 Byte Array) Byte Block (Erase Only) Bulk (Erase Only)		— — —	2 5 10	10 15 50	ms
Low Voltage Reset Voltage		3.6	3.8	—	V

NOTES:
1. All values shown reflect average measurements.
2. Typical vlaues at midpoint of voltage range, 25 °C only.
3. Wait I_{DD} with active systems: Timer, SIOP, and A/D.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} −0.2 V
6. Stop I_{DD} measured with OSC1 = V_{SS}
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.
8. Stop I_{DD} maximum values given with LVR option enabled.

13.7 Active Reset Characteristics

Rise Time	Fall Time	Pulse Width	C$_{Load}$	Pullup
0.5 µs	13 ns	2.4 µs	50 pF	10 K
1.0 µs	20 ns	2.7 µs	100 pF	10 K
2.5 µs	20 ns	2.7 µs	250 pF	10 K

NOTE: V_{DD} = 4.5 Vdc, V_{SS} = 0 Vdc, T_A = 125 °C

13.8 A/D Converter Characteristics

(V_{DD} = 4.5 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = –40 °C to +125 °C, unless otherwise noted)

Characteristic	Min	Max	Unit	Comments
Resolution	8	8	Bits	
Absolute Accuracy ($V_{DD} \geq V_{REFH}$ > 4.5)	—	± 1 1/2	LSB	Including quantization
Conversion Range V_{REFH}	V_{SS} V_{SS}	V_{REFH} V_{DD}	V	A/D accuracy may decrease proportionately as V_{REFH} is reduced below 4.5
Input Leakage AD0, AD1, AD2, AD3 V_{REFH}	— —	± 1 ± 1	µA µA	
Conversion Time (Includes Sampling Time)	32	32	t_{AD}*	
Monotonicity	Inherent (Within Total Error)			
Zero Input Reading	00	01	Hex	V_{in} = 0 V
Full-Scale Reading	FE	FF	Hex	V_{in} = V_{REFH}
Sample Time	12	12	t_{AD}*	
Input Capacitance	—	12	pF	
Analog Input Voltage	V_{SS}	V_{REFH}	V	

*t_{AD} = t_{cyc} if clock source equals MCU

13.9 SIOP Timing

(V_{DD} = 4.5 Vdc $\pm$ 10%, V_{SS} = 0 Vdc, T_A = –40 °C to +125 °C, unless otherwise note)

Number	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	$f_{SIOP(M)}$ $f_{SIOP(S)}$	1 dc	1 1	f_{OP}
1	Cycle Time Master Slave	$t_{SCK(M)}$ $t_{SCK(S)}$	1 —	1 1	t_{CYC}
2	SCK Low Time	t_{CYC}	238	—	ns
3	SDO Data Valid Time	t	—	200	ns
4	SDO Hold Time	t_{HO}	0	—	ns
5	SDI Setup Time	t_S	100	—	ns
6	SDI Hold Time	t_H	100	—	ns

NOTES:
1. f_{OP} = f_{OSC} ÷ 2 = 2.1 MHz max; t_{CYC} = 1 ÷ f_{OP}
2. In master mode, the SCK rate is determined by the programmable option in MOR1.
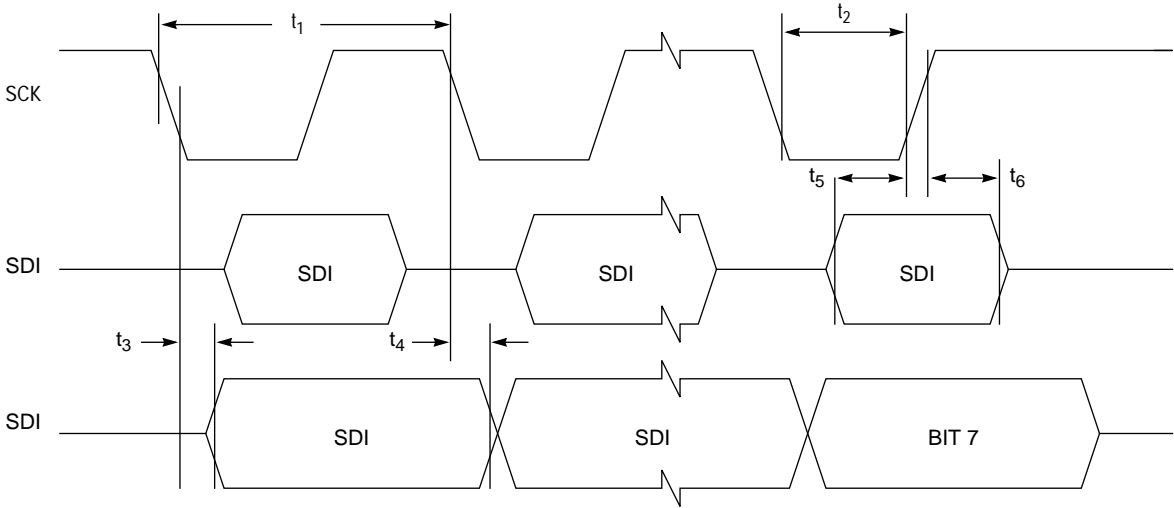
Figure 13-1. SIOP Timing Diagram

13.10 OSC Out Timing

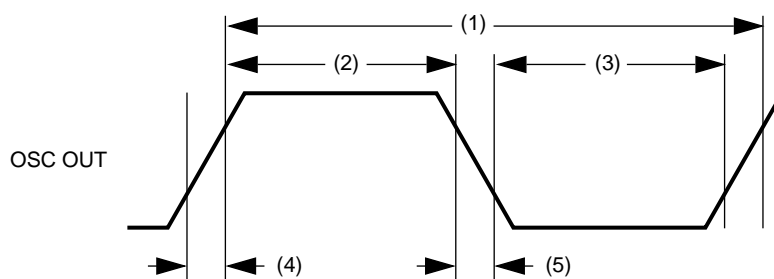| Characteristic	Symbol	Min	Max	Unit
Cycle Time	1*	476	—	ns
Rise Time	4*	3.5	12	ns
Fall Time	5*	7.5	27.5	ns
Pulse Width	2 and 3*	200	—	ns

*Refer to Figure 13-2


Figure 13-2. OSC Out Timing

NOTE

All timing is shown with respect to 20% and 70% V_{DD}. Maximum rise and fall times assume 44% duty cycle. Minimum rise and fall times assume 55% duty cycle

---

13.11 Control Timing

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = –40 °C to +125 °C, unless otherwise note)

Characteristic	Symbol	Min	Max	Unit

Frequency of Operation
    Crystal Option
    External Clock Option
	f_{OSC}	—
		dc
			4.2
			4.2
				MHz

Internal Operating Frequency
    Crystal ($f_{osc} \div 2$)
    External Clock ($f_{osc} \div 2$)
	f_{OP}	—
		dc
			2.1
			2.1
				MHz

Cycle Time	t_{CYC}	476	—	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}
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	125	—	ns
Interrupt Pulse Period	t_{ILIL}	*	—	t_{CYC}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns
A/D On Current Stabilization Time	t_{ADON}	—	100	µs
RC Oscillator Stabilization Time (A/D)	t_{RCON}	—	5.0	µs

* The minimum period t_{ILIL} should not be less than the number of cycles it takes to execute the interrupt service routine plus 21 t_{CYC}.

Figure 13-3. Power-On Reset and External Reset Timing Diagram
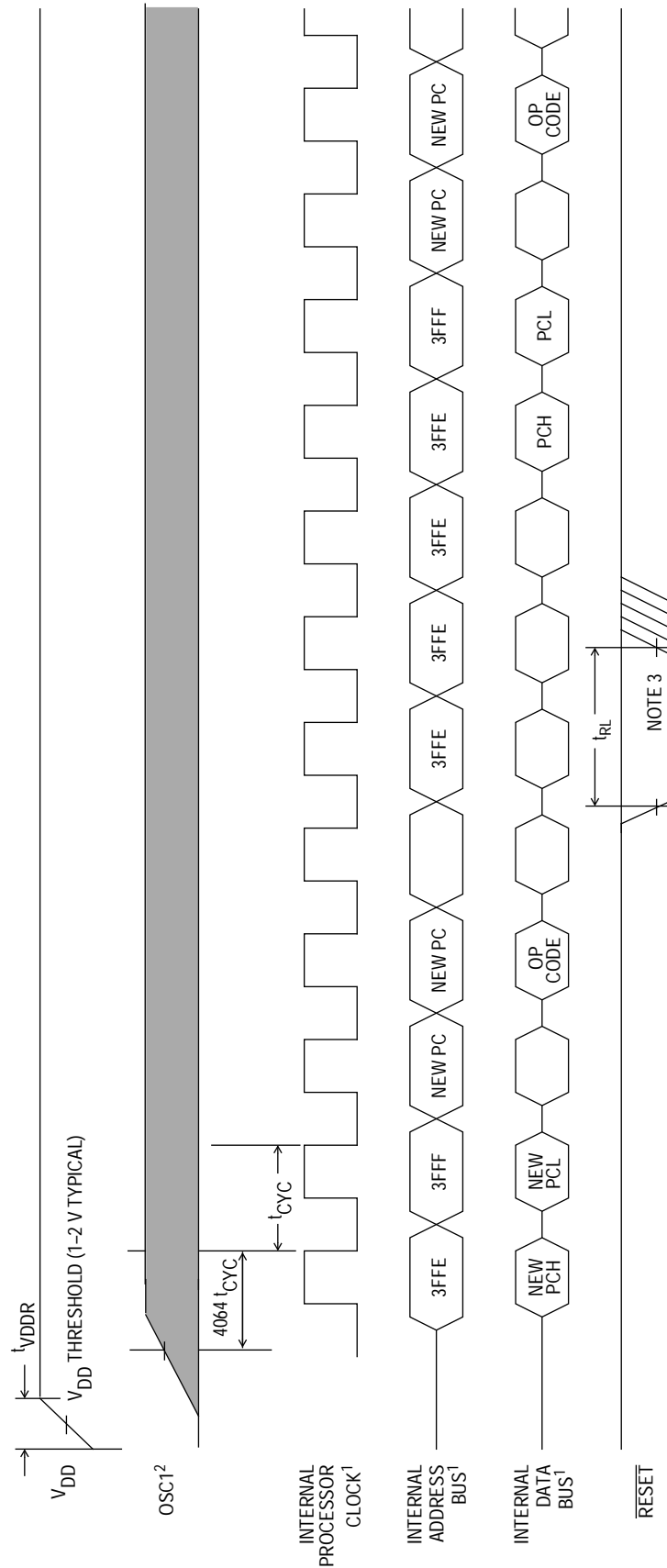
NOTES:
1. Internal timing signal and bus information not available externally.
2. OSC1 line is not meant to represent frequency. It is only used to represent time.
3. The next rising edge of the PH2 clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence.

SECTION 14
MECHANICAL SPECIFICATIONS

14.1 Introduction

This section provides package dimension drawings for the 28-pin dual in-line (DIP) or 28-pin small outline (SOIC) packages.

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

- Local Motorola Sales Office

- Motorola Mfax
  – Phone 602-244-6609
  – EMAIL rmfax0@email.sps.mot.com

- Worldwide Web (wwweb) at http://design-net.com

Follow Mfax or wwweb on-line instructions to retrieve the current mechanical specifications.

14.2 28-Pin Dual In-Line Package (Case #710)
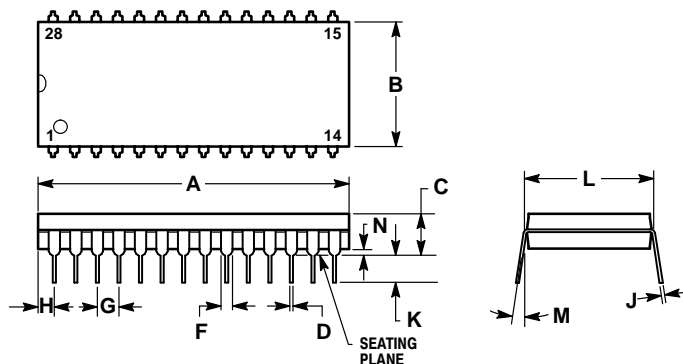
NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	36.45	37.21	1.435	1.465
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°	15°	0°	15°
N	0.51	1.02	0.020	0.040

14.3 28-Pin Small Outline Package (Case #751F)
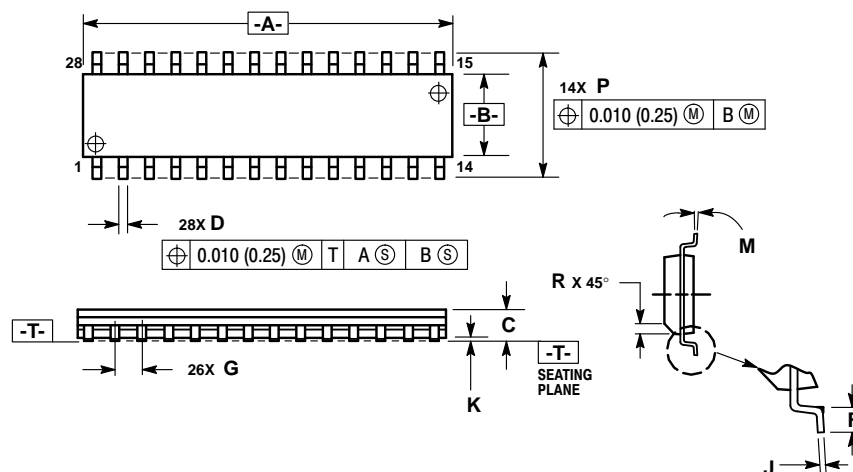
NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.13 (0.005) TOTAL IN EXCESS OF D DIMENSION AT MAXIMUM MATERIAL CONDITION.

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	17.80	18.05	0.701	0.711
B	7.40	7.60	0.292	0.299
C	2.35	2.65	0.093	0.104
D	0.35	0.49	0.014	0.019
F	0.41	0.90	0.016	0.035
G	1.27 BSC		0.050 BSC	
J	0.23	0.32	0.009	0.013
K	0.13	0.29	0.005	0.011
M	0°	8°	0°	8°
P	10.05	10.55	0.395	0.415
R	0.25	0.75	0.010	0.029

SECTION 15
ORDERING INFORMATION

15.1 Introduction

This section contains instructions for ordering the MC68HC805P18.

15.2 MC Order Numbers

Table 15-1 shows the MC order numbers for the available package types.

Table 15-1. MC Order Numbers

MC Order Number	Operating Temperature Range
MC68HC805P18P (Standard)	0 °C to 70 °C
MC68HC805P18DW (Standard)	0 °C to 70 °C
MC68HC805P18CP (Extended)	−40 °C to +85 °C
MC68HC805P18CDW (Extended)	−40 °C to +85 °C
MC68HC805P18MP (Automotive)	−40 °C to +125 °C
MC68HC805P18MDW (Automotive)	−40 °C to +125 °C

P = Plastic Dual In-Line Package
DW = Small Outline (Wide Body) Package

**APPENDIX A
EMULATION**

This appendix discusses the functional differences between the P-series devices.
The MC68HC805P18 can be used to emulate the following devices:

MC68HC05P1A	MC68HC05P7
MC68HC05P2	MC68HC05P7A
MC68HC05P3	MC68HC05P8
MC68HC05P4	MC68HC05P9
MC68HC05P4A	MC68HC705P9
MC68HC05P6	MC68HC05P10
MC68HC705P6	MC68HC05P18

These functional differences will be summarized in:
   Table A-1. Elements of Memory
   Table A-2. Memory Breakdown by Types
   Table A-3. P-Series Features
   Table A-4. Mask Options

Table A-1. Elements of Memory

Device	RAM	User ROM	EPROM	EEPROM	User EEPROM	ROM Security
P1A	128 b 0080-00FF	R 2320 b 0020-004F 0100-08FF*	N	N	N	N
P2	96 b 00A0-00FF	R 3088 b 0020-004F 1300-1EFF	N	N	N	N
P3	128 b 0080-00FF	R 3072 b 0020-004F 0300-0EFF	N	128 b 0100-017F	N	N
705P3	128 b 0080-00FF	N	3072 b 0020-004F 0300-0EFF	128 b 0100-017F	N	N
P4/P4A	176 b 0050-00FF	R 4160 b 0020-004F 0100-10FF	N	N	N	N/Y
P6	176 b 0050-00FF	R 4672 b 0020-004F 0100-10FF*	N	N	N	N
705P6	176 b 0050-00FF	N	4672 b 0020-004F 0100-12FF*	N	N	N
P7/P7A	128 b 0080-00FF	R 2112 b 0020-004F 0100-08FF*	N	N	N	N/Y
P8	112 b 0090-00FF	R 2064 b 1680-1E7F	N	32 b 0030-004F	N	N
P9/P9A	128 b 0080-00FF	R 2112 b 0020-004F 0100-08FF	N	N	N	N/Y
705P9	128 b 0080-00FF	N	2112 b 0020-004F 0100-08FF*	N	N	N
P10	128 b 0080-00FF	R 4160 b 0020-004F 0100-10FF	N	N	N	N
P18	192 b 0050-010F	R 8064 b 0020-004F 1FC0-3EFF	N	128 b 0140-01BF	N	N
805P18	192 b 0050-010F	N	N	128 b 0140-01BF	8064 b 0020-004F 1FC0-3EFF	Y

Table A-2. Memory Breakdown by Types

Range	P1a	P2	P3/705P3	P4/P4a	P6	705 P6	P7/P7a	P8	P9/705P9	P10	P18	805 P18
0020–004F	ROM	ROM	ROM/E	ROM	ROM	ROM	ROM		ROM/E	ROM	ROM	UEE
0030–004F								EE				
0050–007F				RAM	RAM	RAM					RAM	RAM
0080–008F	RAM		RAM	RAM	RAM	RAM	RAM		RAM	RAM	RAM	RAM
0090–009F	RAM		RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM
00A0–00FF	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM
0100–010F	ROM		EE	ROM	ROM	E	ROM		ROME	ROM	RAM	RAM
0110–013F	ROM		EE	ROM	ROM	E	ROM		ROME	ROM		
0140–017F	ROM		EE	ROM	ROM	E	ROM		ROME	ROM	EE	EE
0180–01BF	ROM			ROM	ROM	E	ROM		ROME	ROM	EE	EE
01C0–02D1	ROM			ROM	ROM	E	ROM		ROME	ROM		ROM
02D2–02FF	ROM			ROM	ROM	E	ROM		ROME	ROM		
0300–08FF	ROM		ROME	ROM	ROM	E	ROM		ROME	ROM		
0900–0EFF			ROME	RON	ROM	E				ROM		
0F00–0FEF			ROM	ROM	ROM	E				ROM		
0FF0–10FF				ROM	ROM	E				ROM		
1100–12FF					ROM	E						
1300–167F		ROM										
1680–1EFF		ROM						ROM				
1F00		ROM		ROM			ROM	ROM	ROM	ROM		
1F01–1FBF		ROM		ROM	ROM	ROM	ROM	ROM	ROM	ROM		
1FC0–1FEF		ROM		ROM	ROM	ROM	ROM	ROM	ROM	ROM	ROM	UEE
1FF0–3EFF											ROM	UEE
3F00–3F01												UEE
3F02–3FEF												ROM

NOTE: I/O registers are common to all parts so they are not included in the table. There are an additional 16 bytes of user vectors in the memory map for each device.

E = EPROM
EE = EEPROM
PEE = User EEPROM

Table A-3. P-Series Features

Devices	Mask Option	MOR	A/D	LVR	High Current
P1A	Y	N	N	N	Y
P2	Y	N	N	N	N
P3/705P3	Y/N	N/Y	N	N	N
P4/P4A	Y	N	N	N	N/Y
P6/705P6	Y/N	N/Y	Y	N	N
P7/P7A	Y	N	N	N	N/Y
P8	Y	N	Y	N	N
P9/705P9/P9A	Y/N/Y	N/Y/N	Y	N	N/N/Y
P10	Y	N	N	N	N
P18/805P18	Y/N	N/Y	Y	Y	Y

Table A-4. Mask Options

Devices	XTAL/RC	SIOP Clock Rate	SIOP MSB/LSB	Port A PU/INT	STOP to HALT
P1A	Y	N	N	Y	Y
P2	Y	N	N	N	N
P3	N	N	N	N	N
P4/P4A	Y	N	Y	N/Y	N/Y
P6	Y	Y	Y	N	Y
P7/P7A	Y	N	Y	N/Y	N/Y
P8	N	N	N	N	N
P9/P9A	N	N	Y	N/Y	N/Y
P10	Y	N	Y	Y	N
P18	N	Y[*]	Y	Y	Y

* The MC68HC05P18 and MC68HC805P18 have selectable clock rates that are four times as fast as the MC68HC05P6 selectable rates.

		Bit 7	6	5	4	3	2	1	Bit 0
$0F	Read:	0	1	0	0	1	0	OPTCOP	OPTIRQ
	Write:								
	Reset:				Unaffected by reset				

Figure A-1. MC68HC705P3 Mask Option Register

		Bit 7	6	5	4	3	2	1	Bit 0
$900	Read:	—	RC	SWAIT	SPR1	SPR0	LSBF	IRQ	COP
	Write:								
	Reset:				Unaffected by reset				

Figure A-2. MC68HC705P6 Mask Option Register

		Bit 7	6	5	4	3	2	1	Bit 0
$900	Read:	—	—	—	—	—	SIOP	IRQ	COP
	Write:								
	Reset:				Unaffected by reset				

Figure A-3. MC68HC705P9 Mask Option Register

		Bit 7	6	5	4	3	2	1	Bit 0
MOR1	Read:	CLKOUT	LVRE	SWAIT	SPR1	SPR0	LSBF	LEVIRQ	COPEN
	Write:								
	Reset:				Unaffected by reset				

		Bit 7	6	5	4	3	2	1	Bit 0
MOR2	Read:	PA7PU	PA6PU	PA5PU	PA4PU	PA3PU	PA2PU	PA1PU	PA0PU
	Write:								
	Reset:				Unaffected by reset				

Figure A-4. MC68HC805P18 Mask Option Register

How to reach us:

USA/EUROPE: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036. 1-800-441-2447

MFAX: RMFAX0@email.sps.mot.com – TOUCHTONE (602) 244-6609

INTERNET: http://Design-NET.com

JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC, Toshikatsu Otsuki, 6F Seibu-Butsuryu-Center, 3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 03-3521-8315

HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park, 51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298

MOTOROLA