MC68HC05P1A
MC68HCL05P1A
MC68HSC05P1A

General Release Specification

February 25, 1997

CSIC MCU Design Center
Austin, Texas

List of Sections

List of Sections

Table of Contents

Section 1. General Description

Section 2. Memory

Table of Contents

Section 3. CPU Core

Section 4. Interrupts

Section 5. Resets

Section 6. Operating Modes

Section 7. Input/Output Ports

Section 8. 16-Bit Timer

Section 9. Instruction Set

Table of Contents

Section 10. Electrical Specifications

Section 11. Mechanical Specifications

Section 12. Ordering Information

Appendix A. MC68HCL05P1A

Appendix B. MC68HSC05P1A

Table of Contents

List of Figures

List of Figures

List of Tables

List of Tables

Section 1. General Description

1.1 Contents

1.2 Introduction

The Freescale MC68HC05P1A microcontroller unit (MCU) is pin compatible with the MC68HC05P1 with port and interrupt enhancements available. This device is available in a 28-pin dual in-line package (DIP) or a small outline integrated circuit (SOIC) package. A functional block diagram of the MC68HC05P1A is shown in **Figure 1-1**.

1.3 Features

Features of the MC68HC05P1A include:

- Low-Cost, M68HC05 Core
- 28-Pin Dual In-Line Package (DIP) or Small Outline Integrated Circuit (SOIC) Package
- On-Chip Crystal/Ceramic Resonator or RC Oscillator (Mask Option)
- 2320 Bytes of User Read-Only Memory (ROM) Including:
  - 48 Bytes of Page-Zero ROM
  - 16 Bytes of User Vectors
- 128 Bytes of On-Chip Random Access Memory (RAM)
- 16-Bit Timer with Output Compare and Input Capture
- Edge/Level-Sensitive Interrupt or Edge-Sensitive Only (Mask Option)
- Computer Operating Properly (COP) Watchdog Timer
- 20 Bidirectional Input/Output (I/O) Lines and One Input-Only Line Including:
  - Individual Mask Selectable Pullups/Interrupts on Port A Pins
  - High Current Sink and Source on Two I/O Pins (PC0 and PC1)
- Single-Chip Mode
- Power-Saving Stop and Wait Modes
- Stop Conversion to Halt Mode (Mask Option)

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

NOTE: *Any reference to a 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 10. Electrical Specifications****.*

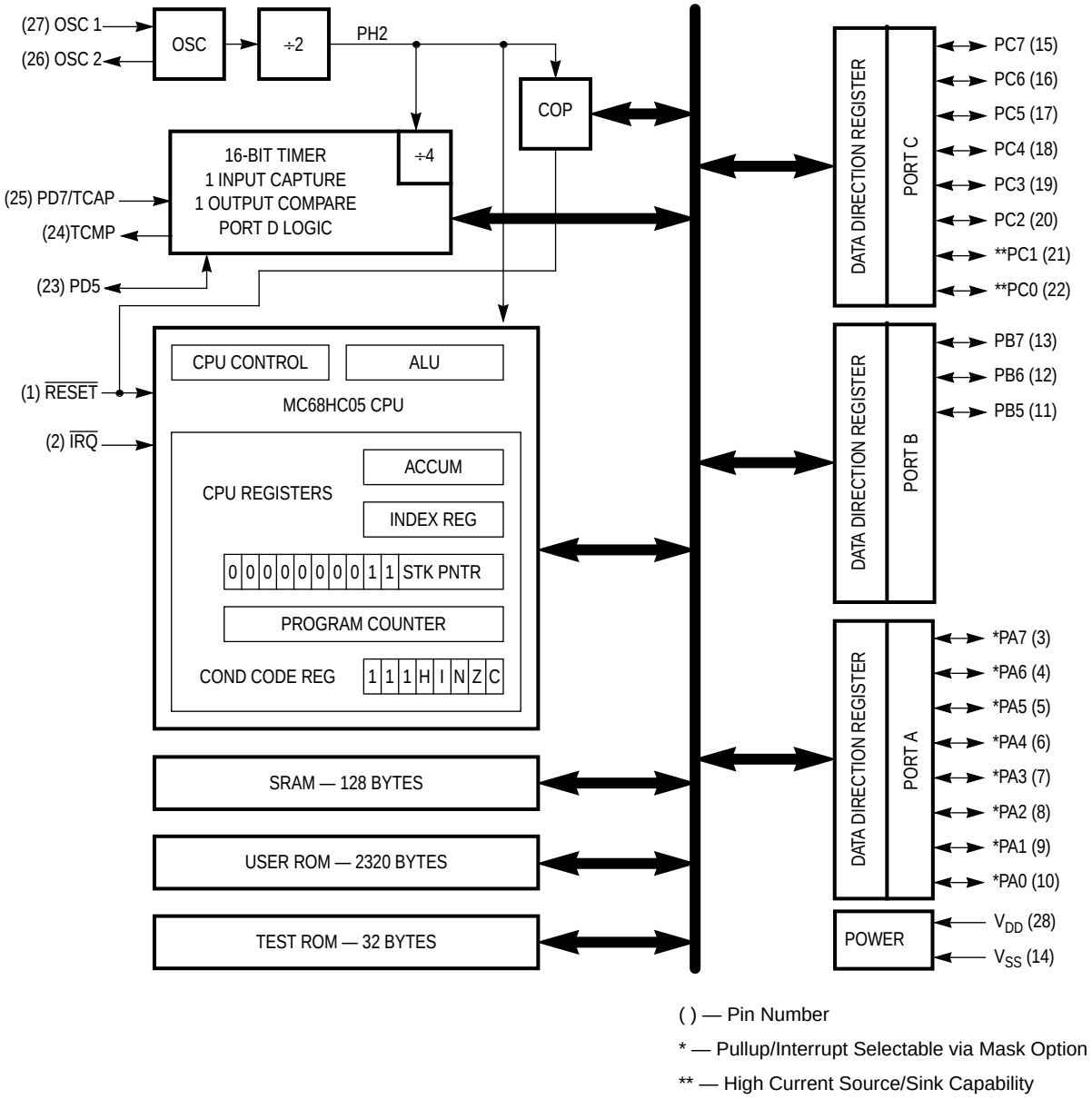


Figure 1-1. MC68HC05P1A Block Diagram

1.4 Mask Options

The MC68HC05P1A has 12 mask options. The default state of these mask options and their alternate states are:

1. $\overline{IRQ}$ is Edge- and Level-Sensitive — Option for Edge-Sensitive Only
2. Crystal/Ceramic Resonator Oscillator Mode — Option for Resistor/Capacitor (RC) Mode
3. COP Watchdog Timer Enabled — Option to Disable
4. Stop Instruction Enabled — Option to Convert to Halt
5. Eight (8) Port A Pullups/Interrupts Disabled — Option to Individually Enable

1.5 Functional Pin Description

The following paragraphs describe the functionality of each pin on the MC68HC05P1A package. The device also is available with an alternate pinout where V_{DD} and V_{SS} are adjacent to reduce radio frequency (RF) emissions. This also improves the ability to decouple V_{DD} and V_{SS}, which may provide some benefit for conducted RF emissions as well. The pinouts are shown in **Figure 1-2** and **Figure 1-3**. They are compatible with the MC68HC05P1 microcontroller unit (MCU).

1.5.1 V_{DD} and V_{SS}

Power is supplied to the MCU through V_{DD} and V_{SS}. V_{DD} is connected to a regulated +5 volt 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.

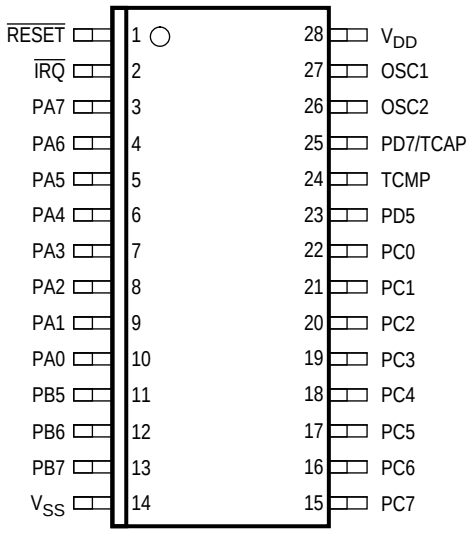


Figure 1-2. Single-Chip Pinout

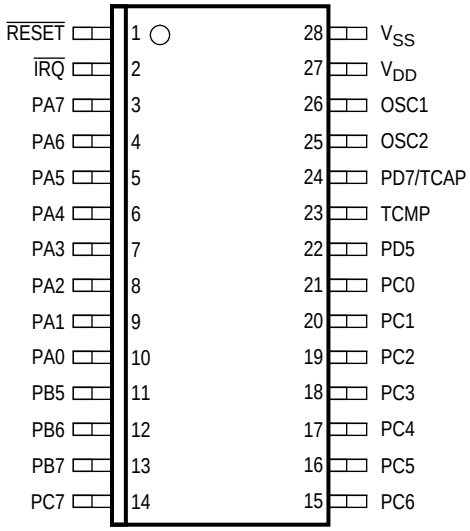
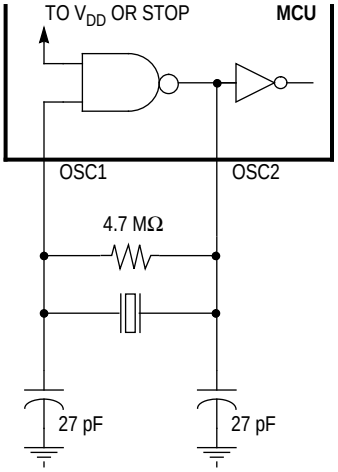


Figure 1-3. Low Noise Single-Chip Pinout

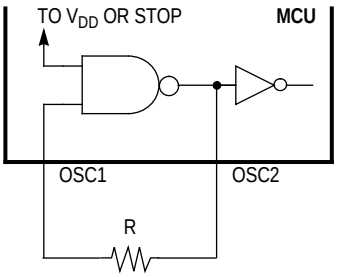
1.5.2 OSC1 and OSC2

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

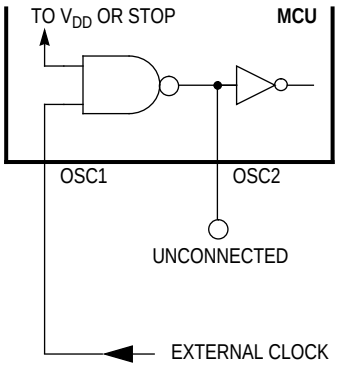
1. A crystal, as shown in **Figure 1-4 (a)**.
2. A ceramic resonator, as shown in **Figure 1-4 (a)**.
3. An external resistor, as shown in **Figure 1-4 (b)**.
4. An external clock signal, as shown in **Figure 1-4 (c)**.



(a) Crystal or Ceramic Resonator Connections



(b) RC Oscillator Connections



(c) External Clock Source Connections

Figure 1-4. Oscillator Connections

NOTE: *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}.*

1.5.2.1 Crystal

The circuit in **Figure 1-4 (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 closely as possible to the pins for startup stabilization and to minimize output distortion.

1.5.2.2 Ceramic Resonator

In cost-sensitive applications, use a ceramic resonator in place of a crystal. Use the circuit in **Figure 1-4 (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 closely as possible to the pins for startup stabilization and to minimize output distortion.

1.5.2.3 RC Oscillator

The lowest cost oscillator uses the RC mask option and an external resistor. With this option, a resistor is connected to the oscillator pins, as shown in **Figure 1-4 (b)**. The relationship between R and f_{op} is shown in **Figure 1-5**. Consult the factory for tolerance limits and design specifications.

1.5.2.4 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 1-4 (c)**.

General Description

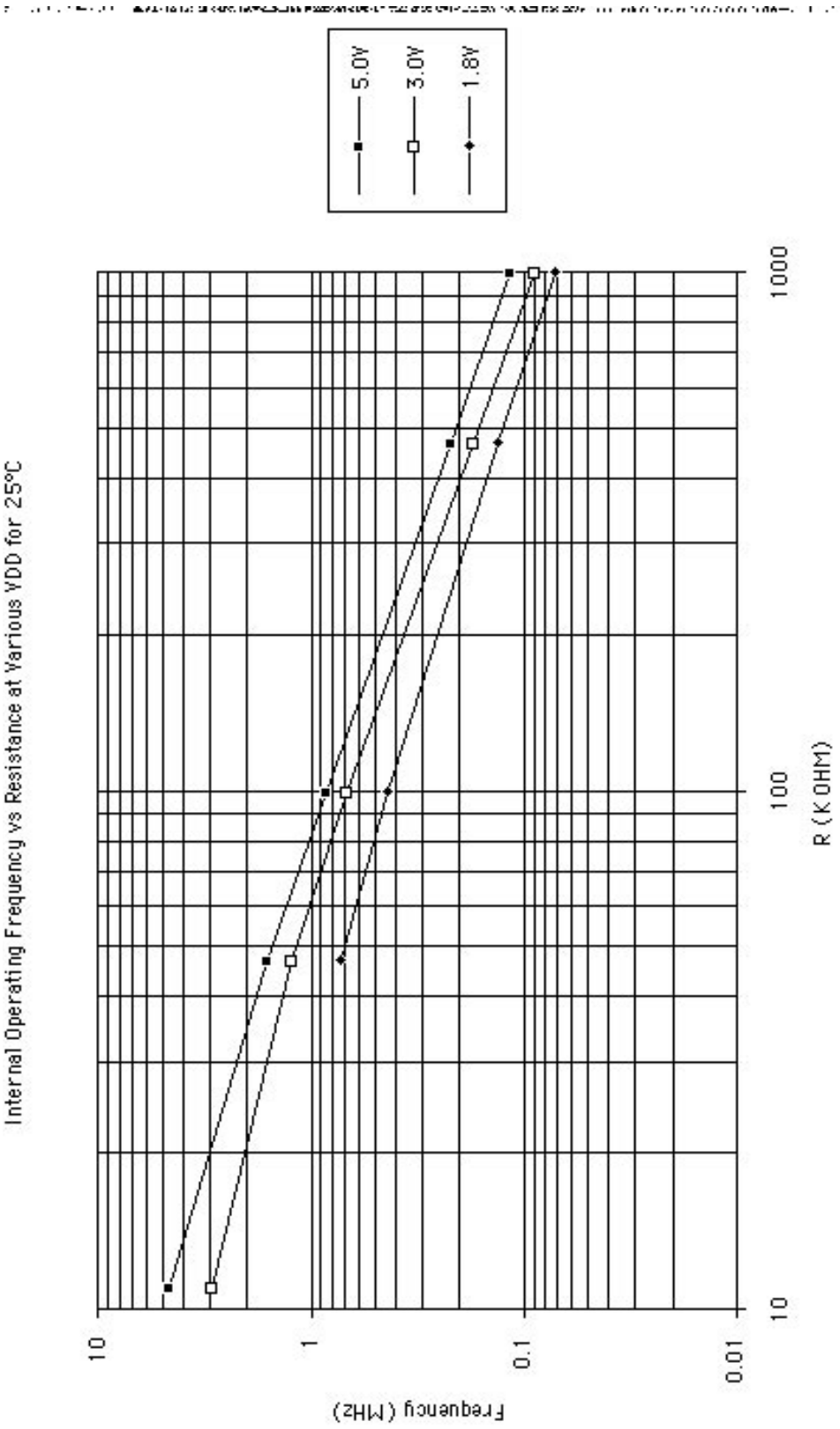


Figure 1-5. Typical Frequency versus Resistance for RC Oscillator Mask Option

1.6 $\overline{\text{RESET}}$

Driving this input low will reset the MCU to a known startup state. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger to improve its noise immunity. Refer to **Section 5. Resets**.

1.6.1 PA0–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. Eight individual mask options can be chosen to enable pullups and interrupts (active low) on each port A pin. Refer to **Section 4. Interrupts** and **Section 7. Input/Output Ports**.

1.6.2 PB5, PB6, and PB7

These three I/O pins comprise port B. The state of any pin is software programmable and all port B lines are configured as inputs during power-on or reset. Refer to **Section 7. Input/Output Ports**.

1.6.3 PC0–PC7

These eight I/O pins comprise port C. The state of any pin is software programmable and all port C lines are configured as inputs during power-on or reset. PC0 and PC1 are capable of sourcing and sinking more current than a typical I/O pin. Refer to **Section 7. Input/Output Ports** and **Section 10. Electrical Specifications**.

1.6.4 PD5 and PD7/TCAP

These two pins comprise port D and are shared with the 16-bit timer subsystem. The state of PD5 is software programmable and is configured as an input during power-on or reset. PD7 is always an input. It may be read at any time, regardless of the mode of operation of the 16-bit timer. Refer to **Section 7. Input/Output Ports** and **Section 8. 16-Bit Timer**.

1.6.5 TCMP

This pin is the output from the 16-bit timer's output compare function. It is low after reset. Refer to **Section 8. 16-Bit Timer**.

1.6.6 $\overline{IRQ}$ (Maskable Interrupt Request)

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 $\overline{IRQ}$ interrupt request. When $\overline{IRQ}$ 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 mask option selected, the $\overline{IRQ}$ pin will trigger this interrupt on either a negative going edge at the $\overline{IRQ}$ pin and/or while the $\overline{IRQ}$ pin is held in the low state. In either case, the $\overline{IRQ}$ pin must be held low for at least one t_{ILIH} time period. The $\overline{IRQ}$ input requires an external resistor connected to V_{DD} for wired-OR operation. If the $\overline{IRQ}$ pin is not used, it must be tied to the V_{DD} supply. The $\overline{IRQ}$ pin contains an internal Schmitt trigger as part of its input circuitry to improve noise immunity. Refer to **Section 4. Interrupts**.

NOTE: *Each of the port A I/O pins may be connected as an OR function with the IRQ interrupt function by a mask option. This capability allows keyboard scan applications where the transitions or levels on the I/O pins will behave the same as the $\overline{IRQ}$ pin. The edge or level sensitivity selected by a separate mask option for the $\overline{IRQ}$ pin also applies to the I/O pins OR'ed to create the IRQ signal.*

NOTE: *If the voltage level applied to the $\overline{IRQ}$ pin exceeds 1.5 V_{DD}, it may affect the MCU's mode of operation. See **Section 6. Operating Modes**.*

Section 2. Memory

2.1 Introduction

2.2 Introduction

The MC68HC05P1A utilizes 13 address lines to access an internal memory space covering 8 Kbytes. This memory space is divided into I/O, RAM, and ROM areas.

2.3 Single-Chip Mode Memory Map

When the MC68HC05P1A is in the single-chip mode, the 32 bytes of I/O, 128 bytes of RAM, 2256 bytes of user ROM, 48 bytes of user page zero ROM, 32 bytes of test ROM, and 16 bytes of user vectors ROM are all active, as shown in **Figure 2-1**.

2.4 Input/Output (I/O) and Control Registers

Figure 2-2 and **Figure 2-3** 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.

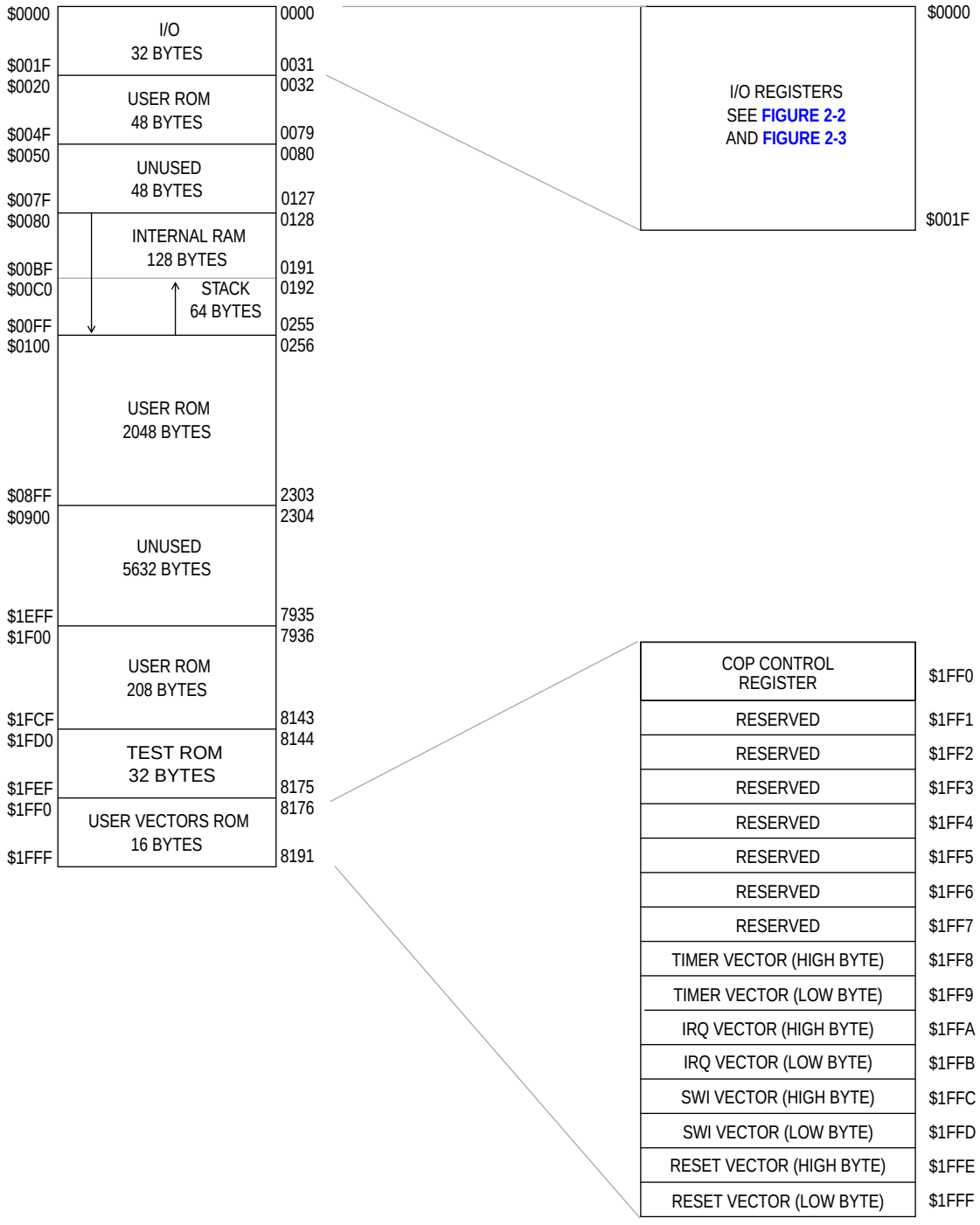


Figure 2-1. Single-Chip Mode Memory Map

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data (PORTA)	Read:	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
		Write:								
$0001	Port B Data (PORTB)	Read:	PB7	PB6	PB5	0	0	0	0	0
		Write:								
$0002	Port C Data (PORTC)	Read:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
		Write:								
$0003	Port D Data (PORTD)	Read:	PD7	0	PD5	1	0	0	0	0
		Write:								
$0004	Port A Data Direction (DDRA)	Read:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
$0005	Port B Data Direction (DDRB)	Read:	DDRB7	DDRB6	DDRB5	1	1	1	1	1
		Write:				U	U	U	U	U
$0006	Port C Data Direction (DDRC)	Read:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Write:								
$0007	Port D Data Direction (DDRD)	Read:	0	0	DDRD5	0	0	0	0	0
		Write:								
$0008	Unimplemented	Read:								
		Write:								
$0009	Unimplemented	Read:								
		Write:								
$000A	Unimplemented	Read:								
		Write:								
$000B	Unimplemented	Read:								
		Write:								
$000C	Unimplemented	Read:								
		Write:								

= Unimplemented

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

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000D	Unimplemented	Read:								
		Write:								
$000E	Unimplemented	Read:								
		Write:								
$000F	Unimplemented	Read:								
		Write:								

= Unimplemented

Figure 2-2. I/O and Control Registers $0000–$000F (Continued)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$010D	Unimplemented	Read:								
		Write:								
$0011	Unimplemented	Read:								
		Write:								
$0012	Timer Control Register (TCR)	Read:	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
		Write:								
$0013	Timer Status Register (TSR)	Read:	ICF	OCF	TOF	0	0	0	0	0
		Write:								
$0014	Input Capture MSB (ICRH)	Read:	ICRH7	ICRH6	ICRH5	ICRH4	ICRH3	ICRH2	ICRH1	ICRH0
		Write:								
$0015	Input Capture LSB (ICRL)	Read:	ICRL7	ICRL6	ICRL5	ICRL4	ICRL3	ICRL2	ICRL1	ICRL0
		Write:								
$0016	Output Compare MSB (OCRH)	Read:	OCRH7	OCRH6	OCRH5	OCRH4	OCRH3	OCRH2	OCRH1	OCRH0
		Write:								

= Unimplemented R = Reserved

Figure 2-3. I/O and Control Registers $010D–$001F

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0017	Output Compare MSB (OCRL)	Read:	OCRL7	OCRL6	OCRL5	OCRL4	OCRL3	OCRL2	OCRL1	OCRL0
		Write:								
$0018	Timer MSB (TIMRH)	Read:	TMRH7	TMRH6	TMRH5	TMRH4	TMRH3	TMRH2	TMRH1	TMRH0
		Write:								
$0019	Timer LSB (TMRL)	Read:	TMRL7	TMRL6	TMRL5	TMRL4	TMRL3	TMRL2	TMRL1	TMRL0
		Write:								
$001A	Alternate Counter MSB (ACRH)	Read:	ACRH7	ACRH6	ACRH5	ACRH4	ACRH3	ACRH2	ACRH1	ACRH0
		Write:								
$001B	Alternate Counter LSB (ACRL)	Read:	ACRL7	ACRL6	ACRL5	ACRL4	ACRL3	ACRL2	ACRL1	ACRL0
		Write:								
$001C	Unimplemented	Read:								
		Write:								
$001D	Unimplemented	Read:								
		Write:								
$001E	Unimplemented	Read:								
		Write:								
$001F	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:								

= Unimplemented R = Reserved

Figure 2-3. I/O and Control Registers $010D–$001F (Continued)

2.5 Random-Access Memory (RAM)

The user RAM consists of 128 bytes (including the stack) at locations $0080–$00FF. 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.*

2.6 Read-Only Memory (ROM)

There are 2256 bytes of user ROM at locations $0100–$08FF and $1F00–$1FCF, with 48 bytes in user page zero locations $0020–$004F, and 16 additional bytes for user vectors at locations $1FF0–$1FFF. The test ROM and test ROM vectors are at locations $1FD0–$1FEF.

Section 3. CPU Core

3.1 Contents

3.2 Introduction

This section describes the registers of the M68HC05 central processor unit (CPU). The stop and wait modes, initiated by software instructions, are also described here.

3.3 CPU Registers

The CPU contains the following registers:

- Accumulator (A)
- Index register (X)
- Stack pointer (SP)
- Program counter (PC)
- Condition code register (CCR)

These registers are hard-wired within the CPU and are not part of the memory map. **Figure 3-1** is a block diagram of the M68HC05 CPU.

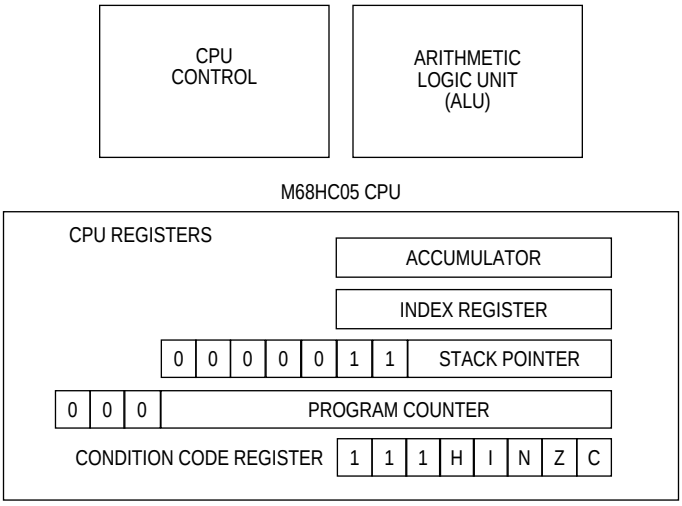


Figure 3-1. CPU Block Diagram

Figure 3-2 shows the five CPU registers.

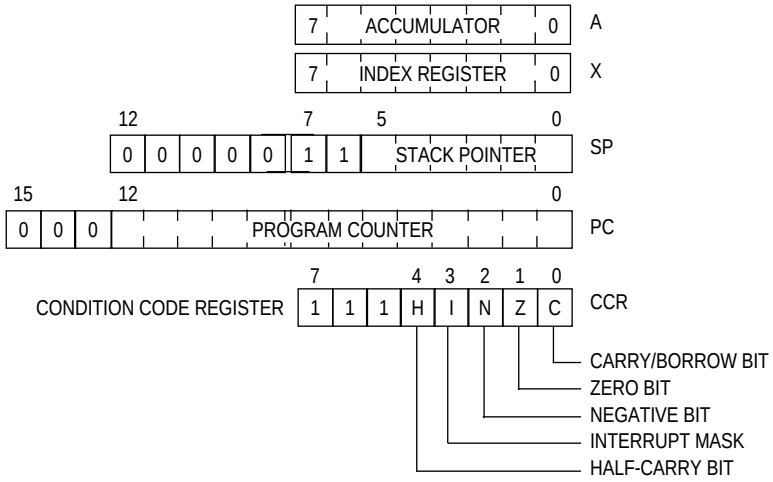
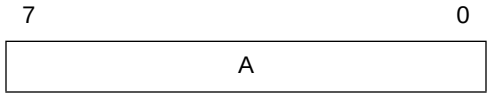


Figure 3-2. Programming Model

3.3.1 Accumulator

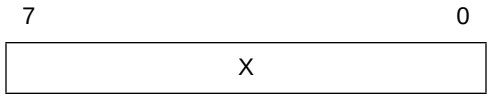
The accumulator (A) is a general-purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.



3.3.2 Index Register

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

In indexed addressing with no offset, the index register contains the low byte of the operand address, and the high byte is assumed to be $00. In indexed addressing with an 8-bit offset, the CPU finds the operand address by adding the index register contents to an 8-bit immediate value. In indexed addressing with a 16-bit offset, the CPU finds the operand address by adding the index register contents to a 16-bit immediate value.

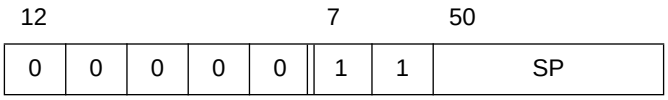


3.3.3 Stack Pointer

The stack pointer (SP) 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.

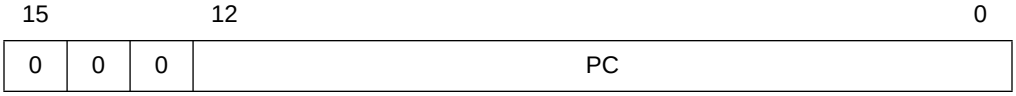
When accessing memory, the eight most significant bits (MSB) are permanently set to 00000011. These eight bits are appended to the six least significant bits (LSB) to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around

and loses the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.



3.3.4 Program Counter

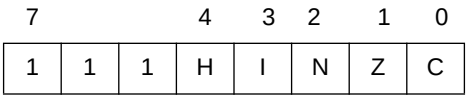
The program counter (PC) is a 13-bit register that contains the address of the next byte to be fetched. Because addresses are often 16-bit values, the program counter may be thought of as having three additional upper bits that are always zeros.



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

3.3.5 Condition Code Register

The condition code register (CCR) is a 5-bit register in which the H, N, Z, and C bits are used to indicate the results of the instruction just executed, and the I bit is used to enable or disable interrupts. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Consider the condition code register as having three additional upper bits that are always ones.



H — Half Carry

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

I — Interrupt

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 I bit is cleared.

N — Negative

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

Z — Zero

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

C — Carry/Borrow

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

Section 4. Interrupts

4.1 Contents

4.2 Introduction

The MCU can be interrupted by:

1. Non-maskable Software Interrupt Instruction (SWI)
2. External Asynchronous Interrupt ($\overline{IRQ}$)
3. Optional External Asynchronous Interrupt on Each Port A Pin ($\overline{IRQ}$, Enabled by Pullup Mask Option)
4. Input Capture Interrupt (TIMER)
5. Output Compare Interrupt (TIMER)
6. Timer Overflow Interrupt (TIMER)

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 $\overline{RESET}$, 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 by the I bit being clear in the condition code register (CCR) and the corresponding interrupt enable bit being 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 state of the I bit.

When an interrupt is 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 $1FF0–$1FFF. 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 executed when the interrupt took place. **Figure 4-1** shows the event sequence that occurs during interrupt processing.

Table 4-1. Vector Addresses for Interrupts and Reset

Register	Flag Name	Enable Bit	Interrupt	CPU Interrupt	Vector Address
N/A	N/A	N/A	Reset	RESET	$1FFE–$1FFF
N/A	N/A	N/A	Software	SWI	$1FFC–$1FFD
N/A	N/A	N/A	External Interrupt	IRQ	$1FFA–$1FFB
TSR	ICF	ICIE	Timer Input Capture	TIMER	$1FF8–$1FF9
TSR	OCF	OCIE	Timer Output Compare	TIMER	$1FF8–$1FF9
TSR	TOF	TOIE	Timer Overflow	TIMER	$1FF8–$1FF9
N/A	N/A	N/A	Unimplemented	N/A	$1FF6-$1FF7
N/A	N/A	N/A	Unimplemented	N/A	$1FF4-$1FF5
N/A	N/A	N/A	Unimplemented	N/A	$1FF2-$1FF3
N/A	N/A	N/A	Unimplemented	N/A	$1FF0-$1FF1

4.3 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 $1FFE and $1FFF. The I bit in the condition code register is also set. The MCU is configured to a known state during a reset, as described in **Section 5. Resets**.

4.4 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 $1FFC and $1FFD.

4.5 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. The hardware interrupts are explained in the following sections.

4.5.1 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-flops 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 mask 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. The interrupt service routine address is specified by the contents of memory locations $1FFA and $1FFB. A block diagram of the IRQ function is shown in **Figure 4-2**.

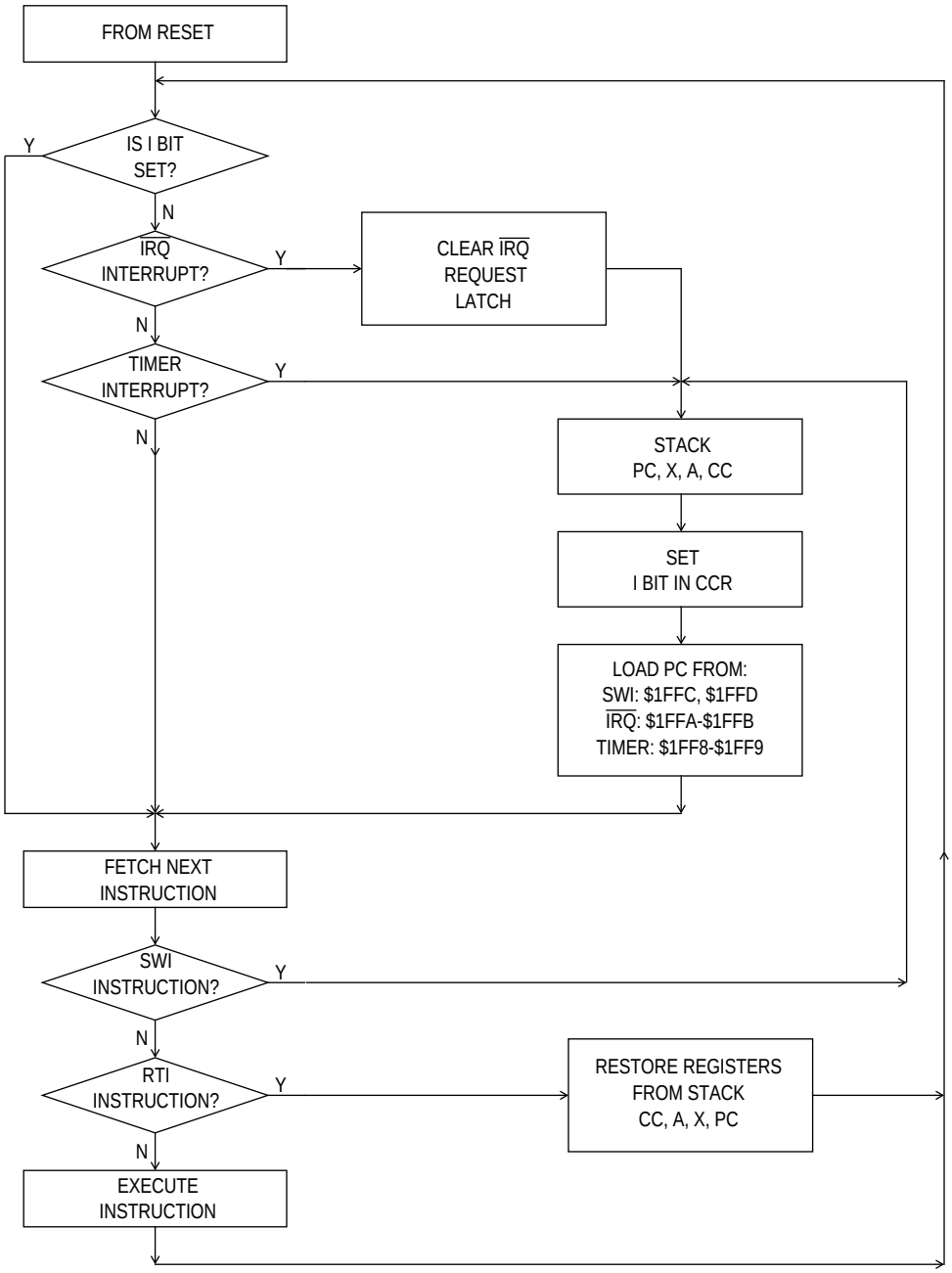


Figure 4-1. Interrupt Processing Flowchart

NOTE: *The internal interrupt latch is cleared nine PH2 clock cycles after the interrupt is recognized (after location $1FFA is read). Therefore, another external interrupt pulse can be latched during the IRQ service routine.*

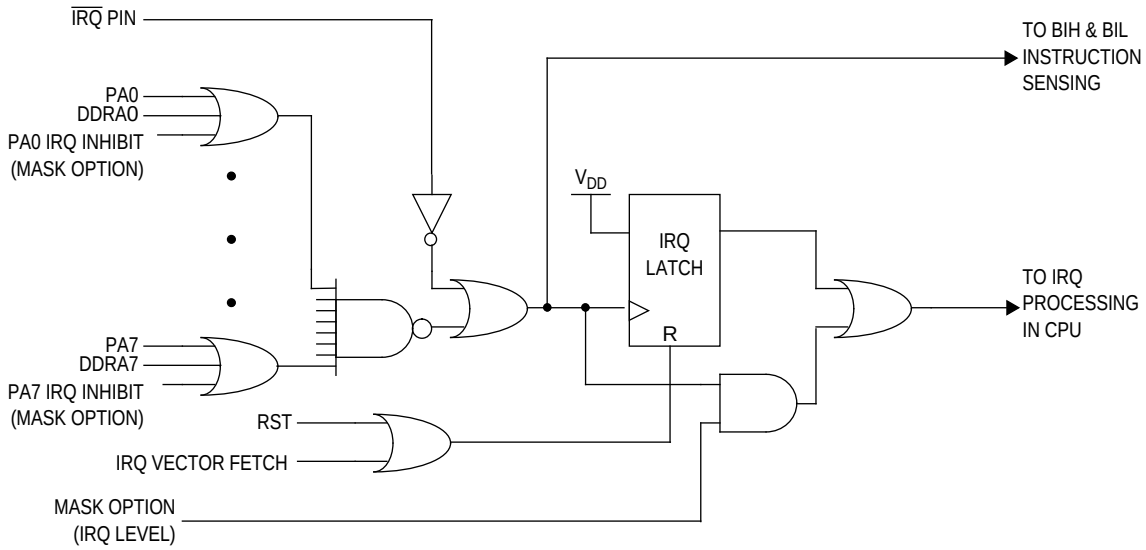


Figure 4-2. IRQ Function Block Diagram

NOTE: *When the edge- and level-sensitive mask 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 to avoid the processor re-entering the IRQ service routine.*

The $\overline{IRQ}$ pin is one source of an IRQ interrupt, and a mask option can also enable the port A pins (PA0–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.

Any enabled IRQ interrupt source sets the IRQ latch on the falling edge of the $\overline{IRQ}$ 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 $\overline{IRQ}$ 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 $\overline{IRQ}$ 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 on 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 $1FFA and $1FFB. The IRQ latch is automatically cleared by entering the interrupt service routine.

4.5.2 Optional External Interrupts (PA0–PA7)

The IRQ interrupt can be triggered by the inputs on the PA0–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–PA7 interrupt pins and the $\overline{\text{IRQ}}$ pin. The BIH and BIL instructions do not exclusively test the state of the $\overline{\text{IRQ}}$ pin.*

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

4.5.3 Input Capture Interrupt

The input capture interrupt is generated by the 16-bit timer as described in **Section 8. 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 for the input capture interrupt to

be enabled. The interrupt service routine address is specified by the contents of memory locations $1FF8 and $1FF9.

4.5.4 Output Compare Interrupt

The output compare interrupt is generated by the 16-bit timer as described in **Section 8. 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 for the output compare interrupt to be enabled. The interrupt service routine address is specified by the contents of memory locations $1FF8 and $1FF9.

4.5.5 Timer Overflow Interrupt

The timer overflow interrupt is generated by the 16-bit timer as described in **Section 8. 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 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 $1FF8 and $1FF9.

Section 5. Resets

5.1 Contents

5.2 Introduction

The MCU can be reset from three sources: one external input and two internal reset conditions. The $\overline{\text{RESET}}$ 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.

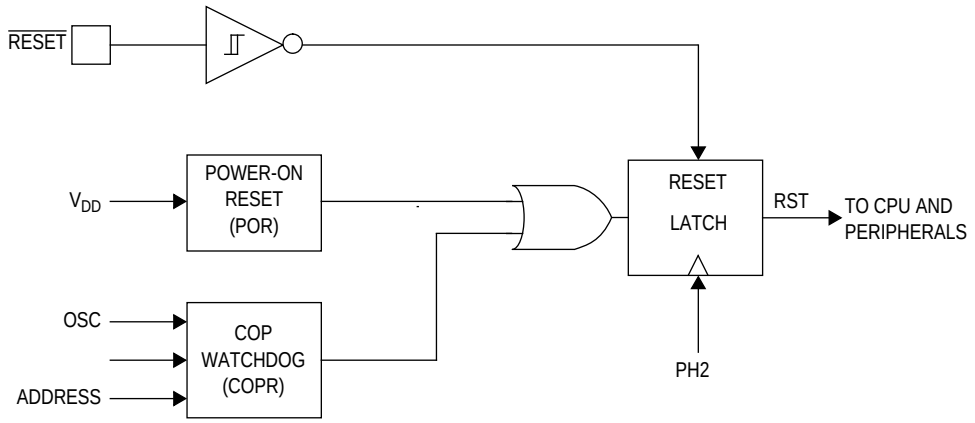


Figure 5-1. Reset Block Diagram

5.3 External Reset ($\overline{\text{RESET}}$)

The reset input is the only external reset and is connected to an internal Schmitt trigger. The external reset occurs whenever the reset input is driven below the lower threshold and remains in reset until the $\overline{\text{RESET}}$ pin rises above the upper threshold. The upper and lower thresholds are given in **Section 10. Electrical Specifications**.

5.4 Internal Resets

The two internally generated resets are the initial power-on reset (POR) function and the COP watchdog timer function.

5.4.1 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.

5.4.2 Computer Operating Properly (COP) Reset

When the COP watchdog timer is enabled (by mask option), 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 timeout period of 65.5 ms when a 4-MHz oscillator is used. The COP watchdog counter is cleared by writing a logic 0 to bit zero at location $1FF0.

The COP watchdog timer can be disabled by mask option or by applying $2 \times V_{DD}$ to the $\overline{\text{IRQ}}$ pin. When the $\overline{\text{IRQ}}$ pin is returned to its normal operating voltage range (between V_{SS} and V_{DD}), the COP watchdog timer output will be restored if the COP mask option is enabled.

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 returns the MSB of the unimplemented user interrupt vector. Writing a logic 0 to this location clears the COP watchdog timer.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$1FF0	Unimplemented Vector and COP Watchdog Timer	Read:	R	R	R	R	R	R	R	R
		Write:								COPC

R = Reserved

Figure 5-2. COP Watchdog Timer Location

Section 6. Operating Modes

6.1 Contents

6.2 Introduction

The MC68HC05P1A uses single-chip mode. The conditions required to enter this mode are shown in **Table 6-1**. The mode of operation is determined by the voltages on the $\overline{IRQ}$ and PD7/TCAP pins on the rising edge of the external $\overline{RESET}$ pin.

Table 6-1. Operating Mode Conditions After Reset

$\overline{RESET}$ Pin	$\overline{IRQ}$ Pin	PD7/TCAP	Mode
	V_{SS} to V_{DD}	V_{SS} to V_{DD}	Single-Chip

$V_{TST} = 2 \times V_{DD}$

The mode of operation is also determined whenever the internal COP watchdog timer resets the MCU. When the COP timer expires, the voltage applied to the $\overline{IRQ}$ pin affects the mode of operation, while the voltage applied to PD7/TCAP is ignored if the voltage at the $\overline{IRQ}$ pin

exceeds V_{TST}. In this case, the voltage applied to PD7/TCAP during the last rising edge on $\overline{\text{RESET}}$ is stored in a latch and used to determine the mode of operation when the COP watchdog timer resets the MCU.

6.3 Single-Chip Mode

The single-chip 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. Single-chip mode is entered on the rising edge of $\overline{\text{RESET}}$ if the $\overline{\text{IRQ}}$ pin is within the normal operating voltage range. The pinout for the single-chip mode is shown in **Figure 1-2 . Single-Chip Pinout**.

In the single-chip mode, two 8-bit I/O ports, one 3-bit I/O port, and a 1-bit I/O port are shared with the 16-bit timer subsystem. The 16-bit timer subsystem also has one input-only pin and one output-only pin.

6.4 Low-Power Modes

The MC68HC05P1A is capable of running in a low-power mode in each of its configurations. The WAIT and STOP instructions provide two 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. The stop conversion mask option 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 6-1**.

6.5 STOP Instruction

The STOP instruction can result in one of two modes of operation, depending on the stop conversion mask option. If the stop conversion is not chosen, the STOP instruction will behave like a normal STOP instruction in the M68HC05 Family and place the MCU in the stop mode. If the stop conversion is chosen, the STOP instruction will behave like a WAIT instruction (with the exception of a variable delay at startup) and place the MCU in halt mode.

6.5.0.1 Stop Mode

Execution of the STOP instruction without conversion to halt places the MCU in its lowest-power consumption mode. In stop mode, the internal oscillator is turned off, halting *all* internal processing, including the COP watchdog timer. Execution of the STOP instruction automatically clears the I bit in the condition code register so that the external interrupt is enabled. All other registers and memory remain unaltered. All input/output lines remain unchanged.

The MCU can be brought out of stop mode only by an external interrupt or an externally generated reset. When exiting the stop mode, the internal oscillator will resume after a 4064 PH2 clock cycle oscillator stabilization delay.

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

6.5.0.2 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 external interrupt. All other registers, memory, and input/output lines remain in their previous states.

If the 16-bit timer interrupt is enabled, the processor will exit halt mode and resume normal operation. Halt mode also can be exited when an external interrupt or external 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 stop mode), which has been free-running (a feature of wait mode).

NOTE: *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 inadvertently executed.*

6.5.1 WAIT Instruction

The WAIT instruction places the MCU in a low-power mode, which consumes more power than 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 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 wait mode and resume normal operation. The 16-bit timer may be used to generate a periodic exit from the wait mode. Wait mode may also be exited when an external interrupt or reset occurs.

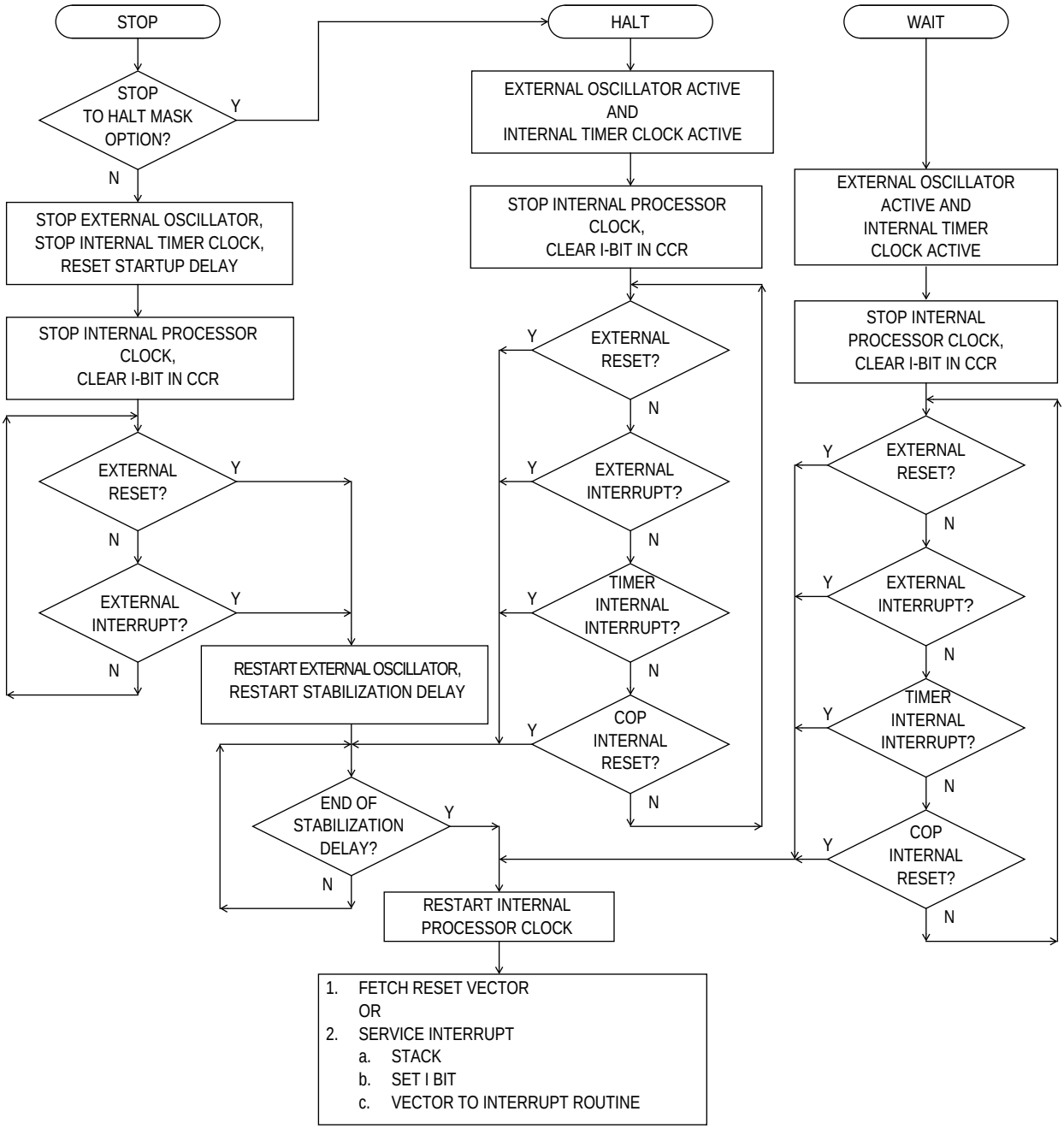


Figure 6-1. STOP/HALT/WAIT Flowchart

6.6 COP Watchdog Timer Considerations

The COP watchdog timer is active in single-chip mode of operation when selected by mask option. Executing the STOP instruction without conversion to halt (via mask option) will cause the COP to be disabled. Therefore, it is recommended that the STOP instruction be modified to produce halt mode (via mask option) 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 timeout period.

COP watchdog timer interactions are summarized in **Table 6-2**.

Table 6-2. COP Watchdog Timer Recommendations

IF the Following Condition Exists:		THEN the COP Watchdog Timer Should Be:
STOP Instruction Modes	**Wait Period**	
Halt Mode Selected via Mask Option	Wait Period Less Than COP Timeout	Enable or Disable COP via Mask Option
Halt Mode Selected via Mask Option	Wait Period More Than COP Timeout	Disable COP via Mask Option
Stop Mode Selected via Mask Option	Any Length Wait Period	Disable COP via Mask Option

Section 7. Input/Output Ports

7.1 Contents

7.2 Introduction

In the single-chip 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.

7.3 Port A

Port A is an 8-bit bidirectional port, which can share its pins with the 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.

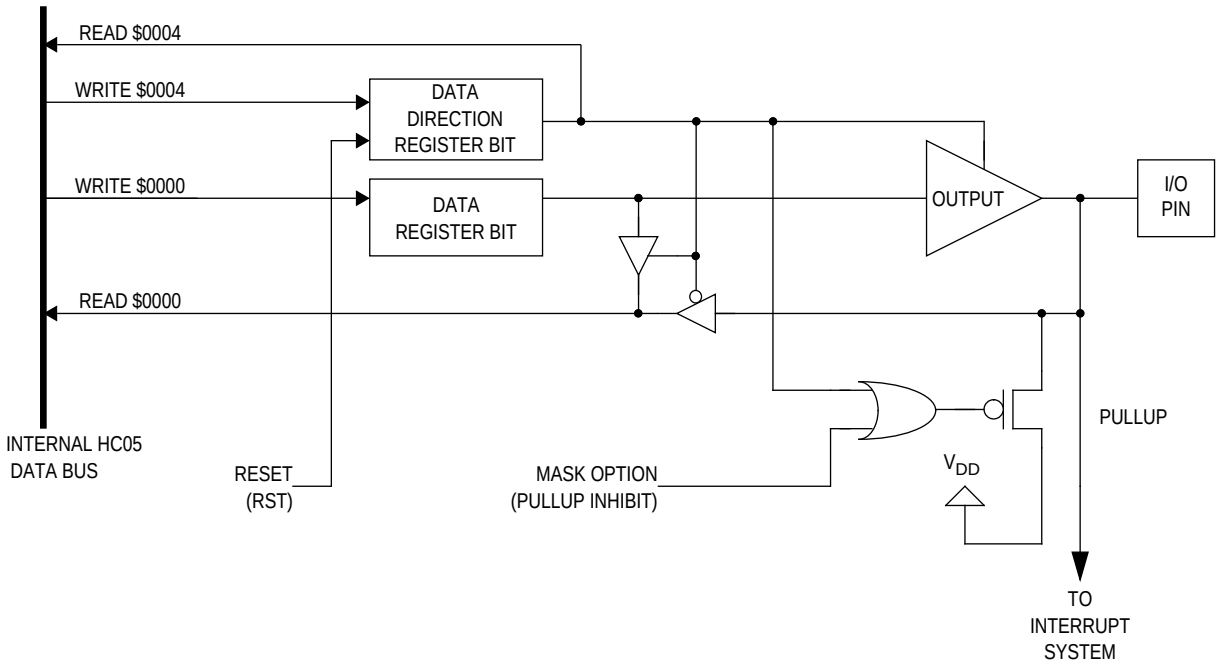


Figure 7-1. Port A I/O Circuitry

7.4 Port B

Port B is a 3-bit bidirectional port that does not share any of its pins with other subsystems. 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 1 to a DDR bit sets the corresponding port pin to output mode (see **Figure 7-2**).

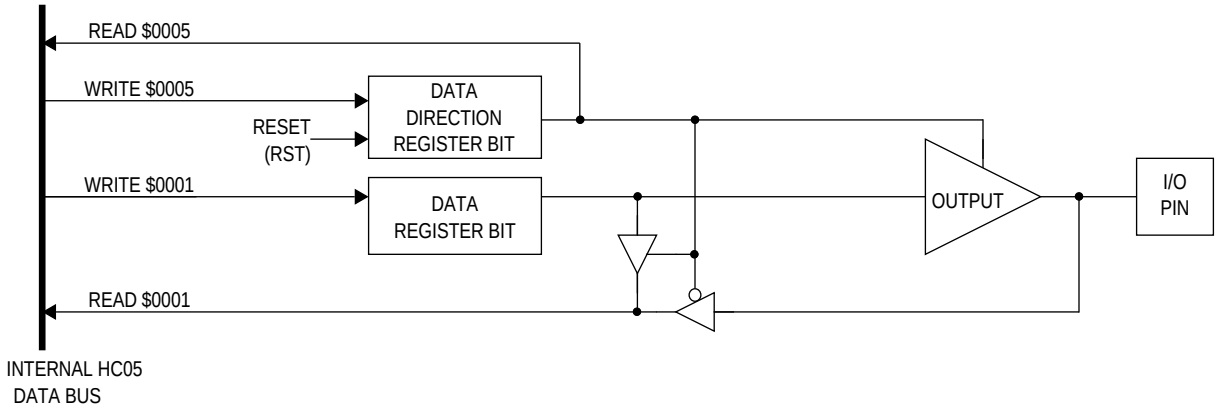


Figure 7-2. Port B I/O Circuitry

7.5 Port C

Port C is an 8-bit bidirectional port that does not share any of its pins with other subsystems. 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 1 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 10. Electrical Specifications** regarding current specifications.

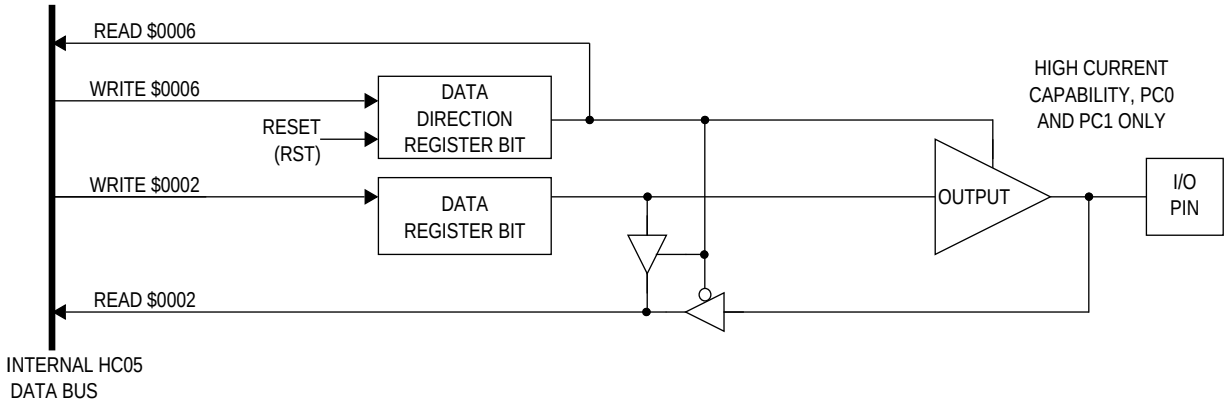


Figure 7-3. Port C I/O Circuitry

7.6 Port D

Port D is a 2-bit port with one bidirectional pin (PD5) and one input-only pin (PD7). Pin PD7 is shared with the 16-bit timer. 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 to input mode. Writing a 1 to DDR bit 5 sets PD5 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.

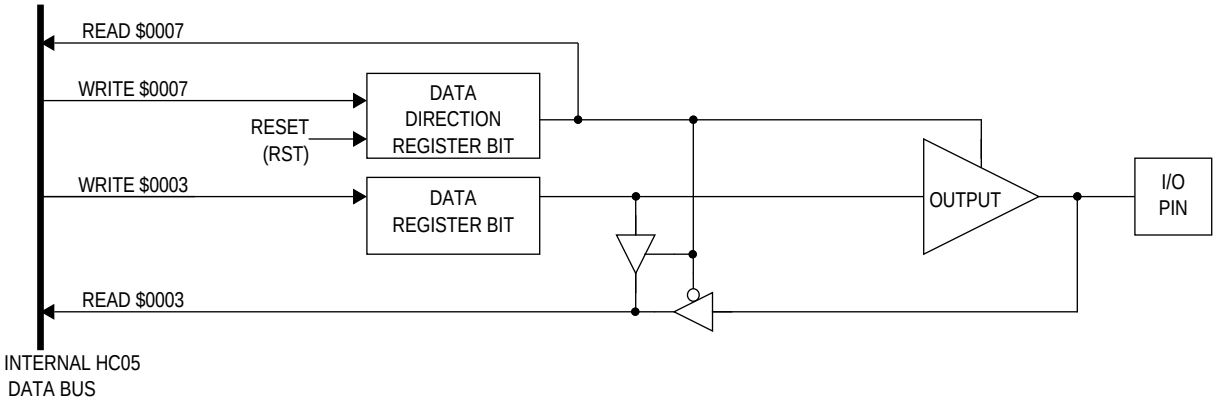


Figure 7-4. Port D I/O Circuitry

7.7 I/O Port Programming

Each pin on ports A through 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 1. A pin is configured as an input if its corresponding DDR bit is cleared to a logic 0.

Table 7-1. Port A I/O Pin Functions

DDRA	I/O Pin Mode	Accesses to DDRA @ $0004	Accesses to Data Register @ $0000		IRQ Source
		Read/Write	Read	Write	
0	Input, Hi-Z	DDRA0–DDRA7	I/O Pin	*	Enabled**
1	Output	DDRA0–DDRA7	PA0–PA7	PA0–PA7	Disabled

*Does not affect input, but stored to data register
**If enabled via mask option

Table 7-2. Port B I/O Pin Functions

DDRB	I/O Pin Mode	Accesses to DDRB @ $0005	Accesses to Data Register @ $0001	
		Read/Write	Read	Write
0	Input, Hi-Z	DDRB5–DDRB7	I/O Pin	*
1	Output	DDRB0–DDRB7	PB5–PB7	PB5–PB7

*Does not affect input, but stored to data register

Table 7-3. Port C I/O Pin Functions

DDRC	I/O Pin Mode	Accesses to DDRC @ $0006	Accesses to Data Register @ $0002	
		Read/Write	Read	Write
0	Input, Hi-Z	DDRC0–DDRA7	I/O Pin	*
1	Output	DDRC0–DDRA7	PC0–PC7	PC0–PC7

*Does not affect input, but stored to data register

Table 7-4. Port D I/O Pin Functions

DDRD	I/O Pin Mode	Accesses to DDRD @ $0007	Accesses to Data Register @ $0003	
		Read/Write	Read	Write
0	Input, Hi-Z	DDRD5	I/O Pin	*
1	Output	DDRD5	PD5	PD5

*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 1 to the corresponding data direction register.*

At power-on or reset, all DDRs are cleared, which configures all port pins as inputs. The DDRs are capable of being written to or read by the processor. During the programmed output state, a read of the data register will actually read the value of the output data latch and not the level on the I/O port pin.

Section 8. 16-Bit Timer

8.1 Contents

8.2 Introduction

The MC68HC05P1A 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 also is capable of generating periodic interrupts. See **Figure 8-1**.

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.

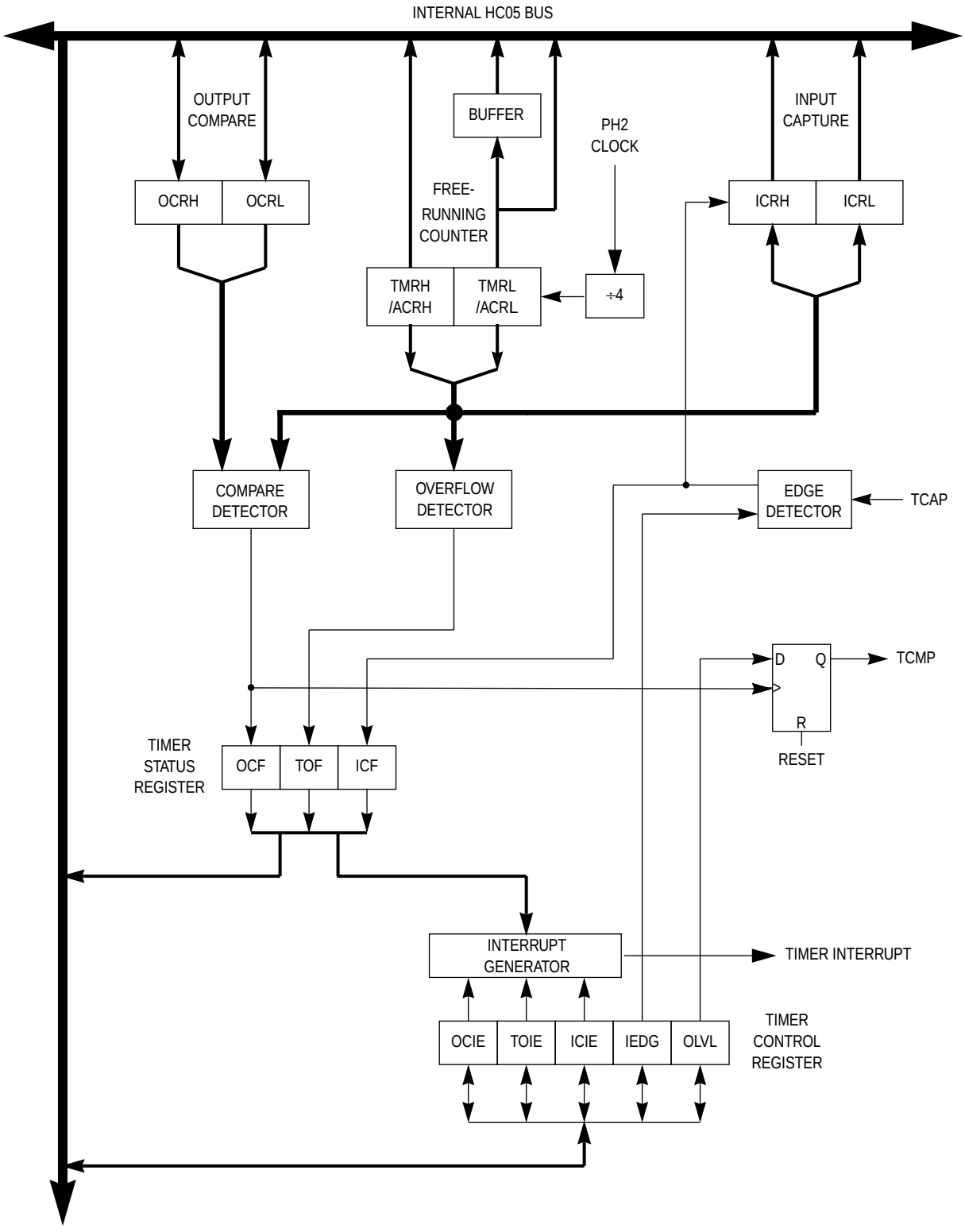


Figure 8-1. 16-Bit Timer Block Diagram

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 that the high and low bytes are accessed.*

8.3 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 least significant bit (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 most significant bit (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 8-2** and **Figure 8-3**.

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: A read 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 8-4**.

TMRH $0018

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

= Unimplemented

TMRL $0019

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

= Unimplemented

Figure 8-2. Timer Registers (TMRH/TMRL)

ACRH $001A

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

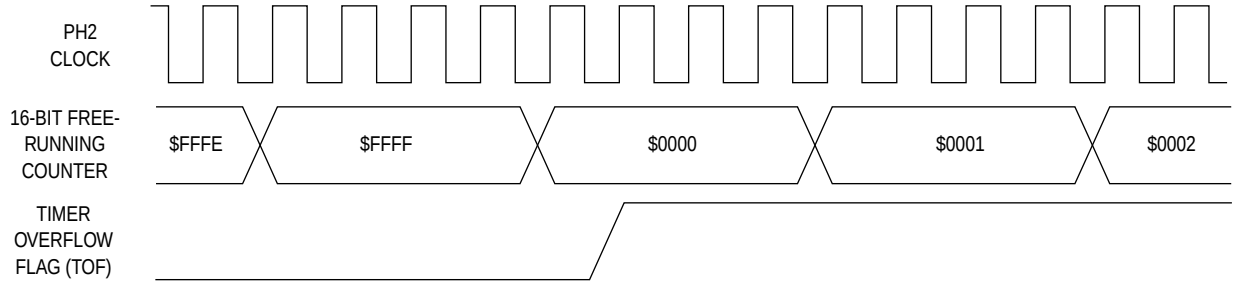
= Unimplemented

ACRL $001B

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

= Unimplemented

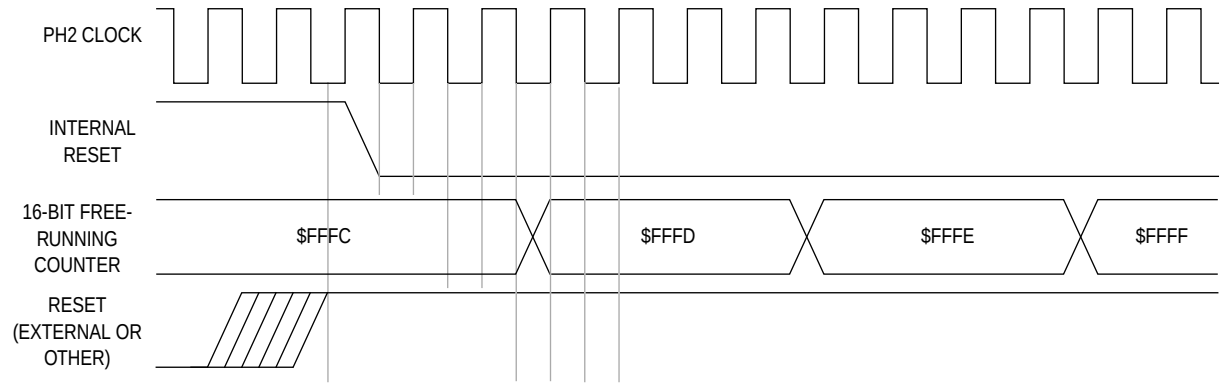
Figure 8-3. Alternate Counter Registers (ACRH/ACRL)



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 8-4. State Timing Diagram for Timer Overflow

The free-running counter is initialized to $FFFC during reset. It 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. When counter rollover occurs, an interrupt also can be enabled by setting the timer overflow interrupt enable bit (TOIE) in register TCR.
See **Figure 8-5**.



NOTE: The counter and control registers are the only 16-bit timer registers affected by reset.

Figure 8-5. State Timing Diagram for Timer Reset

8.4 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 may be used for data storage. See **Figure 8-2**.

OCRH $0016

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	OCRH7	OCRH6	OCRH5	OCRH4	OCRH3	OCRH2	OCRH1	OCRH0
Write:								
Reset:	Unaffected by Reset							

= Unimplemented

OCRL $0017

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	OCRL7	OCRL6	OCRL5	OCRL4	OCRL3	OCRL2	OCRL1	OCRL0
Write:								
Reset:	Unaffected by Reset							

= Unimplemented

Figure 8-6. 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. After each successful comparison, the values in the output compare registers and output level bit should be changed to control an output waveform or to establish a new elapsed timeout. An interrupt also can 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).

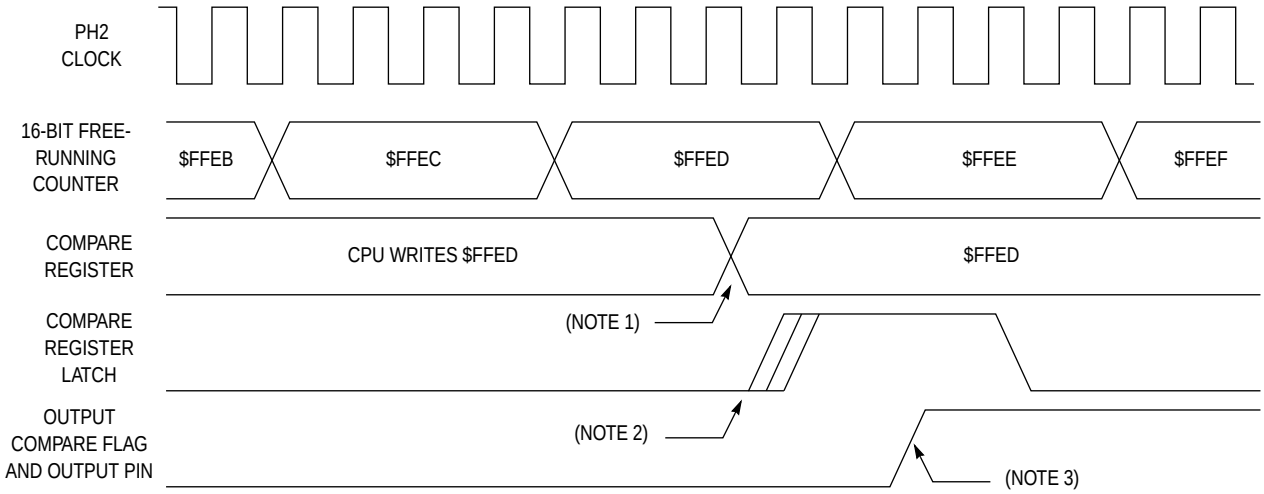
Since neither the output compare flag (OCF) nor the output compare registers are affected by reset, care must be exercised when initializing the output compare function. This procedure is recommended:

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 8-7** and a state timing diagram is shown in **Figure 8-8**.

```
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 COM-
.   .   .   .                   PARE
.   .   .   .                   .
9A          CLI                 .
```

Figure 8-7. Output Compare Software Initialization Example



NOTES:

1. The CPU write to the compare register may take place at any time, but a compare only occurs at timer state T01. Thus, up to a four cycle difference may exist between the write to the compare register and the actual compare.
2. Internal compare takes place during timer state T01.
3. The output compare flag bit (OCF) is set at timer state T11 which follows the comparison match ($FFED in this example).

Figure 8-8. State Timing Diagram for Output Compare

8.5 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 that 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 8-2**.

ICRH $0014

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ICRH7	ICRH6	ICRH5	ICRH4	ICRH3	ICRH2	ICRH1	ICRH0
Write:								
Reset:	Unaffected by Reset							

= Unimplemented

ICRL $0015

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ICRL7	ICRL6	ICRL5	ICRL4	ICRL3	ICRL2	ICRL1	ICRL0
Write:								
Reset:	Unaffected by Reset							

= Unimplemented

Figure 8-9. Input Compare Registers (ICRH/ICRL)

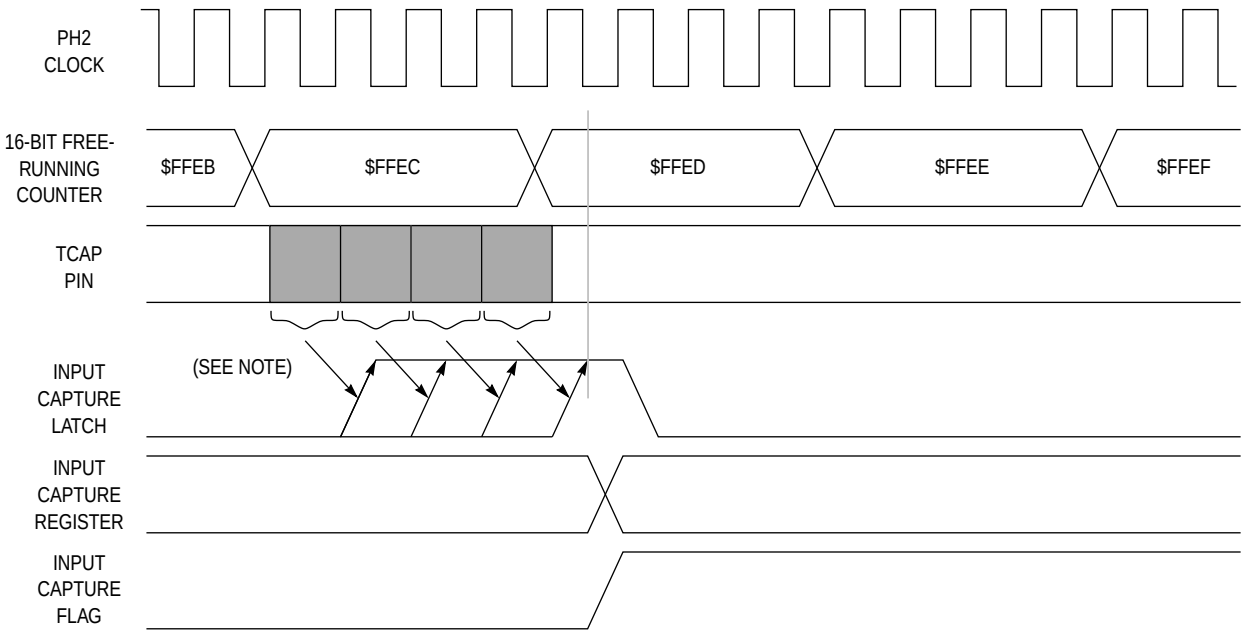
The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the PH2 clock preceding the external transition (see **Figure 8-10**). This delay is required for internal synchronization. Resolution is affected by the prescaler, allowing the free-running 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 that 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: Although the input capture pin is sampled at the rate of PH2, the internal function is updated at the rate of PH4.

Figure 8-10. State Timing Diagram for Input Capture

8.6 Timer Control Register

The timer control (TCR) 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.

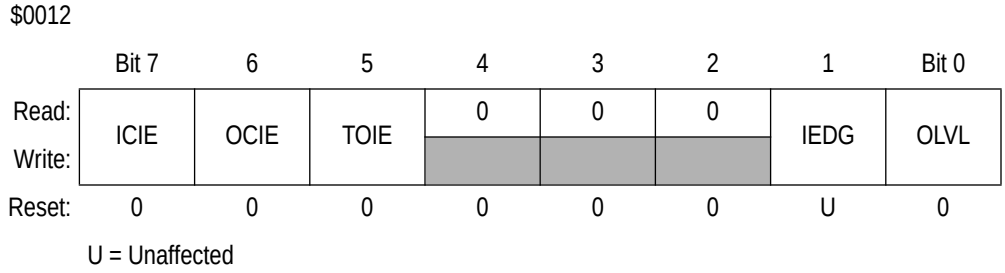


Figure 8-11. Timer Control Register (TCR)

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.

8.7 Timer Status Register

Reading the timer status register (TSR) satisfies the first condition required to clear status flags and interrupts. 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 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.
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.

$0013

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

U = Unaffected

Figure 8-12. 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).

8.8 Timer Operation During Wait Mode

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

8.9 Timer Operation During Stop Mode

When the MCU enters stop mode, the free-running counter stops counting (the PH2 clock is stopped). It remains at that particular count value until stop mode is exited by applying a low signal to the $\overline{\text{IRQ}}$ pin, at which time the counter resumes from its stopped value as if nothing had happened. If stop mode is exited via an external $\overline{\text{RESET}}$ (logic low applied to the $\overline{\text{RESET}}$ 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 stop mode is exited by an external reset, no input capture flag or data will be present even if a valid input capture edge was detected during stop mode.

Section 9. Instruction Set

9.1 Contents

9.2 Introduction

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

9.3 Addressing Modes

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

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

9.3.1 Inherent

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

9.3.2 Immediate

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

9.3.3 Direct

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

9.3.4 Extended

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

When using the Freescale 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.

9.3.5 Indexed, No Offset

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

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

9.3.6 Indexed, 8-Bit Offset

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

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

9.3.7 Indexed,16-Bit Offset

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

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

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

9.3.8 Relative

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

When using the Freescale 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.

9.4 Instruction Types

The MCU instructions fall into five categories:

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

9.4.1 Register/Memory Instructions

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

Table 9-1. 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

9.4.2 Read-Modify-Write Instructions

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

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

Table 9-2. Read-Modify-Write Instructions

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

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

9.4.3 Jump/Branch Instructions

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

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

Table 9-3. Jump and Branch Instructions

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

9.4.4 Bit Manipulation Instructions

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

Table 9-4. Bit Manipulation Instructions

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

9.4.5 Control Instructions

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

Table 9-5. Control Instructions

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

9.5 Instruction Set Summary

Table 9-6. Instruction Set Summary

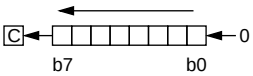
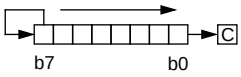
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr* 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* AN*D opr* AND *opr*,X AND *opr*,X AND ,X	Logical AND	A ← (A) ∧ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A4 B4 C4 D4 E4 F4	ii dd hh ll ee ff ff	2 3 4 5 4 3
ASL *opr* ASLA ASLX ASL *opr*,X ASL ,X	Arithmetic Shift Left (Same as LSL)		—	—	↕	↕	↕	DIR INH INH IX1 IX	38 48 58 68 78	dd ff 	5 3 3 6 5
ASR *opr* ASRA ASRX ASR *opr*,X ASR ,X	Arithmetic Shift Right		—	—	↕	↕	↕	DIR INH INH IX1 IX	37 47 57 67 77	dd ff 	5 3 3 6 5
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BCLR *n opr*	Clear Bit n	Mn ← 0	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + 2 + *rel* ? Z = 1	—	—	—	—	—	REL	27	rr	3
BHCC *rel*	Branch if Half-Carry Bit Clear	PC ← (PC) + 2 + *rel* ? H = 0	—	—	—	—	—	REL	28	rr	3
BHCS *rel*	Branch if Half-Carry Bit Set	PC ← (PC) + 2 + *rel* ? H = 1	—	—	—	—	—	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? C ∨ Z = 0	—	—	—	—	—	REL	22	rr	3
BHS *rel*	Branch if Higher or Same	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3

Table 9-6. Instruction Set Summary (Continued)

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

Table 9-6. Instruction Set Summary (Continued)

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

Table 9-6. Instruction Set Summary (Continued)

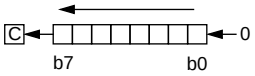
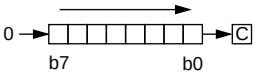
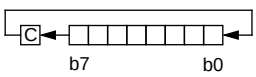
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Effective Address	—	—	—	—	—	DIR EXT IX2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

Table 9-6. Instruction Set Summary (Continued)

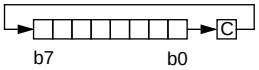
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ROR *opr* RORA RORX ROR *opr*,X ROR ,X	Rotate Byte Right through Carry Bit		—	—	↕×	↕	↕	DIR INH INH IX1 IX	36 46 56 66 76	dd ff 	5 3 3 6 5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2
RTI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

Table 9-6. Instruction Set Summary (Continued)

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

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

9.6 Opcode Map

See **Table 9-7** on page 94.

Table 9-7. Opcode Map

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

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

LSB of Opcode in Hexadecimal

MSB / LSB	0
0	5 BRSET0 3 DIR

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

Section 10. Electrical Specifications

10.1 Contents

10.2 Introduction

This section contains the MCU electrical specifications and timing information.

10.3 Maximum Ratings

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

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

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	–0.3 to + 7.0	V
Input Voltage	V_{IN}	V_{SS} –0.3 to $V_{DD} \pm 0.3$	V
Current Drain per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Storage Temperature Range	T_{STG}	–65 to + 150	°C

NOTE: *This device is not guaranteed to operate properly at the maximum ratings. Refer to* ***10.7 5.0 Volt DC Electrical Characteristics*** *and* ***10.8 3.3 Volt DC Electrical Characteristics*** *for guaranteed operating conditions.*

10.4 Operating Range

Characteristic	Symbol	Value	Unit
Operating Temperature Range MC68HC05P1A (Standard) MC68HC05P1A (Extended) MC68HC05P1A (V)	T_A	T_L to T_H 0 to +70 –40 to +85 –40 to +105	°C

10.5 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal Resistance PDIP SOIC	θ_{JA}	 56 71	°C/W

10.6 Power Considerations

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

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

where:

T_A = Ambient temperature, °C

θ_{JA} = Package thermal resistance, junction-to-ambient, °C/W

$P_D = P_{INT} + P_{I/O}$

$P_{INT} = I_{DD} \times V_{DD}$ watts (chip internal power)

$P_{I/O}$ = Power dissipation on input and output pins (user-determined)

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

The following is an approximate relationship between P_D and T_J (neglecting $P_{I/O}$):

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

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

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

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

10.7 5.0 Volt DC Electrical Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD}-0.1$	— —	0.1 —	V
Output High Voltage (I_{Load} = –0.8 mA) PA0–PA7, PC2–PC7, PB7–PB5, TCMP, PD5 (I_{Load} = –5.0 mA) PC0–PC1	V_{OH}	$V_{DD}-0.8$ $V_{DD}-0.8$	— —	0.4 0.4	V
Output Low Voltage (I_{Load} = –1.6 mA) PA0–PA7, PB5–PB7,PC2–PC7, PD5, TCMP (I_{Load} = 20 mA) PC0–PC1	V_{OL}	— —	— —	0.4 0.4	V
Input High Voltage PA0–PA7, PB5–PB7, PC0–PC7, PD5, TCAP/PD7, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0–PA7, PB5–PB7, PC0–PC7, PD5, TCAP/PD7, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Supply Current Run (Note 3) Wait (Note 4) Stop (Note 5) 25 °C 0 °C to +70 °C (Standard) –40 °C to +85 °C (Extended) –40 °C to +105 °C (V)	I_{DD}	 — — — — — —	 3.5 1.8 1 2 4 6	 5 3.5 15 20 30 50	 mA mA µA µA µA µA
I/O Ports Hi-Z Leakage Current PA0–PA7, PB5–PB7, PC0–PC7, PD5	I_{IL}	—	—	±10	µA
Input Pullup Current PA0–PA7	I_{IL}	5	10	30	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, OSC1, PD5, PD7/TCAP	I_{IN}	—	—	±1	µA
Capacitance PA7–PA0, PB5–PB0 (Input or Output) $\overline{RESET}$, $\overline{IRQ}$, OSC1, OSC2	 C_{OUT} C_{IN}	 — —	 — —	 12 8	 pF

NOTES:

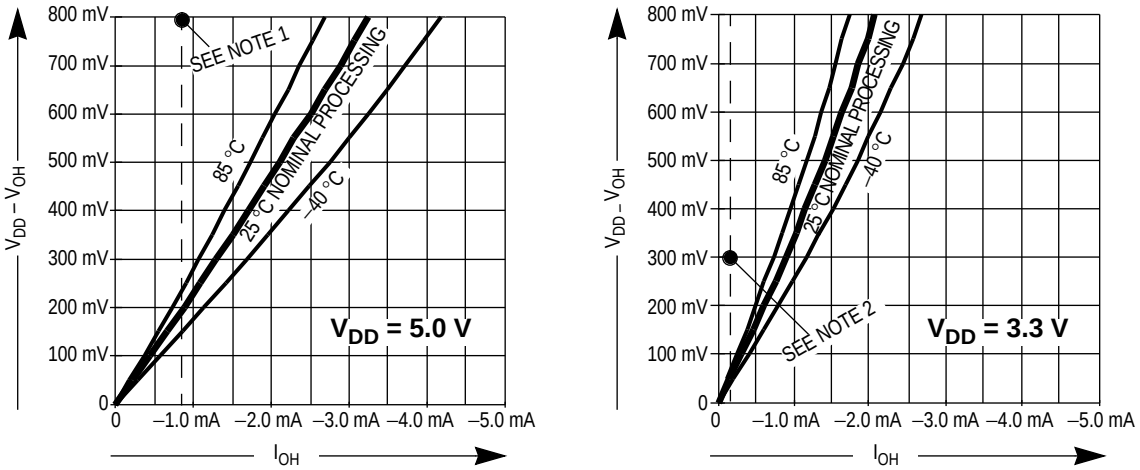
1. V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = –40 °C to +125 °C, unless otherwise noted
2. All values shown reflect average measurements at midpoint of voltage range at 25 °C.
3. Run (operating) I_{DD} and wait I_{DD} measured using external square wave clock source (f_{osc}= 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
4. Wait I_{DD}: Only timer system active. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} –0.2 V. Wait I_{DD} is affected linearly by the OSC2 capacitance.
5. Stop I_{DD} measured with OSC1 = V_{SS}.

10.8 3.3 Volt DC Electrical Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— V_{DD}–0.1	— —	0.1 —	V
Output High Voltage (I_{Load} = –0.2 mA) PA0–PA7, PB5–PB7, PC2–PC7, PD5, TCMP (I_{Load} = –1.5 mA) PC0–PC1	V_{OH}	V_{DD}–0.3 V_{DD}–0.3	— —	— —	V
Output Low Voltage (I_{Load} = 0.4 mA) PA0–PA7, PB5–PB7, PC2–PC7, PD5, TCMP (I_{Load} = 6.0 mA) PC0–PC1	V_{OL}	— —	— —	0.3 0.3	V
Input High Voltage PA0–PA7, PB5–PB7, PC0–PC7, PD5, TCAP/PD7, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA0–PA7, PB5–PB7, PC0–PC7, PD5, TCAP/PD7, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Supply Current Run (Note 3) Wait (Note 4) Stop (Note 5) 25 °C 0 °C to +70 °C (Standard) –40 °C to +85 °C (Extended) –40 °C to +105 °C (V)	I_{DD}	 — — — — — —	 1.0 0.5 0.5 1 2 5	 2.5 1.4 10 15 20 40	 mA mA µA µA µA µA
I/O Ports Hi-Z Leakage Current PA0–PA7, PB5–PB7, PC0–PC7, PD5, TCAP/PD7	I_{IL}	—	—	±10	µA
I/O Pullup Current PA0–PA7	I_{IL}	5	—	10	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
Input Pullup Current (Pullup Device On) PA7–PA0	I_{IN}	1	3	20	µA

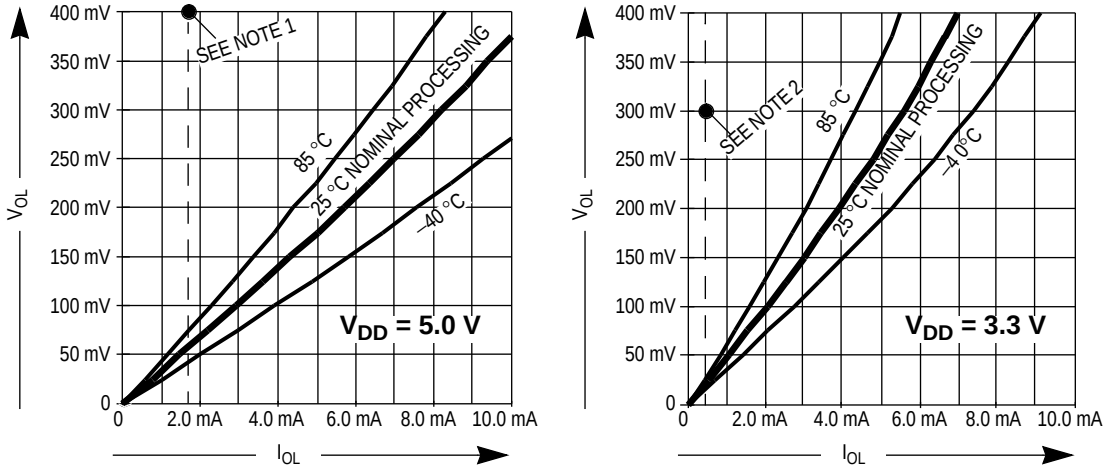
NOTES:

1. V_{DD}= 3.3 Vdc ±10%, V_{SS}= 0 Vdc, T_A = –40 °C to +125 °C, unless otherwise noted
2. All values shown reflect average measurements at midpoint of voltage range at 25 °C.
3. Run (operating) I_{DD} and wait I_{DD} measured using external square wave clock source (f_{osc} = 2.1 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
4. Wait I_{DD}: Only timer system active. Wait, stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} –0.2 V. Wait I_{DD} is affected linearly by the OSC2 capacitance.
5. Stop I_{DD} measured with OSC1 = V_{SS}.



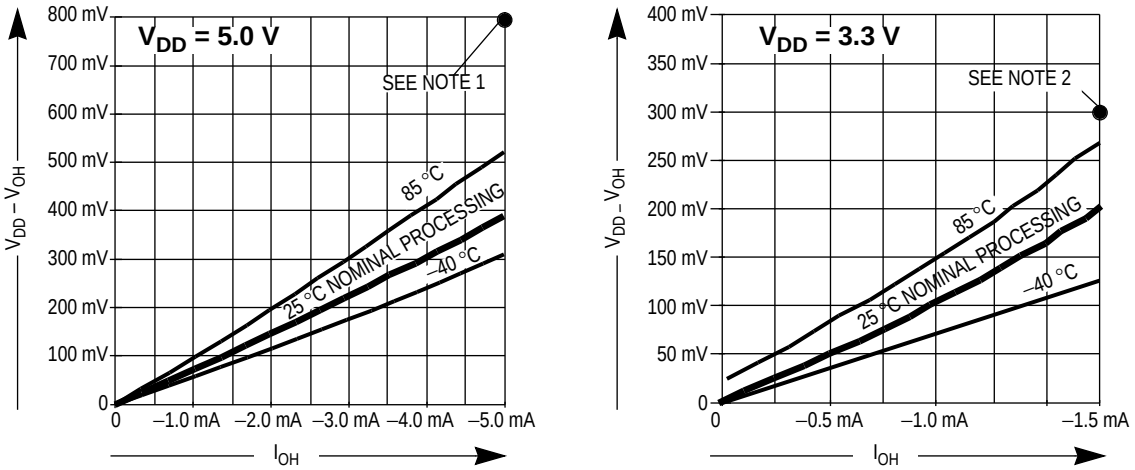
NOTES:
1. At V_{DD} = 5.0 V, devices are specified and tested for ($V_{DD} - V_{OH}$) ≤ 800 mV @ I_{OL} = –0.8 mA.
2. At V_{DD} = 3.3 V, devices are specified and tested for ($V_{DD} - V_{OH}$) ≤ 300 mV @ I_{OL} = –0.2 mA.

Figure 10-1. PA0–PA7, PB5–PB7, PC2–PC5, PD5, and TCMP Typical High-Side Driver Characteristics



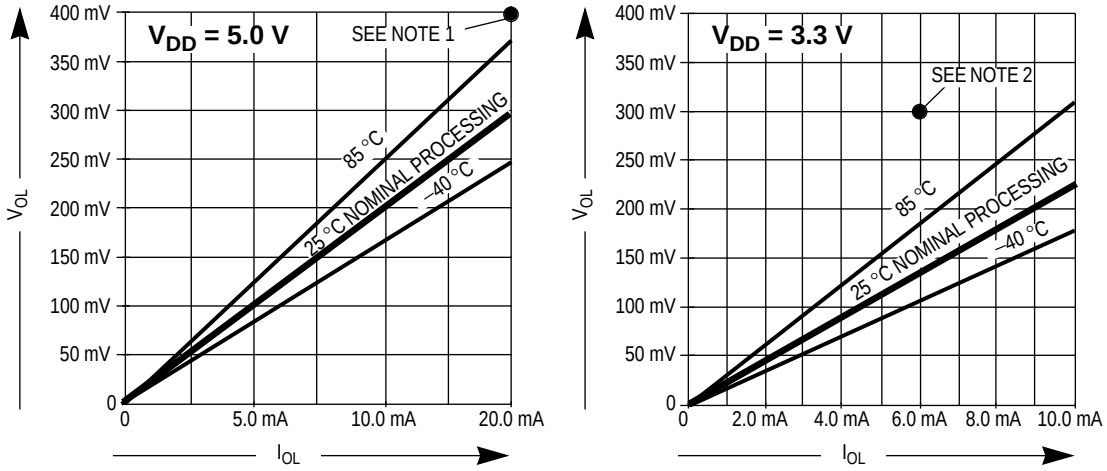
NOTES:
1. At V_{DD} = 5.0 V, devices are specified and tested for V_{OL} ≤ 400 mV @ I_{OL} = 1.6 mA.
2. At V_{DD} = 3.3 V, devices are specified and tested for V_{OL} ≤ 300 mV @ I_{OL} = 0.4 mA.

Figure 10-2. PA0–PA7, PC2–PC5, PB0–PB5, PD5, and TCMP Typical Low-Side Driver Characteristics



NOTES:
1. At V_{DD} = 5.0 V, devices are specified and tested for ($V_{DD} – V_{OH}$) ≤ 800 mV @ I_{OL} = –5.0 mA.
2. At V_{DD} = 3.3 V, devices are specified and tested for ($V_{DD} – V_{OH}$) ≤ 300 mV @ I_{OL} = –1.5 mA.

Figure 10-3. PC0–PC1 Typical High-Side Driver Characteristics



NOTES:
1. At V_{DD} = 5.0 V, devices are specified and tested for V_{OL} ≤ 400 mV @ I_{OL} = 20 mA.
2. At V_{DD} = 3.3 V, devices are specified and tested for V_{OL} ≤ 300 mV @ I_{OL} = 6.0 mA.

Figure 10-4. PC0–PC1 Typical Low-Side Driver Characteristics

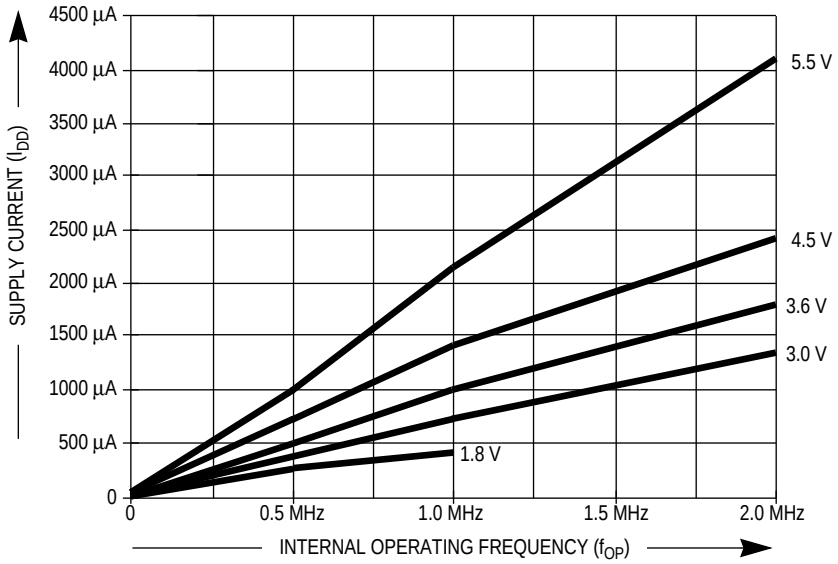


Figure 10-5. Typical Operating I_{DD} (25 °C)

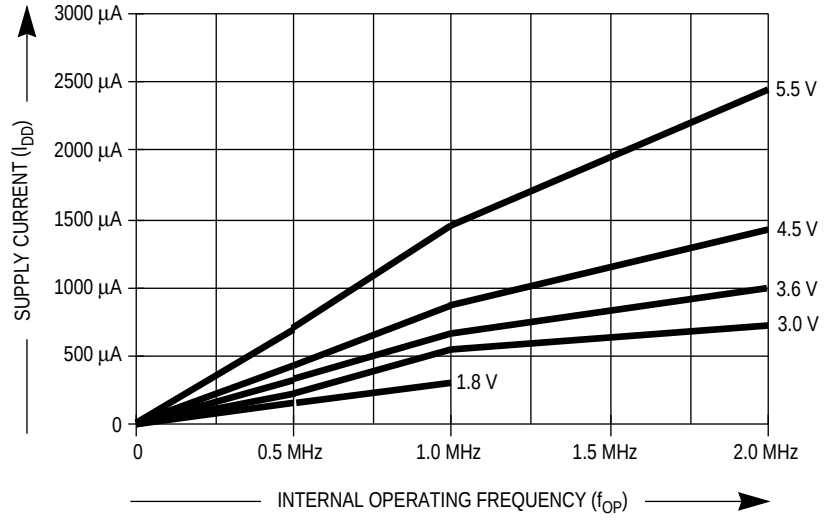


Figure 10-6. Typical Wait Mode I_{DD} (25 °C)

10.9 5.0 Volt Control Timing

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}	Note 2	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

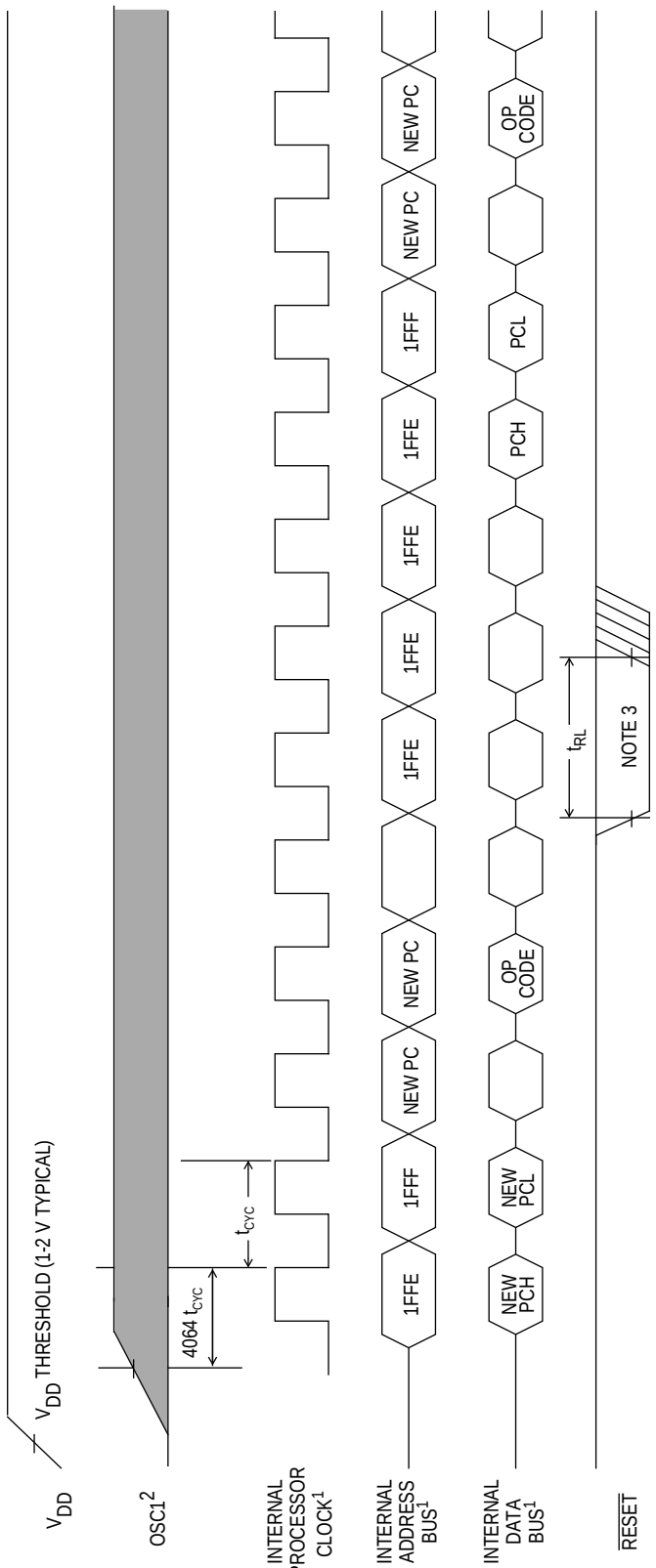
NOTES:
1. V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = –40 °C to +125 °C, unless otherwise noted
2. The minimum period, t_{ILIL}, should not be less than the number of cycles it takes to execute the interrupt service routine plus 19 t_{cyc}.

10.10 3.3 Volt Control Timing

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal/Ceramic Resonator (Note 2) RC Oscillator External Clock Option	f_{osc}	— dc —	2.0 2.0 2.0	MHz
Internal Operating Frequency ($f_{osc} \div 2$) Crystal/Ceramic Oscillator RC Oscillator External Clock	f_{op}	— dc —	2.1 2.1 2.1	MHz
Cycle Time ($2 \div f_{osc}$)	t_{cyc}	1000	—	ns
$\overline{\text{RESET}}$ Pulse Width Low (Edge-Triggered)	t_{RL}	1.5	—	t_{cyc}
Time Resolution (Note 3)	t_{RESL}	4.0	—	t_{cyc}
$\overline{\text{IRQ}}$ Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	250	—	ns
$\overline{\text{IRQ}}$ Interrupt Pulse Period	t_{ILIL}	Note 4	—	t_{cyc}
PA3–PA0 Interrupt Pulse Width High (Edge-Triggered)	t_{IHIL}	250	—	t_{cyc}
PA3–PA0 Interrupt Pulse Period	t_{IHIH}	Note 4	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	400	—	ns

NOTES:

1. V_{DD} = 3.3 Vdc ±10%, V_{SS} = 0 Vdc, T_A = –40 °C to +125 °C, unless otherwise noted
2. Use only AT-cut crystals.
3. The 2-bit timer prescaler is the limiting factor in determining timer resolution.
4. The minimum period, t_{ILIL} or t_{IHIH}, should not be less than the number of cycles it takes to execute the interrupt service routine plus 19 t_{cyc}.



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.

Figure 10-7. Power-On Reset and External Reset Timing Diagram

Electrical Specifications

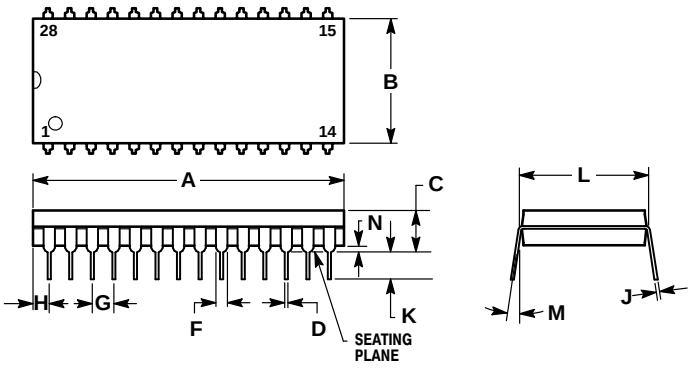
Section 11. Mechanical Specifications

11.1 Contents

11.2 Introduction

This section gives the dimensions of the dual in-line package (DIP) and the small outline integrated circuit (SOIC) package.

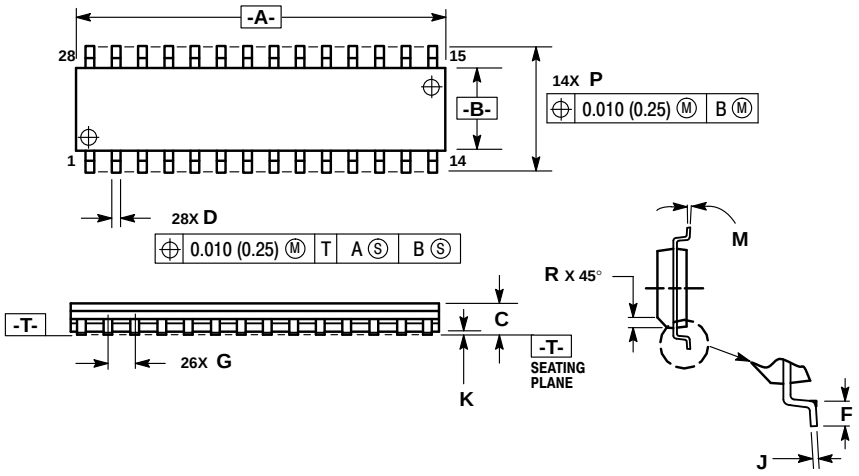
11.3 Dual In-Line Package (Case 710)



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.

	MILLIMETERS		INCHES	
DIM	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

11.4 Small Outline Integrated Circuit (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.

	MILLIMETERS		INCHES	
DIM	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 12. Ordering Information

12.1 Contents

12.2 Introduction

This section contains instructions for ordering custom-masked ROM MCUs.

12.3 MCU Ordering Forms

To initiate an order for a ROM-based MCU, first obtain the current ordering form for the MCU from a Freescale representative. Submit these items when ordering MCUs:

- A current MCU ordering form that is **completely filled out** (Contact your Freescale sales office for assistance.)
- A copy of the customer specification if the customer specification deviates from the Freescale specification for the MCU
- Customer's application program on one of the media listed in **12.4 Application Program Media**

The current MCU ordering form is also available through the Freescale Freeware Bulletin Board Service (BBS). The telephone number is (512) 891-FREE. After making the connection, type bbs in lower-case letters. Then press the return key to start the BBS software.

12.4 Application Program Media

Please deliver the application program to Freescale in one of the following media:

- Macintosh®[1] 3 1/2-inch diskette (double-sided 800 K or double-sided high-density 1.4 M)
- MS-DOS®[2] or PC-DOS™[3] 3 1/2-inch diskette (double-sided 720 K or double-sided high-density 1.44 M)
- MS-DOS® or PC-DOS™ 5 1/4-inch diskette (double-sided double-density 360 K or double-sided high-density 1.2 M)

Use positive logic for data and addresses.

When submitting the application program on a diskette, clearly label the diskette with this information:

- Customer name
- Customer part number
- Project or product name
- File name of object code
- Date
- Name of operating system that formatted diskette
- Formatted capacity of diskette

On diskettes, the application program must be in Motorola's S-record format (S1 and S9 records), a character-based object file format generated by M6805 cross assemblers and linkers.

1. Macintosh is a registered trademark of Apple Computer, Inc.
2. MS-DOS is a registered trademark of Microsoft Corporation.
3. PC-DOS is a trademark of International Business Machines Corporation.

NOTE: *Begin the application program at the first user ROM location. Program addresses must correspond exactly to the available on-chip user ROM addresses as shown in the memory map. Write $00 in all non-user ROM locations or leave all non-user ROM locations blank. Refer to the current MCU ordering form for additional requirements.Freescalemay request pattern re-submission if non-user areas contain any non-zero code.*

If the memory map has two user ROM areas with the same address, then write the two areas in separate files on the diskette. Label the diskette with both file names.

In addition to the object code, a file containing the source code can be included. Freescale keeps this code confidential and uses it only to expedite ROM pattern generation in case of any difficulty with the object code. Label the diskette with the file name of the source code.

12.5 ROM Program Verification

The primary use for the on-chip ROM is to hold the customer's application program. The customer develops and debugs the application program and then submits the MCU order along with the application program.

Freescale inputs the customer's application program code into a computer program that generates a listing verify file. The listing verify file represents the memory map of the MCU. The listing verify file contains the user ROM code and may also contain non-user ROM code, such as self-check code. Freescale sends the customer a computer printout of the listing verify file along with a listing verify form.

To aid the customer in checking the listing verify file, Freescale will program the listing verify file into customer-supplied blank preformatted Macintosh or DOS disks. All original pattern media are filed for contractual purposes and are not returned.

Check the listing verify file thoroughly, then complete and sign the listing verify form and return the listing verify form to Freescale. The signed listing verify form constitutes the contractual agreement for the creation of the custom mask.

12.6 ROM Verification Units (RVUs)

After receiving the signed listing verify form, Freescale manufactures a custom photographic mask. The mask contains the customer's application program and is used to process silicon wafers. The application program cannot be changed after the manufacture of the mask begins. Freescale then produces 10 MCUs, called RVUs, and sends the RVUs to the customer. RVUs are usually packaged in unmarked ceramic and tested to 5 Vdc at room temperature. RVUs are not tested to environmental extremes because their sole purpose is to demonstrate that the customer's user ROM pattern was properly implemented. The 10 RVUs are free of charge with the minimum order quantity. These units are not to be used for qualification or production. RVUs are not guaranteed by Freescale Quality Assurance.

12.7 MC Order Numbers

Table 12-1 shows the MC order numbers for the available package types.

Table 12-1. MC Order Numbers

Package Type	Temperature	MC Order Number
28-Pin Plastic Dual In-Line Package (DIP)	0 °C to +70 °C	MC68HC05P1AP
28-Pin Small Outline Integrated Circuit (SOIC)	0 °C to +70 °C	MC68HC05P1ADW

Appendix A. MC68HCL05P1A

A.1 Contents

A.2 Introduction

This appendix introduces the MC68HCL05P1A, a low-power version of the MC68HC05P1A. All of the information in this document applies to the MC68HCL05P1A with the exceptions given in this appendix.

A.3 DC Electrical Characteristics

The data in **10.7 5.0 Volt DC Electrical Characteristics** and **10.8 3.3 Volt DC Electrical Characteristics** applies to the MC68HCL05P1A with the exceptions given in **Table A-1**, **Table A-2**, and **Table A-3**.

Table A-1. Low-Power Output Voltage (V_{DD} = 1.8–2.4 Vdc)

Characteristic	Symbol	Min	Typ	Max	Unit
Output High Voltage (I_{Load} = –0.1 mA) PA0–PA7, PB5–PB7, PC2–PC7, PD5, TCMP	V_{OH}	V_{DD} –0.3	—	—	V
Output Low Voltage (I_{Load} = 0.2 mA) PA0–PA3, PB5–PB7, PC2–PC7, PD5, TCMP	V_{OL}	—	—	0.3	V

Table A-2. Low-Power Output Voltage (V_{DD} = 2.5–3.6 Vdc)

Characteristic	Symbol	Min	Typ	Max	Unit
Output High Voltage (I_{Load} = –0.2 mA) PA0–PA7, PB5–PB7, PC2–PC7, PD5, TCMP	V_{OH}	V_{DD} –0.3	—	—	V
Output Low Voltage (I_{Load} = 0.4 mA) PA0–PA3, PB5–PB7, PC2–PC7, PD5, TCMP	V_{OL}	—	—	0.3	V

Table A-3. Low-Power Supply Current

Characteristic	Symbol	Min	Typ(1)	Max	Unit
Supply Current (V_{DD} = 4.5–5.5 Vdc, f_{op} = 2.1 MHz)					
Run (Note 2)	I_{DD}	—	3.0	4.25	mA
Wait (Note 3)		—	1.6	2.25	mA
Stop (Note 4)					
25 °C		—	0.5	15	µA
0 °C to +70 °C (Standard)		—	2.0	25	µA
Supply Current (V_{DD} = 2.5–3.6 Vdc, f_{op} = 1.0 MHz)					
Run (Note 2)	I_{DD}	—	1.0	1.6	mA
Wait (Note 3)		—	0.7	1.0	mA
Stop (Note 4)					
25 °C		—	0.2	5.0	µA
0 °C to +70 °C (Standard)		—	2.0	10.0	µA
Supply Current (V_{DD} = 2.5-3.6 Vdc, f_{op} = 500 kHz)					
Run (Note 2)	I_{DD}	—	600	800	µA
Wait (Note 3)		—	350	500	
Stop (Note 4)					
25 °C		—	0.2	5.0	
0 °C to +70 °C (Standard)		—	2.0	10.0	
Supply Current (V_{DD} = 1.8-2.4 Vdc, f_{op} = 500 kHz)					
Run (Note 2)	I_{DD}	—	300	600	µA
Wait (Note 3)		—	200	400	
Stop (Note 4)					
25 °C		—	0.1	2	
0 °C to +70 °C (Standard)		—	2.0	5	

NOTES:

1. Typical values reflect average measurements at midpoint of voltage range at 25 °C.
2. Run (operating) I_{DD} and wait I_{DD} measured using external square wave clock source with all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
3. Wait I_{DD} measured using external square wave clock source with all inputs 0.2 V from rail, no dc loads, less than 50 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. V_{IL} = 0.2 V, V_{IH} = V_{DD} –0.2 V. OSC2 capacitance linearly affects wait I_{DD}.
4. Stop I_{DD} measured with OSC1 = V_{SS}. All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} –0.2 V.

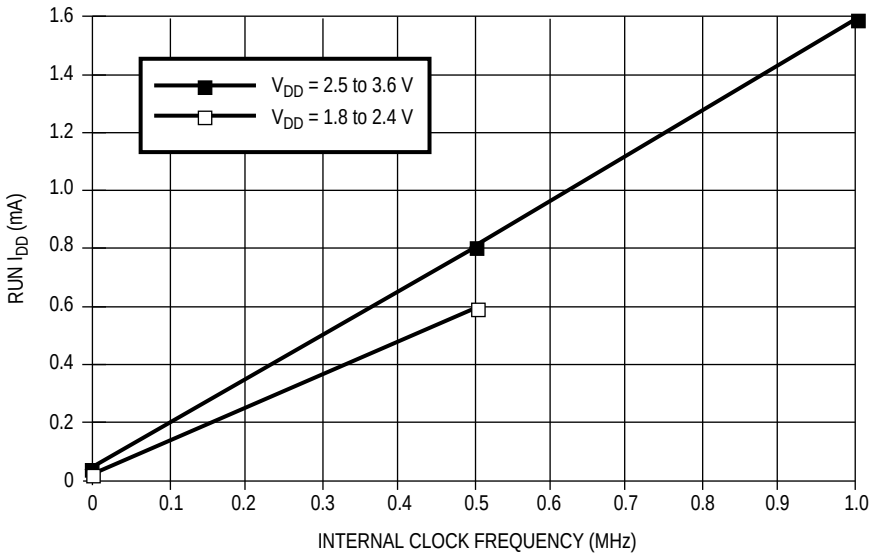


Figure A-1. Maximum Run Mode I_{DD} versus Internal Clock Frequency

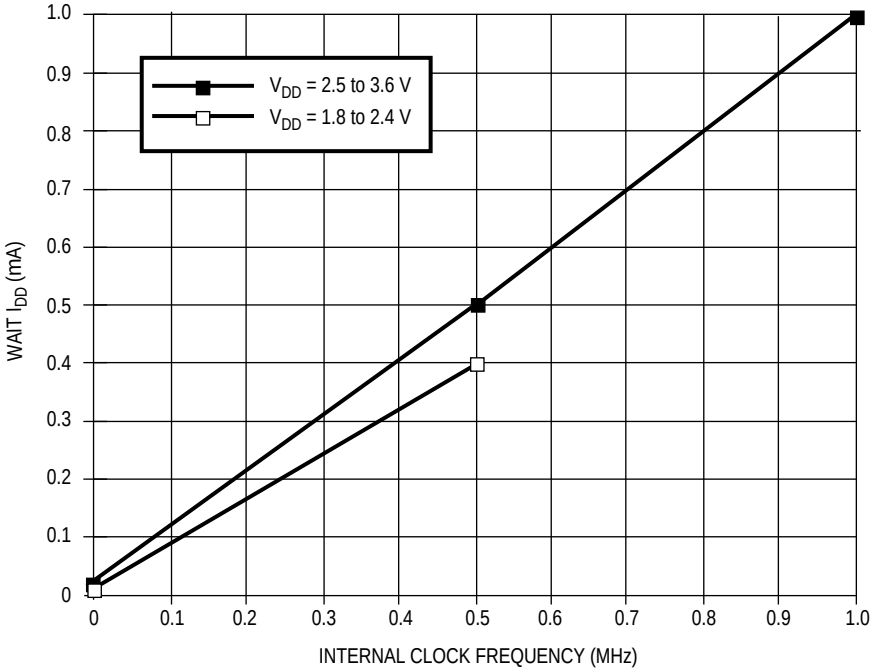


Figure A-2. Maximum Wait Mode I_{DD} versus Internal Clock Frequency

A.4 MC Ordering Information

Table A-4 provides ordering information for available package types.

Table A-4. MC Order Numbers

Package Type	Temperature	MC Order Number
28-Pin Plastic Dual In-Line Package (DIP)	0 °C to +70 °C	MC68HCL05P1AP
28-Pin Small Outline Integrated Circuit (SOIC)	0 °C to +70 °C	MC68HCL05P1ADW

Appendix B. MC68HSC05P1A

B.1 Contents

B.2 Introduction

This appendix introduces the MC68HSC05P1A, a high-speed version of the MC68HC05P1A. All of the information in this document applies to the MC68HCSC05P1A with the exceptions given in this appendix.

B.3 DC Electrical Characteristics

The data in **10.7 5.0 Volt DC Electrical Characteristics** and **10.8 3.3 Volt DC Electrical Characteristics** applies to the MC68HSC05P1A with the exceptions given in **Table B-1**.

Table B-1. High-Speed Supply Current

Characteristic	Symbol	Min	Typ (Note 1)	Max	Unit
Supply Current (V_{DD} = 4.5–5.5 Vdc, f_{op} = 4.0 MHz) Run (Note 3) Wait (Note 4) Stop (Note 5)	I_{DD}	 — — —	 6.0 3.5 2.0	 7.0 3.5 20	 mA mA µA
Supply Current (V_{DD} = 3.0–3.6 Vdc, f_{op} = 2.1 MHz) Rum (Note 3) Wait (Note 4) Stop (Note 5)	I_{DD}	 — — —	 2.5 1.3 2.0	 3.5 2.5 10	 mA mA µA

NOTES:

1. T_A = 0 °C to 70 °C
2. Typical values at midpoint of voltage range, 25 °C only.
3. Run (Operating) I_{DD} and wait I_{DD} measured using external square wave clock source with all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
4. Wait I_{DD} measured using external square wave clock source with all inputs 0.2 V from rail, no dc loads, less than 50 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. V_{IL} = 0.2 V, V_{IH} = V_{DD} –0.2 V. OSC2 capacitance linearly affects wait I_{DD}.
5. Stop I_{DD} measured with OSC1 = V_{SS}. All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} –0.2 V.

B.4 Control Timing

The data in **10.9 5.0 Volt Control Timing** and **10.10 3.3 Volt Control Timing** applies to the MC68HSC05P1A with the exceptions given in **Table B-2** and **Table B-3**.

Table B-2. High-Speed Control Timing (V_{DD} = 5.0 Vdc ± 10%)

Characteristic	Symbol	Min	Max	Unit
Oscillator Frequency Crystal Option External Clock Option	f_{osc}	 — dc	 8.0 8.0	MHz
Internal Operating Frequency Crystal (f_{osc} ÷ 2) External Clock (f_{osc} ÷ 2)	f_{op}	 — dc	 4.0 4.0	MHz
Internal Clock Cycle Time	t_{cyc}	250	—	ns
Input Capture Pulse Width	t_{TH}, t_{TL}	63	—	ns
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	63	—	ns
OSC1 Pulse Width	t_{OH}, t_{OL}	45	—	ns

Table B-3. High-Speed Control Timing (V_{DD} = 3.3 Vdc ± 10%)

Characteristic	Symbol	Min	Max	Unit
Oscillator Frequency Crystal Option External Clock Option	f_{osc}	 — dc	 4.2 4.2	MHz
Internal Operating Frequency Crystal (f_{osc} ÷ 2) External Clock (f_{osc} ÷ 2)	f_{op}	 — dc	 2.1 2.1	MHz
Internal Clock Cycle Time	t_{cyc}	480	—	ns
Input Capture Pulse Width	t_{TH}, t_{TL}	125	—	ns
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	125	—	ns
OSC1 Pulse Width	t_{OH}, t_{OL}	90	—	ns

B.5 MC Ordering Information

Table B-4 provides ordering information for available package types.

Table B-4. MC Order Numbers

Package Type	Temperature	MC Order Number
28-Pin Plastic Dual In-Line Package (DIP)	0 °C to +70 °C	MC68HSC05P1AP
28-Pin Small Outline Integrated Circuit (SOIC)	0 °C to +70 °C	MC68HSC05P1ADW

Home Page:
www.freescale.com
email:
support@freescale.com
USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130
support@freescale.com
Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com
Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 2666 8080
support.japan@freescale.com
Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com
For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

RoHS-compliant and/or Pb- free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb- free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale.s Environmental Products program, go to http://www.freescale.com/epp.

Freescale Semiconductor, Inc.