MC68HC05P1/D
REV 1

HC05

MC68HC05P1

TECHNICAL
DATA

MOTOROLA

MC68HC05P1

HCMOS MICROCONTROLLER UNIT

TABLE OF CONTENTS

Paragraph Number	Title	Page Number

Section 1
Introduction

Section 2
Pin Descriptions

Section 3
Central Processor Unit

TABLE OF CONTENTS (Continued)

Section 4
Resets and Interrupts

TABLE OF CONTENTS (Continued)

Paragraph Number	Title	Page Number

Section 5
Memory

Section 6
Capture/Compare Timer

Section 7
Self-Check Mode

Section 8
Electrical Specifications

TABLE OF CONTENTS (Concluded)

LIST OF FIGURES

LIST OF FIGURES (Concluded)

LIST OF TABLES

SECTION 1
INTRODUCTION

The MC68HC05P1 high-density complementary metal-oxide semiconductor (HCMOS) microcontroller unit (MCU) is a member of the popular M68HC05 Family of microcontrollers. This high-performance, low-cost MCU is a complete system on a single chip. The MCU features include the following:

- Popular M68HC05 Central Processor Unit (CPU)
- Memory-Mapped Input/Output (I/O) Registers
- 2112 Bytes of User ROM Including 16 User Vector Locations
- 128 Bytes of User Static Random Access Memory (SRAM) (Contents Saved in Data-Retention Mode)
- 20 Bidirectional I/O Lines plus One Fixed Input and One Timer Output
- Fully Static Operation (No Minimum Clock Speed)
- On-chip Oscillator with Crystal and Resistor/Capacitor (RC) Mask Options
- 16-Bit Capture/Compare Timer
- Self-Check Mode
- Power-Saving STOP, WAIT, and Data-Retention Modes
- Single 3.0-Volt to 5.5-Volt Power Requirement
- 8×8 Unsigned Multiply Instruction
- 28-pin Dual-In-Line Package (DIP) or Small Outline Integrated Circuit (SOIC)
- Edge Sensitive or Edge- and Level-Sensitive External Interrupt Trigger Mask Options

Figure 1-1 shows the structure of the MC68HC05P1 MCU.

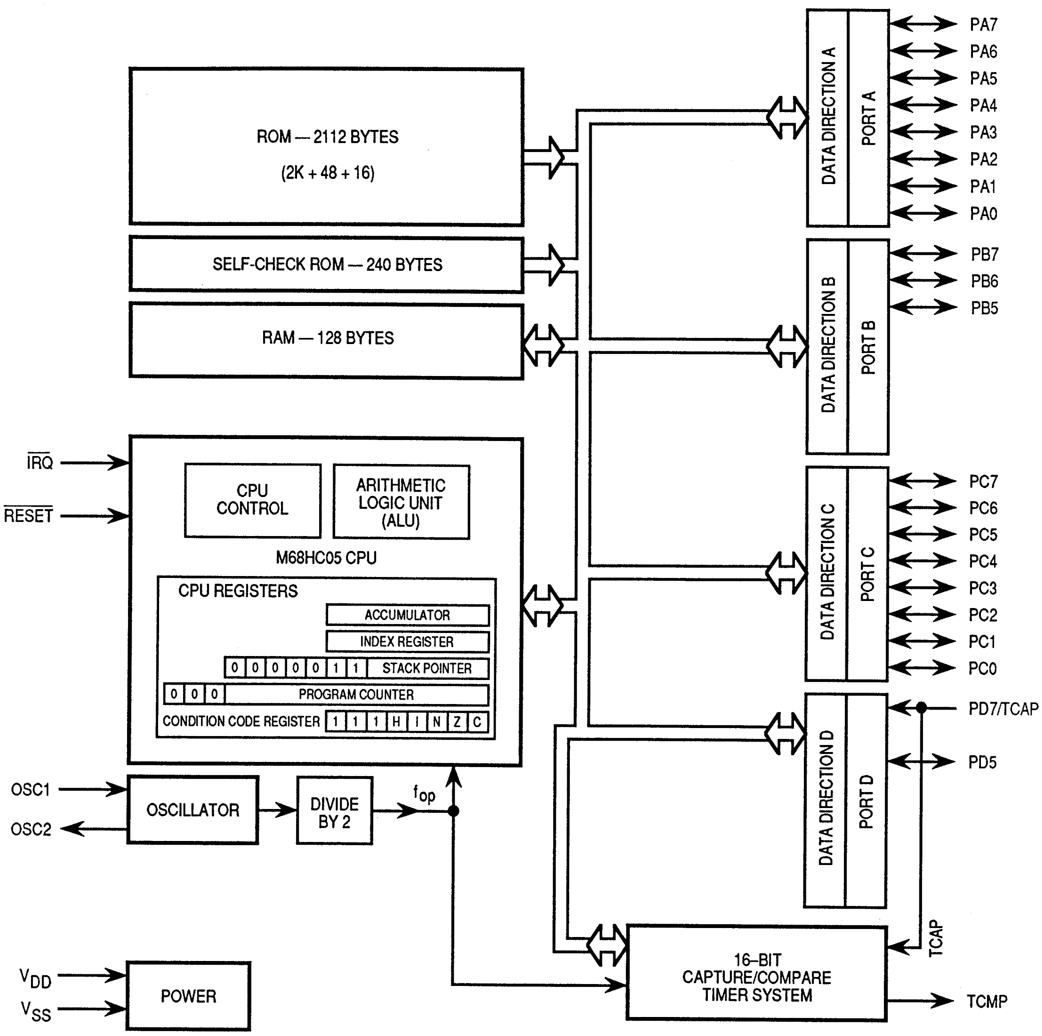


Figure 1-1. MC68HC05P1 Block Diagram

SECTION 2
PIN DESCRIPTIONS

This section describes the functions of the MC68HC05P1 pins. Figure 2-1 shows the pin assignments.

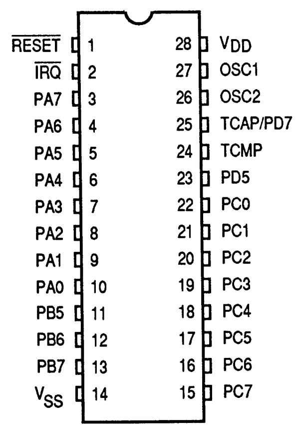


Figure 2-1. Pin Assignments

2.1 V_{DD} and V_{SS}

Power is supplied to the MCU through V_{DD} and V_{SS}. V_{DD} is the power supply, and V_{SS} is ground. The MCU operates from a single 5-volt (nominal) power supply.

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, special care must be taken to provide good power supply bypassing at the MCU. Bypass capacitors should have good high-frequency characteristics and be as close to the MCU as possible. Bypassing requirements vary, depending on how heavily the MCU pins are loaded.

2.2 OSC1 and OSC2 (Oscillator Inputs)

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

- A crystal (Refer to Figure 2-2.)
- A ceramic resonator (Refer to Figure 2-2.)
- An external clock signal connected to OSC1 (Refer to Figure 2-3.)
- A resistor between OSC1 and OSC2 to form an RC circuit with an internal capacitor (Refer to Figure 2-4 for connections and Figure 2-5 for resistance-frequency relationship.)

A factory-set mask option selects either a crystal/ceramic resonator or a resistor as the frequency-determining element. The frequency (f_{OSC}) of the oscillator connected to OSC1 and OSC2 is divided by two to produce the internal operating frequency, f_{op}.

2.2.1 Crystal

The circuit in Figure 2-2 shows a typical crystal oscillator circuit for an AT-cut, parallel resonant crystal. Follow the crystal supplier's recommendations, as the crystal parameters determine the external component values required to provide maximum stability and reliable starting. The load capacitance values used in the oscillator circuit design should include all stray layout capacitances. Mount the crystal and components as close as possible to the input pins to minimize output distortion and start-up stabilization time.

2.2.2 Ceramic Resonator

A ceramic resonator can be used in place of the crystal in cost-sensitive applications. The circuit in Figure 2-2 can be used for a ceramic resonator. Follow the resonator manufacturer's recommendations, as the resonator parameters determine the external component values required to provide maximum stability and reliable starting. The load capacitance values used in the oscillator circuit design should include all stray layout capacitances.

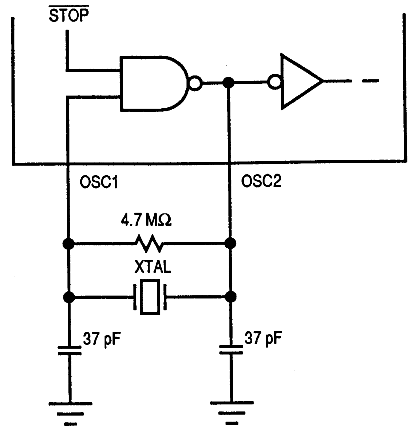


Figure 2-2. Crystal/Ceramic Resonator Connections

2.2.3 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 2-3. When ordering the MCU to use with an external clock, specify the crystal/ceramic resonator mask option.

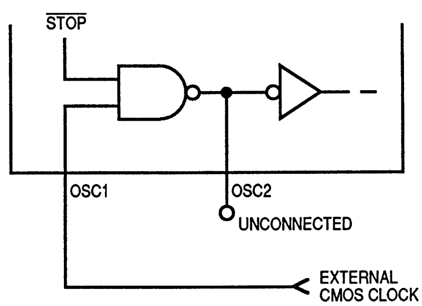


Figure 2-3. External Clock Source Connections

2.2.4 RC Oscillator

With this option, a resistor is connected to the oscillator pins as shown in Figure 2-4. The relationship between R and f_{op} is shown in Figure 2-5. Because the accuracy of the RC oscillator is ±50%, the nominal design frequency must be limited to 66% of the maximum frequency. This ensures that the operating frequency remains below the upper limit. This 50% tolerance only allows for the MCU variation. Make additional allowance for the tolerances of any external components. Operation with a crystal (or ceramic resonator) is preferred.

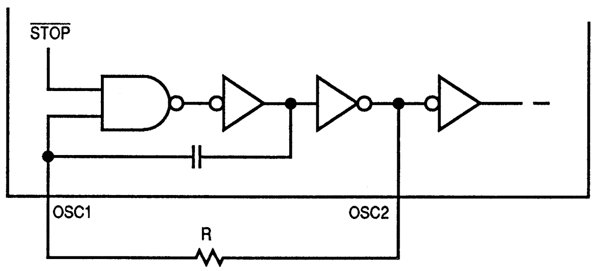


Figure 2-4. RC Oscillator Connections

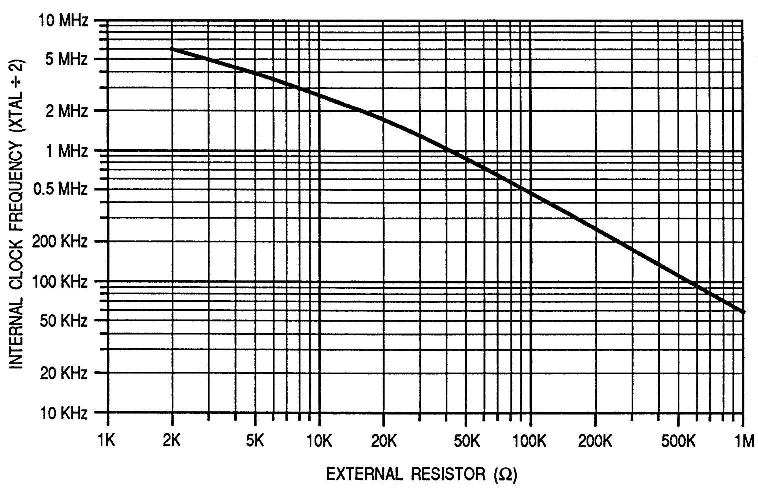


Figure 2-5. RC Oscillator Frequency vs Resistance

2.3 $\overline{\text{RESET}}$

A logical zero on the $\overline{\text{RESET}}$ pin forces the MCU to a known start-up state. (Refer to **4.1 Resets** for more information.)

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

The $\overline{\text{IRQ}}$ pin allows the application of asynchronous external interrupt requests to the MCU. Two different external interrupt triggering sensitivities are available. The factory-set mask options are the following:

- Negative edge-sensitive triggering only
- Both negative edge-sensitive triggering and level-sensitive triggering

Refer to **4.2 Interrupts** for more information.

The $\overline{\text{IRQ}}$ pin is also used in changing operating modes. (Refer to **7.1 Self-Check Circuit**.)

2.5 I/O Port Function

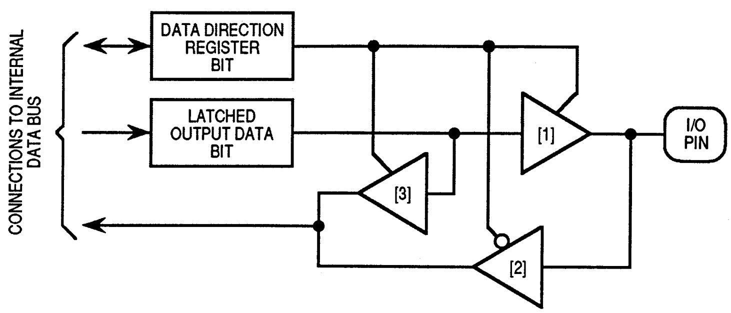
The MCU's 20 I/O pins form four I/O ports. Each I/O pin is programmable as an input or an output. The contents of the data direction register (DDR) determine the data direction for the port. Writing a logical one to a DDR bit enables the output buffer for that pin; a logical zero disables the output buffer. On reset, all implemented DDR bits are initialized to logical zero to put the pins in the input mode.

NOTE

Connect any unused inputs and I/O pins to an appropriate logical level (e.g., either V_{DD} or V_{SS}). Although the I/O ports do not require termination for proper operation, termination is recommended to reduce the possibility of electrostatic damage.

A reset does not initialize the four port data registers. The data registers for ports A, B, C, and D are at addresses $0000, $0001, $0002, and $0003, respectively. To avoid undefined levels, write the data registers before writing the DDR bits.

When a pin is programmed as an output, reading the associated port bit actually reads the value of the output data latch and not the voltage on the pin itself. When a pin is programmed as an input, reading the port bit reads the voltage level on the I/O pin. The output data latch can always be written, regardless of the state of its DDR bit. (Refer to Figure 2-6 for typical port circuitry, and Table 2-1 for a summary of I/O pin functions.)



[1] Output buffer. Enables latched output to drive pin when DDR bit is 1 (output mode).

[2] Input buffer. Enabled when DDR bit is 0 (input mode).

[3] Input buffer. Enabled when DDR bit is 1 (output mode).

Figure 2-6. Parallel I/O Port Circuit

Table 2-1. I/O Pin Functions

$R/\overline{W}$	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch, which drives the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in output mode. The output data latch is read.

NOTE: $R/\overline{W}$ is an internal signal.

2.6 Port A

PA7–PA0 form an 8-bit general-purpose bidirectional I/O port. The contents of data direction register A (DDRA) determine whether each pin is an input or an output. Figure 2-7 shows the port A data register and DDRA.

Bit 7	6	5	4	3	2	1	Bit 0	
								$0000 PORTA
PORT OUTPUT REGISTER STATES NOT CHANGED BY RESET								RESET CONDITION
DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0	$0004 DDRA
0	0	0	0	0	0	0	0	RESET CONDITION: INPUT MODE
I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O	PIN DIRECTIONS
PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PIN NAMES

Figure 2-7. Port A Data Register and DDRA

2.7 Port B

PB7–PB5 form a 3-bit general-purpose bidirectional I/O port. The contents of data direction register B (DDRB) determine whether each pin is an input or an output. Figure 2-8 shows the port B data register DDRB. Bits 4–0 of these registers are not implemented.

Bit 7	6	5	4	3	2	1	Bit 0	
			0	0	0	0	0	$0001 PORTB
PORT OUTPUT REGISTER STATES NOT CHANGED BY RESET								RESET CONDITION
DDRB7	DDRB6	DDRB5	1	1	1	1	1	$0005 DDRB
0	0	0	1	1	1	1	1	RESET CONDITION: INPUT MODE
I/O	I/O	I/O	–	–	–	–	–	PIN DIRECTIONS
PB7	PB6	PB5	–	–	–	–	–	PIN NAMES

Figure 2-8. Port B Data Register and DDRB

2.8 Port C

PC7–PC0 form an 8-bit general-purpose bidirectional I/O port. The contents of data direction register C (DDRC) determine whether each pin is an input or an output. Figure 2-9 shows the port C data register and DDRC.

Bit 7	6	5	4	3	2	1	Bit 0	
								$0002 PORTC
PORT OUTPUT REGISTER STATES NOT CHANGED BY RESET								RESET CONDITION
DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0	$0006 DDRC
0	0	0	0	0	0	0	0	RESET CONDITION: INPUT MODE
I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O	PIN DIRECTIONS
PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PIN NAMES

Figure 2-9. Port C Data Register and DDRC

2.9 Port D and Timer Capture (TCAP)

PD7/TCAP and PD5 form a 2-bit special-function I/O port. The PD7/TCAP pin serves both as the edge-detecting input capture line for the capture/compare timer and as a general-purpose digital input. PD7/TCAP can be used as a digital input even when the timer is using it as the input capture pin. There is no output driver associated with the PD7/TCAP pin. PD5 is a general-purpose digital I/O pin whose direction is controlled by bit 5 of DDRD. Figure 2-10 shows the port D data register and DDRD.

Bit 7	6	5	4	3	2	1	Bit 0	
	0		1	0	0	0	0	$0003 PORTD
PORT OUTPUT REGISTER STATES NOT CHANGED BY RESET								RESET CONDITION
0	0	DDRD5	0	0	0	0	0	$0007 DDRD
0	0	0	0	0	0	0	0	RESET CONDITION: INPUT MODE
I only	–	I/O	–	–	–	–	–	PIN DIRECTIONS
PD7/ TCAP	–	PD5	–	–	–	–	–	PIN NAMES

Figure 2-10. Port D Data Register and DDRD

The PD7/TCAP pin controls the input capture feature of the capture/compare timer. (Refer to **SECTION 6 CAPTURE/COMPARE TIMER** for more information.)

2.10 Timer Compare (TCMP)

The TCMP pin is the output pin for the output compare feature of the capture/compare timer. (Refer to **SECTION 6 CAPTURE/COMPARE TIMER** for more information.)

SECTION 3
CENTRAL PROCESSOR UNIT

This section describes the registers, instruction set, and addressing modes of the M68HC05 central processor unit (CPU). The STOP and WAIT modes, initiated by software instructions, are also described here.

The M68HC05 CPU executes all instructions of the earlier M6805 and M146805 instruction sets and is upgraded to include an 8 × 8 bit unsigned multiply instruction.

3.1 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 MC68HC05 CPU.

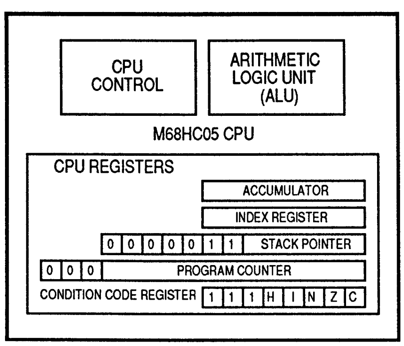


Figure 3-1. CPU Block Diagram

Figure 3-2 shows the five CPU registers.

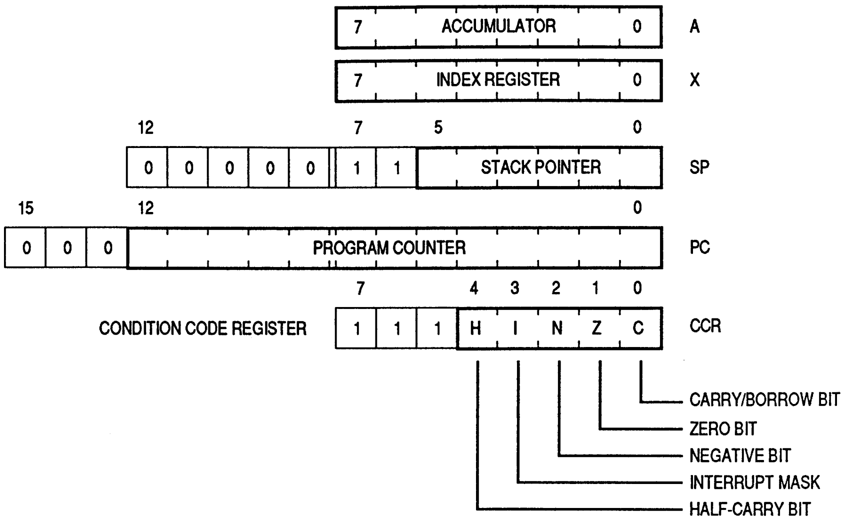


Figure 3-2. Programming Model

3.1.1 Accumulator (A)

The accumulator is a general-purpose 8-bit register. The CPU uses the accumulator to hold operands and results of arithmetic and nonarithmetic operations. (Refer to Figure 3-3.)

Figure 3-3. Accumulator (A)

3.1.2 Index Register (X)

The 8-bit index register can perform two functions:

- Indexed addressing
- Temporary storage

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. (Refer to **3.3 Addressing Modes**.)

The index register can also serve as an auxiliary accumulator for temporary storage. (Refer to Figure 3-4.)

Figure 3-4. Index Register (X)

3.1.3 Stack Pointer (SP)

The stack pointer is a 13-bit register that contains the address of the next free location on the stack. During a reset or after the reset stack pointer (RSP) instruction, the stack pointer contents are set to $00FF. The address in the stack pointer is decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the seven most significant bits of the stack pointer are permanently set to 0000011. (Refer to Figure 3-5.) These seven bits are appended to the six least significant register bits to produce an address within the range of $00FF–$00C0. Subroutines and interrupts may use up to 64 locations. If 64 locations are exceeded, the stack pointer wraps around and writes over the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.

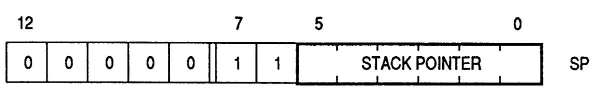


Figure 3-5. Stack Pointer (SP)

3.1.4 Program Counter (PC)

The program counter is a 13-bit register that contains the address of the next instruction or operand 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. (Refer to Figure 3-6.)

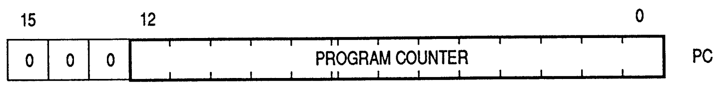


Figure 3-6. Program Counter (PC)

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.

3.1.5 Condition Code Register (CCR)

The 5-bit condition code register uses four bits to indicate the results of the instruction just executed. A fifth bit is the interrupt mask. (Refer to Figure 3-7.) These bits can be individually tested by a program, allowing specific actions as a result of their states. Consider the condition code register as having three additional upper bits that are always ones.

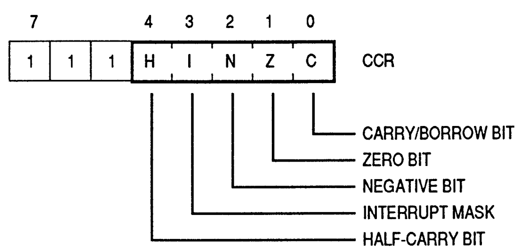


Figure 3-7. Condition Code Register (CCR)

The following paragraphs explain the functions of the lower five bits of the condition code register.

3.1.5.1 Half-Carry Bit (H Bit)

When the half-carry bit is set, it means that a carry occurred between bits 3 and 4 of the accumulator during the last ADD or ADC operation. The half-carry bit is required for binary-coded decimal (BCD) arithmetic operations.

3.1.5.2 Interrupt Mask (I Bit)

When the interrupt mask is set, timer interrupts and external interrupts are disabled. Interrupts are enabled when the interrupt mask is cleared. When an interrupt occurs, the interrupt mask is automatically set after the CPU registers are saved on the stack, but before the interrupt vector is fetched. If an interrupt occurs while the interrupt mask is set, the interrupt is latched. Normally, the interrupt is processed as soon as the interrupt mask is cleared.

A return from interrupt (RTI) instruction pulls the CPU registers from the stack, restoring the interrupt mask to its cleared state. After any reset, the interrupt mask is set and can only be cleared by a software instruction.

3.1.5.3 Negative Bit (N Bit)

The negative bit is set when the result of the last arithmetic operation, logical operation, or data manipulation was negative. (Bit 7 of the result was a logical one.)

The negative bit can also be used to check an often-tested flag by assigning the flag to bit 7 of a register or memory location. Loading the accumulator with the contents of that register or location then sets or clears the negative bit according to the state of the flag.

3.1.5.4 Zero Bit (Z Bit)

The zero bit is set when the result of the last arithmetic operation, logical operation, or data manipulation was zero.

3.1.5.5 Carry/Borrow Bit (C Bit)

The carry/borrow bit is set when a carry out of bit 7 of the accumulator occurred during the last arithmetic operation, logical operation, or data manipulation. The carry/borrow bit is also set or cleared during bit test and branch instructions and during shifts and rotates.

3.2 Arithmetic/Logic Unit (ALU) and CPU Control

The ALU performs the arithmetic and logical operations defined by the instruction set.

The binary arithmetic circuits decode the instruction and set up the ALU for the desired function. Most binary arithmetic is based on the addition algorithm, carrying out subtraction as negative addition. Multiplication is not performed as a discrete instruction but as a chain of addition and shift operations within the ALU. The multiply instruction (MUL) requires 11 internal processor cycles to complete this chain of operations.

The CPU control circuitry sequences the logic elements of the ALU to perform the required operations.

3.3 Addressing Modes

The CPU uses eight addressing modes for flexibility in accessing data. These addressing modes define the manner in which the CPU finds the data required to execute an instruction. The eight addressing modes are as follows:

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

3.3.1 Inherent

The inherent addressing mode is used for instructions with no operand (e.g. STOP) and for some of the instructions that act on data in the CPU registers (e.g., CLRA). No memory address is required for inherent instructions. Inherent instructions are one byte long. Table 3-1 lists the instructions to use in the inherent addressing mode.

Table 3-1. Inherent Addressing Instructions

Instruction	Mnemonic
Arithmetic Shift Left	ASLA, ASLX
Arithmetic Shift Rght	ASRA, ASRX
Clear Carry Bit	CLC
Clear Interrupt Mask	CLI
Clear	CLRA, CLRX
Complement	COMA, COMX
Decrement	DECA, DECX
Increment	INCA, INCX
Logical Shift Left	LSLA, LSLX
Logical Shift Right	LSRA, LSRX
Multiply	MUL
Negate	NEGA, NEGX
No Operation	NOP
Rotate Left through Carry	ROLA, ROLX
Rotate Right through Carry	RORA, RORX
Reset Stack Pointer	RSP
Return from Interrupt	RTI
Return from Subroutine	RTS
Set Carry Bit	SEC
Set Interrupt Mask	SEI
Enable $\overline{IRQ}$ and Stop Oscillator	STOP
Software Interrupt	SWI
Transfer Accumulator to Index Register	TAX
Test for Negative or Zero	TSTA, TSTX
Transfer Index Register to Accumulator	TXA
Enable Interrupt and Halt Processor	WAIT

3.3.2 Immediate

The immediate addressing mode is used for instructions that contain a value to be used in an operation with the value in the accumulator or index register. No memory address is required for immediate instructions. The operand is contained in the byte immediately following the opcode. These are two-byte instructions, one for the opcode and one for the immediate data byte. Table 3-2 lists the instructions to use in the immediate addressing mode.

Table 3-2. Immediate Addressing Instructions

Instruction	Mnemonic
Add with Carry	ADC
Add	ADD
Logical AND	AND
Bit Test Memory with Accumulator	BIT
Compare Accumulator with Memory	CMP
Compare Index Register with Memory	CPX
Exclusive OR Memory with Accumulator	EOR
Load Accumulator from Memory	LDA
Load Index Register from Memory	LDX
Inclusive OR	ORA
Subtract with Carry	SBC
Subtract	SUB

3.3.3 Direct

The direct addressing mode allows the access of data within the first 256 bytes of memory with a single two-byte instruction. In the direct addressing mode, the low byte of the operand's address is contained in the byte following the opcode. The high byte of the address is assumed to be $00. Most direct instructions take two bytes, one for the opcode and one for the operand's address. BRSET and BRCLR are three-byte instructions that use direct addressing to access the operand and relative addressing to specify a branch destination. Table 3-3 lists the instructions to use in the direct addressing mode.

Table 3-3. Direct Addressing Instructions

Instruction	Mnemonic
Add with Carry	ADC
Add	ADD
Logical AND	AND
Arithmetic Shift Left	ASL
Arithmetic Shift Right	ASR
Clear Bit in Memory	BCLR
Bit Test Memory with Accumulator	BIT
Branch if Bit n Is Clear	BRCLR
Branch if Bit n Is Set	BRSET
Set Bit in Memory	BSET
Clear	CLR
Compare Accumulator with Memory	CMP
Complement	COM
Compare Index Register with Memory	CPX
Decrement	DEC
Exclusive OR Memory with Accumulator	EOR
Increment	INC
Jump	JMP
Jump to Subroutine	JSR
Load Accumulator from Memory	LDA
Load Index Register from Memory	LDX
Logical Shift Left	LSL
Logical Shift Right	LSR
Negate	NEG
Inclusive OR	ORA
Rotate Left through Carry	ROL
Rotate Right through Carry	ROR
Subtract with Carry	SBC
Store Accumulator in Memory	STA
Store Index Register in Memory	STX
Subtract	SUB
Test for Negative or Zero	TST

NOTE: ASL = LSL

3.3.4 Extended

The extended addressing mode allows the access of data in any memory location with a single three-byte instruction. In the extended addressing mode, the high and low bytes of the operand's address are contained in the two bytes following the opcode. Extended instructions take three bytes, one for the opcode and two for the operand's address.

When using the Motorola assembler, the user does not need to specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction. Table 3-4 lists the instructions that can be used in the extended addressing mode.

Table 3-4. Extended Addressing Instructions

Instruction	Mnemonic
Add with Carry	ADC
Add	ADD
Logical AND	AND
Bit Test Memory with Accumulator	BIT
Compare Accumulator with Memory	CMP
Compare Index Register with Memory	CPX
Exclusive OR Memory with Accumulator	EOR
Jump	JMP
Jump to Subroutine	JSR
Load Accumulator from Memory	LDA
Load Index Register from Memory	LDX
Inclusive OR	ORA
Subtract with Carry	SBC
Store Accumulator in Memory	STA
Store Index Register in Memory	STX
Subtract	SUB

3.3.5 Indexed, No Offset

The indexed, no offset addressing mode is used to access data with variable addresses within the first 256 memory locations. The CPU finds the low byte of the operand's conditional address by reading the contents of the index register. The high byte is assumed to be $00. These instructions are only one byte long. The indexed, no offset mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location. Table 3-5 lists the instructions that can be used in the indexed, no offset addressing mode.

3.3.6 Indexed, 8-Bit Offset

The indexed, 8-bit offset addressing mode permits the access of data with variable addresses within the first 511 memory locations. The CPU finds the operand's conditional address by adding the unsigned contents of the index register to the unsigned byte following the opcode. This addressing mode is 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). With this two-byte instruction, k would typically be in the index register, and the address of the beginning of the table would be in the byte following the opcode. Table 3-5 lists the instructions to use in the indexed, 8-bit offset addressing mode.

3.3.7 Indexed, 16-Bit Offset

The indexed, 16-bit offset addressing mode is used to access data with variable addresses at any location in memory. The CPU finds the operand's conditional address by adding the unsigned contents of the 8-bit index register to the 16-bit unsigned word formed by the two bytes following the opcode. The first byte after the opcode is the high byte of the 16-bit offset; the second byte is the low byte. This addressing mode can be used in a manner similar to indexed, 8-bit offset, but this three-byte instruction allows tables to be located anywhere in memory.

As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing. Table 3-5 lists the instructions that can be used in the indexed, 16-bit offset addressing mode.

Table 3-5. Indexed Addressing Instructions

Instruction	Mnemonic	No Offset	8-Bit Offset	16-Bit Offset
Add with Carry	ADC	√	√	√
Add	ADD	√	√	√
Logical AND	AND	√	√	√
Arithmetic Shift Left	ASL	√	√	
Arithmetic Shift Right	ASR	√	√	
Bit Test Memory with Accumulator	BIT	√	√	√
Clear	CLR	√	√	
Compare Accumulator with Memory	CMP	√	√	√
Complement	COM	√	√	
Compare Index Register with Memory	CPX	√	√	√
Decrement	DEC	√	√	
Exclusive OR Memory with Accumulator	EOR	√	√	√
Increment	INC	√	√	
Jump	JMP	√	√	√
Jump to Subroutine	JSR	√	√	√
Load Accumulator from Memory	LDA	√	√	√
Load Index Register from Memory	LDX	√	√	√
Logical Shift Left	LSL	√	√	
Logical Shift Right	LSR	√	√	
Negate	NEG	√	√	
Inclusive OR	ORA	√	√	√
Rotate Left through Carry	ROL	√	√	
Rotate Right through Carry	ROR	√	√	
Subtract with Carry	SBC	√	√	√
Store Accumulator in Memory	STA	√	√	√
Store Index Register in Memory	STX	√	√	√
Subtract	SUB	√	√	√
Test for Negative or Zero	TST	√	√	

3.3.8 Relative

The relative addressing mode is used only for branch instructions and bit test and branch instructions. The CPU finds the conditional branch destination by adding the signed byte following the opcode to the contents of the program counter if the branch condition is true. If the branch condition is not true, the program counter goes to the next instruction. To branch either forward or backward, the offset is a signed, twos complement byte that gives a branching range of –127 to +128 bytes from the address of the next location after the branch instruction.

The programmer does not need to calculate the offset when using the Motorola assembler, as it calculates the proper offset and verifies that it is within the span of the branch. Table 3-6 lists the instructions that can be used in the relative addressing mode.

Table 3-6. Relative Addressing Instructions

Instruction	Mnemonic
Branch if Carry Clear	BCC
Branch if Carry Set	BCS
Branch if Equal	BEQ
Branch if Half-Carry Clear	BHCC
Branch if Half-Carry Set	BHCS
Branch if Higher	BHI
Branch if Higher or Same	BHS
Branch if Interrupt Line is High	BIH
Branch if Interrupt Line Is Low	BIL
Branch if Lower	BLO
Branch if Lower or Same	BLS
Branch if Interrupt Mask is Clear	BMC
Branch if Minus	BMI
Branch if Interrupt Mask Is Set	BMS
Branch if Not Equal	BNE
Branch if Plus	BPL
Branch Always	BRA
Branch if Bit n Is Clear	BRCLR
Branch if Bit n Is Set	BRSET
Branch Never	BRN
Branch to Subroutine	BSR

3.4 Instruction Set

This MCU uses all the instructions available in the M146805 CMOS Family plus the unsigned multiply (MUL) instruction. The MUL instruction allows unsigned multiplication of the contents of the accumulator and the index register. The high-order product is then stored in the index register, and the low-order product is stored in the accumulator.

The MCU instructions can be divided into five basic types:

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

3.4.1 Register/Memory Instructions

Most of these instructions use two operands. One operand is in either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. Use most register/memory instructions in the following addressing modes:

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

Table 3-7 lists the register/memory instructions.

Table 3-7. Register/Memory Instructions

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

3.4.2 Read-Modify-Write Instructions

These instructions read a memory location or a register, modify its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence because it does not write a replacement value. Use read-modify-write instructions in the following addressing modes:

- Inherent
- Direct
- Indexed, no offset
- Indexed, 8-bit offset

Table 3-8 lists the read-modify-write instructions.

Table 3-8. Read-Modify-Write Instructions

Instruction	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (Twos Complement)	NEG
Rotate Left through Carry	ROL
Rotate Right through Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST

3.4.3 Jump/Branch Instructions

Jump instructions allow the CPU to interrupt the normal sequence of the program counter. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Jump instructions can be used in the following addressing modes:

- Direct
- Extended
- Indexed, no offset
- Indexed, 8-bit offset
- Indexed, 16-bit offset

Branch instructions allow the CPU to interrupt the normal sequence of the program counter when a test condition is met. If the test condition is not met, the branch is not performed. All branch instructions are used in the relative addressing mode.

Bit test and branch instructions cause a branch based on the condition of any readable bit in the first 256 memory locations. These three-byte instructions use a combination of direct addressing and relative addressing. The direct address of the byte to be tested is in the byte following the opcode. The third byte is the signed offset byte. The CPU finds the conditional branch destination by adding the third byte to the program counter if the specified bit tests true. The bit to be tested and its condition (set or clear) are 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 (C bit) of the condition code register.

Table 3-9 lists the jump and branch instructions.

Table 3-9. Jump and Branch Instructions

Instruction	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Bit n of M = 0	BRCLR
Branch if Bit n of M = 1	BRSET
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
Branch if Higher or Same	BHS
Branch if Carry Set	BCS
Branch if Lower	BLO
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half-Carry Clear	BHCC
Branch if Half-Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Clear	BMC
Branch if Interrupt Mask Set	BMS
Branch if Interrupt Line Low	BIL
Branch if Interrupt Line High	BIH
Branch to Subroutine	BSR
Jump Unconditional	JMP
Jump to Subroutine	JSR

3.4.4 Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory. Port registers, port data direction registers, timer registers, and on-chip RAM locations are in the first 256 bytes of memory. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations. Bit manipulation instructions are used in the direct addressing mode. Table 3-10 lists these instructions.

Table 3-10. Bit Manipulation Instructions

Instruction	Mnemonic
Set Bit n	BSET n (n = 0 . . . 7)
Clear Bit n	BCLR n (n = 0 . . . 7)
Branch if Bit n of M = 0	BRCLR
Branch if Bit n of M = 1	BRSET

3.4.5 Control Instructions

These register reference instructions control CPU operation during program execution. Control instructions, listed in Table 3-11, are used in the inherent addressing mode.

Table 3-11. Control Instructions

Instruction	Mnemonic
Transfer Accumulator to Index Register	TAX
Transfer Index Register to Accumulator	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask	SEI
Clear Interrupt Mask	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No Operation	NOP
Stop	STOP
Wait	WAIT

3.4.6 Instruction Set Summary

Table 3-12 shows all MC68HC05P1 instructions in all possible addressing modes. For each instruction, the operand construction and the execution time in internal clock cycles (t_{cyc}) are shown. One internal clock cycle equals two oscillator input cycles. The following legend summarizes the symbols and abbreviations used in Table 3-12.

Abbreviations and Symbols

A	–	Accumulator
C	–	Carry/borrow bit in condition code register
CCR	–	Condition code register
dd	–	Address of operand in direct addressing mode (1 byte)
dd rr	–	Address (dd) of operand and offset (rr) of branch instruction for bit test instructions
DIR	–	Direct addressing mode
ee ff	–	High (ee) and low (ff) bytes of offset in indexed, 16-bit offset addressing mode (2 bytes)
EXT	–	Extended addressing mode
ff	–	Offset byte in indexed, 8-bit offset addressing mode (1 byte)
H	–	Half-carry bit in condition code register
hh ll	–	High (hh) and low (ll) bytes of operand address in extended addressing mode (2 bytes)
I	–	Interrupt mask in condition code register
ii	–	Operand byte for immediate addressing mode
IMM	–	Immediate addressing mode
INH	–	Inherent addressing mode
IX	–	Indexed, no offset addressing mode
IX1	–	Indexed, 8-bit offset addressing mode
IX2	–	Indexed, 16-bit offset addressing mode
M	–	Any memory location (1 byte)
N	–	Negative bit in condition code register
n	–	Any bit (7,6,5 . . . 0)
opr	–	Operand byte
PC	–	Program counter
PCH	–	Program counter high byte
PCL	–	Program counter low byte
REL	–	Relative addressing mode
rel	–	Offset byte for relative addressing mode
rr	–	Offset byte of branch instruction
SP	–	Stack pointer
X	–	Index register
Z	–	Zero bit in condition code register

$\updownarrow$	–	Set if true; clear if not true	$-(\ )$	–	Negation (twos complement)
–	–	Not affected	+	–	Inclusive OR
?	–	If	$\oplus$	–	Exclusive OR
0	–	Cleared (logical zero)	$\overline{}$	–	NOT
1	–	Set (logical one)	$\times$	–	Multiplication
( )	–	Contents of	+	–	Addition
←	–	Is loaded with	–	–	Subtraction
•	–	AND	:	–	Concatenated with

Table 3-12. Instruction Set (Sheet 1 of 4)

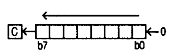
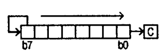
Source Form(s)	Operation	Description	Addressing Mode for Operand	Machine Coding (hexadecimal) Opcode	Operand	Cycles	Condition Code H	I	N	Z	C
ADC opr	Add with carry	A ← (A) + (M) + C	IMM	A9	ii	2	↕	–	↕	↕	↕
			DIR	B9	dd	3					
			EXT	C9	hh ll	4					
			IX2	D9	ee ff	5					
			IX1	E9	ff	4					
			IX	F9		3					
ADD opr	Add without carry	A ← (A) + (M)	IMM	AB	ii	2	↕	–	↕	↕	↕
			DIR	BB	dd	3					
			EXT	CB	hh ll	4					
			IX2	DB	ee ff	5					
			IX1	EB	ff	4					
			IX	FB		3					
AND opr	Logical AND	A ← (A) • (M)	IMM	A4	ii	2	–	–	↕	↕	–
			DIR	B4	dd	3					
			EXT	C4	hh ll	4					
			IX2	D4	ee ff	5					
			IX1	E4	ff	4					
			IX	F4		3					
ASL opr	Arithmetic shift left		DIR	38	dd	5	–	–	↕	↕	↕
ASLA			INH	48		3					
ASLX			INH	58		3					
ASL opr			IX1	68	ff	6					
ASL opr			IX	78		5					
ASR opr	Arithmetic shift right		DIR	37	dd	5	–	–	↕	↕	↕
ASRA			INH	47		3					
ASRX			INH	57		3					
ASR opr			IX1	67	ff	6					
ASR opr			IX	77		5					
BCC rel	Branch if carry bit clear	? C = 0	REL	24	rr	3	–	–	–	–	–
BCLR n opr	Clear bit n	Mn ← 0	DIR (b0)	11	dd	5	–	–	–	–	–
			DIR (b1)	13	dd	5					
			DIR (b2)	15	dd	5					
			DIR (b3)	17	dd	5					
			DIR (b4)	19	dd	5					
			DIR (b5)	1B	dd	5					
			DIR (b6)	1D	dd	5					
			DIR (b7)	1F	dd	5					
BCS rel	Branch if carry bit set	? C = 1	REL	25	rr	3	–	–	–	–	–
BEQ rel	Branch if equal	? Z = 1	REL	27	rr	3	–	–	–	–	–
BHCC rel	Branch if half carry bit clear	? H = 0	REL	28	rr	3	–	–	–	–	–
BHCS rel	Branch if half carry bit set	? H = 1	REL	29	rr	3	–	–	–	–	–
BHI rel	Branch if higher	? C + Z = 0	REL	22	rr	3	–	–	–	–	–
BHS rel	Branch if higher or same	? C = 0	REL	24	rr	3	–	–	–	–	–
BIH rel	Branch if $\overline{IRQ}$ pin high	? $\overline{IRQ}$ = 1	REL	2F	rr	3	–	–	–	–	–
BIL rel	Branch if $\overline{IRQ}$ pin low	? $\overline{IRQ}$ = 0	REL	2E	rr	3	–	–	–	–	–
BIT rel	Bit test accumulator contents with memory contents	(A) • (M)	IMM	A5	ii	2	–	–	↕	↕	–
			DIR	B5	dd	3					
			EXT	C5	hh ll	4					
			IX2	D5	ee ff	5					
			IX1	E5	ff	4					
			IX	F5		3					
BLO rel	Branch if lower	? C = 1	REL	25	rr	3	–	–	–	–	–
BLS rel	Branch if lower or same	? C + X = 1	REL	23	rr	3	–	–	–	–	–
BMC rel	Branch if interrupt mask clear	? I = 0	REL	2C	rr	3	–	–	–	–	–
BMI rel	Branch if minus	? N = 1	REL	2B	rr	3	–	–	–	–	–
BMS rel	Branch if interrupt mask set	? I = 0	REL	2D	rr	3	–	–	–	–	–
BNE rel	Branch if not equal	? Z = 0	REL	26	rr	3	–	–	–	–	–
BPL rel	Branch if plus	? N = 0	REL	2A	rr	3	–	–	–	–	–

Table 3-12. Instruction Set (Sheet 2 of 4)

Source Form(s)	Operation	Description	Addressing Mode for Operand	Machine Coding (hexadecimal) Opcode	Operand	Cycles	Condition Code H	I	N	Z	C
BRA rel	Branch always	? 1 = 1	REL	20	rr	3	–	–	–	–	–
BRCLR n opr rel	Branch if bit n clear	? Mn = 0	DIR (b0)	01	dd rr	5	–	–	–	–	↕
			DIR (b1)	03	dd rr	5					
			DIR (b2)	05	dd rr	5					
			DIR (b3)	07	dd rr	5					
			DIR (b4)	09	dd rr	5					
			DIR (b5)	0B	dd rr	5					
			DIR (b6)	0D	dd rr	5					
			DIR (b7)	0F	dd rr	5					
BRN rel	Branch never	? 1 = 0	REL	21	rr	3	–	–	–	–	–
BRSET n opr rel	Branch if bit n set	? Mn = 1	DIR (b0)	00	dd rr	5	–	–	–	–	↕
			DIR (b1)	02	dd rr	5					
			DIR (b2)	04	dd rr	5					
			DIR (b3)	06	dd rr	5					
			DIR (b4)	08	dd rr	5					
			DIR (b5)	0A	dd rr	5					
			DIR(b6)	0C	dd rr	5					
			DIR (b7)	0E	dd rr	5					
BSET n opr	Set bit n	Mn ← 1	DIR (b0)	10	dd	5	–	–	–	–	–
			DIR (b1)	12	dd	5					
			DIR (b2)	14	dd	5					
			DIR (b3)	16	dd	5					
			DIR (b4)	18	dd	5					
			DIR (b5)	1A	dd	5					
			DIR (b6)	1C	dd	5					
			DIR (b7)	1E	dd	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	INH	98		2	–	–	–	–	0
CLI	Clear interrupt mask	I ← 0	INH	9A		2	–	0	–	–	–
CLR opr	Clear register	M ← $00	DIR	3F	dd	5	–	–	0	1	–
CLRA		A ← $00	INH	4F		3					
CLRX		X ← $00	INH	5F		3					
CLR opr		M ← $00	IX1	6F	ff	6					
CLR opr		M ← $00	IX	7F		5					
CMP opr	Compare accumulator contents with memory contents	(A) – (M)	IMM	A1	ii	2	–	–	↕	↕	↕
			DIR	B1	dd	3					
			EXT	C1	hh ll	4					
			IX2	D1	ee ff	5					
			IX1	E1	ff	4					
			IX	F1		3					
COM opr	Complement register contents	$M \leftarrow \overline{M}$ = $FF – (M)	DIR	33	dd	5	–	–	↕	↕	1
COMA	(ones complement)	$A \leftarrow \overline{A}$ = $FF – (A)	INH	43		3					
COMX		$X \leftarrow \overline{X}$ = $FF – (X)	INH	53		3					
COM opr		$M \leftarrow \overline{M}$ = $FF – (M)	IX1	63	ff	6					
COM opr		$M \leftarrow \overline{M}$ = $FF – (M)	IX	73		5					
CPX opr	Compare index register contents with memory contents	(X) – (M)	IMM	A3	ii	2	–	–	↕	↕	↕
			DIR	B3	dd	3					
			EXT	C3	hh ll	4					
			IX2	D3	ee ff	5					
			IX1	E3	ff	4					
			IX	F3		3					
DEC opr	Decrement register contents	M ← (M) – 1	DIR	3A	dd	5	–	–	↕	↕	–
DECA		A ← (A) – 1	INH	4A		3					
DECX		X ← (X) – 1	INH	5A		3					
DEC opr		M ← (M) – 1	IX1	6A	ff	6					
DEC opr		M ← (M) – 1	IX	7A		5					

Table 3-12. Instruction Set (Sheet 3 of 4)

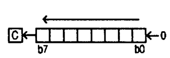
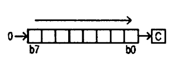
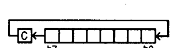
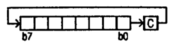
Source Form(s)	Operation	Description	Addressing Mode for Operand	Machine Coding (hexadecimal) Opcode	Operand	Cycles	Condition Code H	I	N	Z	C
EOR opr	Exclusive OR accumulator contents with memory contents	A ← (A) ⊕ (M)	IMM	A8	ii	2	–	–	↕	↕	–
			DIR	B8	dd	3					
			EXT	C8	hh ll	4					
			IX2	D8	ee ff	5					
			IX1	E8	ff	4					
			IX	F8		3					
INC opr	Increment memory or register contents	M ← (M) + 1	DIR	3C	dd	5	–	–	↕	↕	–
INCA		A ← (A) + 1	INH	4C		3					
INCX		X ← (X) + 1	INH	5C		3					
INC opr		M ← (M) + 1	IX1	6C	ff	6					
INC opr		M ← (M) + 1	IX	7C		5					
JMP opr	Unconditional jump	PC ← jump address	DIR	BC	dd	2	–	–	–	–	–
			EXT	CC	hh ll	3					
			IX2	DC	ee ff	4					
			IX1	EC	ff	3					
			IX	FC		2					
JSR opr	Jump to subroutine	PC ← (PC) + n (n = 1, 2, or 3)	DIR	BD	dd	5	–	–	–	–	–
		Push (PCL); SP ← (SP) – 1	EXT	CD	hh ll	6					
		Push (PCH); SP ← (SP) – 1	IX2	DD	ee ff	7					
		PC ← conditional address	IX1	ED	ff	6					
			IX	FD		5					
LDA opr	Load accumulator with memory contents	A ← (M)	IMM	A6	ii	2	–	–	↕	↕	–
			DIR	B6	dd	3					
			EXT	C6	hh ll	4					
			IX2	D6	ee ff	5					
			IX1	E6	ff	4					
			IX	F6		3					
LDX opr	Load index register with memory contents	X ← (M)	IMM	AE	ii	2	–	–	↕	↕	–
			DIR	BE	dd	3					
			EXT	CE	hh ll	4					
			IX2	DE	ee ff	5					
			IX1	EE	ff	4					
			IX	FE		3					
LSL opr	Logical shift left		DIR	38	dd	5	–	–	↕	↕	↕
LSLA			INH	48		3					
LSLX			INH	58		3					
LSL opr			IX1	68	ff	6					
LSL opr			IX	78		5					
LSR opr	Logical shift right		DIR	34	dd	5	–	–	0	↕	↕
LSRA			INH	44		3					
LSRX			INH	54		3					
LSR opr			IX1	64	ff	6					
LSR opr			IX	74		5					
MUL	Unsigned multiply	X : A ← (X) × (A)	INH	42		11	0	–	–	–	0
NEG opr	Negate memory or register contents (twos complement)	M ← –(M) = $00 – (M)	DIR	30	dd	5	–	–	↕	↕	↕
NEGA		A ← –(A) = $00 – (A)	INH	40		3					
NEGX		X ← –(X) = $00 – (X)	INH	50		3					
NEG opr		M ← –(M) = $00 – (M)	IX1	60	ff	6					
NEG opr		M ← –(M) = $00 – (M)	IX	70		5					
NOP	No operation		INH	9D		2	–	–	–	–	–
ORA opr	Inclusive OR accumulator contents with memory contents	A ← (A) + (M)	IMM	AA	ii	2	–	–	↕	↕	–
			DIR	BA	dd	3					
			EXT	CA	hh ll	4					
			IX2	DA	ee ff	5					
			IX1	EA	ff	4					
			IX	FA		3					
ROL opr	Rotate left through carry		DIR	39	dd	5	–	–	↕	↕	↕
ROLA			INH	49		3					
ROLX			INH	59		3					
ROL opr			IX1	69	ff	6					
ROL opr			IX	79		5					

Table 3-12. Instruction Set (Sheet 4 of 4)

Source Form(s)	Operation	Description	Addressing Mode for Operand	Machine Coding (hexadecimal) Opcode	Operand	Cycles	Condition Code H	I	N	Z	C
ROR opr	Rotate right through carry		DIR	36	dd	5	–	–	↕	↕	↕
RORA			INH	46		3					
RORX			INH	56		3					
ROR opr			IX1	66	ff	6					
ROR opr			IX	76		5					
RSP	Reset stack pointer	SP ← $00FF	INH	9C		2	From Stack				
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	Subtract memory contents and carry bit from accumulator contents	A ← (A) – (M) – C	IMM	A2	ii	2	–	–	↕	↕	↕
			DIR	B2	dd	3					
			EXT	C2	hh ll	4					
			IX2	D2	ee ff	5					
			IX1	E2	ff	4					
			IX	F2		3					
SEC	Set carry bit	C ← 1	INH	99		2	–	–	–	–	1
SEI	Set interrupt mask	I ← 1	INH	9B		2	–	1	–	–	–
STA opr	Store accumulator contents in memory	M ← (A)	DIR	B7	dd	4	–	–	↕	↕	–
			EXT	C7	hh ll	5					
			IX2	D7	ee ff	6					
			IX1	E7	ff	5					
			IX	F7		4					
STOP	Enable $\overline{IRQ}$; stop oscillator		INH	8E		2	–	0	–	–	–
STX opr	Store index register contents in memory	M ← (X)	DIR	BF	dd	4	–	–	↕	↕	–
			EXT	CF	hh ll	5					
			IX2	DF	ee ff	6					
			IX1	EF	ff	5					
			IX	FF		4					
SUB opr	Subtract memory contents from accumulator contents	A ← (A) – (M)	IMM	A0	ii	2	–	–	↕	↕	↕
			DIR	B0	dd	3					
			EXT	C0	hh ll	4					
			IX2	D0	ee ff	5					
			IX1	E0	ff	4					
			IX	F0		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 ← ($xFFC) PCL ← ($xFFD) (Vector fetch)	INH	83		10	–	1	–	–	–
TAX	Transfer accumulator contents to index register	X ← (A)	INH	97		2	–	–	–	–	–
TST opr	Test memory, accumulator, or index register contents for negative or zero	(M) – $00	DIR	3D	dd	4	–	–	↕	↕	0
TSTA			INH	4D		3					
TSTX			INH	5D		3					
TST opr			IX1	6D	ff	5					
TST opr			IX	7D		4					
TXA	Transfer index register contents to accumulator	A ← (X)	INH	9F		2	–	–	–	–	–
WAIT	Enable interrupts; halt CPU		INH	8F		2	–	0	–	–	–

3.4.7 Opcode Map

Table 3-13 is an opcode map for the MC68HC05P1 instructions.

Table 3-13. 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	
HI / LO	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111	HI / LO
0 0000	5 BRSET0 3 DIR	5 BSET0 2 DIR	3 BRA 2 REL	5 NEG 2 DIR	3 NEGA 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 0000
1 0001	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 0001
2 0010	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 0010
3 0011	5 BRCLR1 3 DIR	5 BSCLR1 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 0011
4 0100	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 0100
5 0101	5 BRCLR2 3 DIR	5 BCLR2 2 DIR	3 BCS 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 0101
6 0110	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 0110
7 0111	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 0111
8 1000	5 BRSET4 3 DIR	5 BSET4 2 DIR	3 BHCC 2 REL	5 LSL 2 DIR	3 LSLA 1 INH	3 LSLX 1 INH	6 LSL 2 IX1	5 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 1000
9 1001	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	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 1001
A 1010	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 INH	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 1010
B 1011	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 1011
C 1100	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 INH	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 1100
D 1101	5 BRCLR6 3 DIR	5 BCLR6 2 DIR	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	6 TST 2 INH	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 1101
E 1110	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 1110
F 1111	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 INH	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 1111

ABBREVIATIONS FOR ADDRESSING MODES

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

LEGEND

	F 1111		High Byte of Opcode in Hexadecimal / High Byte of Opcode in Binary
Number of Cycles / Opcode Mnemonic / Number of Bytes/Addressing Mode	3 SUB 1 IX	0 0000	Low Byte of Opcode in Hexadecimal / Low Byte of Opcode in Binary

3.5 Low-Power Modes

The following paragraphs describe the STOP and WAIT modes. (Refer also to **5.2 Data-Retention Mode.**)

3.5.1 STOP Mode

The STOP instruction places the MCU in its lowest power-consumption mode. In STOP mode, the internal oscillator turns off, halting all internal processing, including capture/compare timer operation and computer operating properly (COP) timer operation. (Refer to Figure 3-8.)

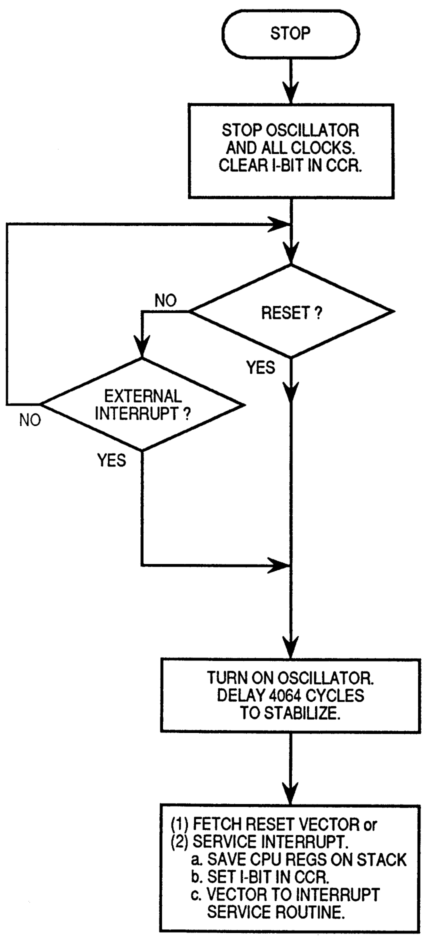


Figure 3-8. STOP Function Flowchart

During STOP mode, the input capture interrupt enable bit (ICIE), the output compare interrupt enable bit (OCIE), and the timer overflow interrupt enable bit (TOIE) in the timer control register (TCR) are cleared to remove any pending timer interrupt requests and to disable any further timer interrupts. The interrupt mask (I bit) in the condition code register is cleared to enable external interrupts. All other registers and memory locations remain unchanged. All I/O lines remain unchanged. The MCU can be brought out of STOP mode only by an external interrupt or a reset. An external interrupt automatically loads the program counter with the contents of $1FFA and $1FFB, the locations of the vector address of the interrupt service routine. A reset automatically loads the program counter with the contents of $1FFE and $1FFF, the locations of the vector address of the reset service routine.

3.5.2 WAIT Mode

The WAIT instruction places the MCU in an intermediate power-consumption mode. All CPU action stops, but the capture/compare timer remains active. If the A/D converter is enabled, it is also active in WAIT mode. An interrupt from the capture/compare timer can cause the MCU to exit WAIT mode. (Refer to Figure 3-9.)

The COP timer is not disabled in WAIT mode. To prevent a timer reset, exit from WAIT and reset the COP timer before timeout.

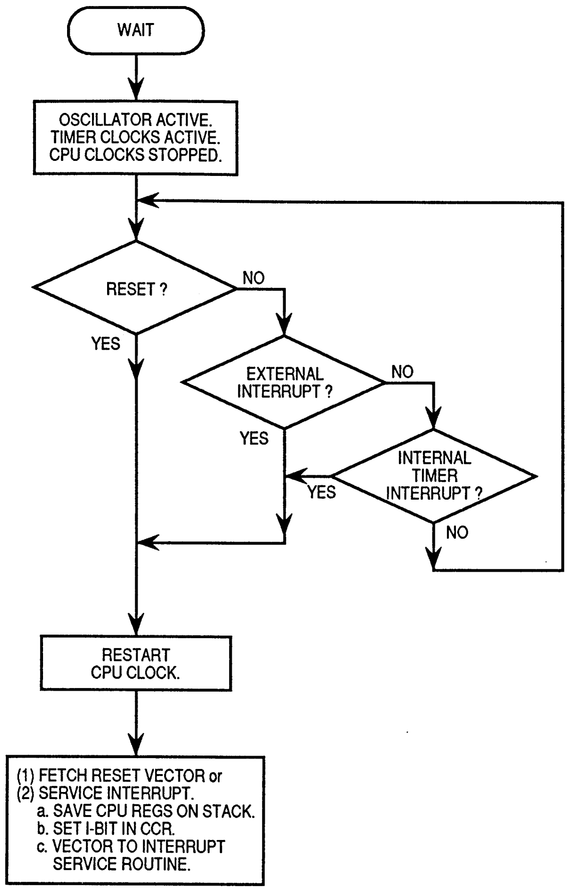


Figure 3-9. WAIT Function Flowchart

During WAIT mode, the interrupt mask (I bit) in the condition code register is cleared to enable interrupts. All other registers, memory locations, and I/O lines remain in their previous states.

SECTION 4
RESETS AND INTERRUPTS

This section describes CPU resets and interrupts.

4.1 Resets

A reset immediately stops execution of the current instruction. A reset forces the internal address bus to a known starting address and forces certain control and status bits to known conditions. The CPU can be reset in the following ways:

- Initial power-up (power-on reset)
- An external, logical zero signal on the reset pin ($\overline{\text{RESET}}$)

NOTE

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

The following internal actions occur as a result of a reset:

- All implemented data direction register bits are cleared, so the corresponding I/O pins become high-impedance inputs.
- The stack pointer is loaded with $FF.
- The interrupt mask (I-bit) is set, inhibiting interrupts, and the $\overline{\text{IRQ}}$ request latch is cleared.
- The capture/compare timer clock divider stages are cleared. The capture/compare timer is loaded with $FFFC. The output compare bit (TCMP) and the output level bit (OLVL) are cleared. All capture/compare timer interrupt enable bits (ICIE, OCIE, and TOIE) are cleared to disable timer interrupts.
- The STOP latch is cleared to enable MCU clocks.
- The WAIT latch is cleared to wake the CPU from the WAIT mode.
- The program counter is loaded with the user-defined reset vector address; the high byte of the program counter is loaded with the contents of location $1FFE, and the low byte of the program counter is loaded from location $1FFF.

4.1.1 Power-On Reset (POR)

A reset is generated on power-up when a positive transition occurs on V_{DD}. The POR is strictly for power-up conditions. It cannot be used to detect a drop in the power supply voltage.

To allow the clock generator to stabilize, there is a 4064 t_{cyc} (internal clock cycle) delay after the oscillator becomes active. If the $\overline{RESET}$ pin is at logical zero at the end of 4064 t_{cyc}, the CPU remains in the reset condition until $\overline{RESET}$ goes to logical one.

4.1.2 External $\overline{RESET}$ Input

The CPU is reset when a logical zero is applied to the $\overline{RESET}$ input for a period of one and one-half internal clock cycles (t_{cyc}). The $\overline{RESET}$ input consists of a Schmitt trigger that senses the logic level at the $\overline{RESET}$ pin.

$\overline{RESET}$ is an input-only pin and does not become active (go to logical zero) when a power-on reset occurs.

4.2 Interrupts

An interrupt temporarily stops normal processing so that some unusual event can be processed. Unlike a reset, an interrupt does not stop the current instruction. An interrupt is considered pending until the current instruction is complete. There are three types of interrupts:

- External interrupt — If the interrupt mask is a logical zero, and the external interrupt pin ($\overline{IRQ}$) goes to logical zero, then the CPU recognizes an external interrupt.
- Capture/compare timer interrupt — When the interrupt mask is a logical zero, the CPU can recognize interrupts from the capture/compare timer. If one of the three timer interrupt flags (ICF, OCF, TOF) in the timer status register goes to logical one, and its corresponding interrupt enable bit (ICIE, OCIE, TOIE) in the timer control register is a logical one, a timer interrupt is requested.
- Software interrupt — The software interrupt is an instruction executed regardless of the state of the interrupt mask.

The following internal actions occur as a result of an interrupt:

- CPU register contents are stored on the stack in the order PCL, PCH, X, A, CCR.
- The interrupt mask is automatically set to prevent additional interrupts.

- An interrupt vector that causes processing to continue at the starting address of the interrupt routine is fetched.
- The RTI (return from interrupt) instruction causes the register contents to be recovered from the stack in the order CCR, A, X, PCH, PCL. Normal processing resumes.

Figure 4-1 shows the stacking and recovery sequence.

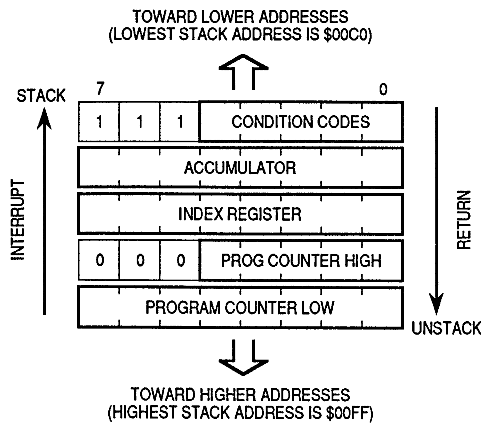


Figure 4-1. Interrupt Stacking Order

As each instruction is completed, the CPU checks for the presence of enabled external interrupt requests and enabled timer interrupt requests. For an external interrupt request to be recognized, the interrupt mask (I-bit) in the condition code register (CCR) must be a logical zero. If the interrupt mask is set or if no qualified interrupt request is pending, the processor fetches and executes the next program instruction.

Recognition of a timer interrupt request requires two conditions: the interrupt mask must be a logical zero, and the corresponding interrupt enable bit (OCIE, ICIE, or TOIE) in the timer control register (TCR) must be a logical one. If the interrupt mask is set, or if no qualified interrupt request is pending, the processor fetches and executes the next program instruction.

If both an external interrupt and a capture/compare timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first.

A software interrupt (SWI) is executed as an instruction, regardless of the state of the interrupt mask. Figure 4-2 shows how interrupts relate to normal instruction execution. CPU control logic determines the sequence of operations.

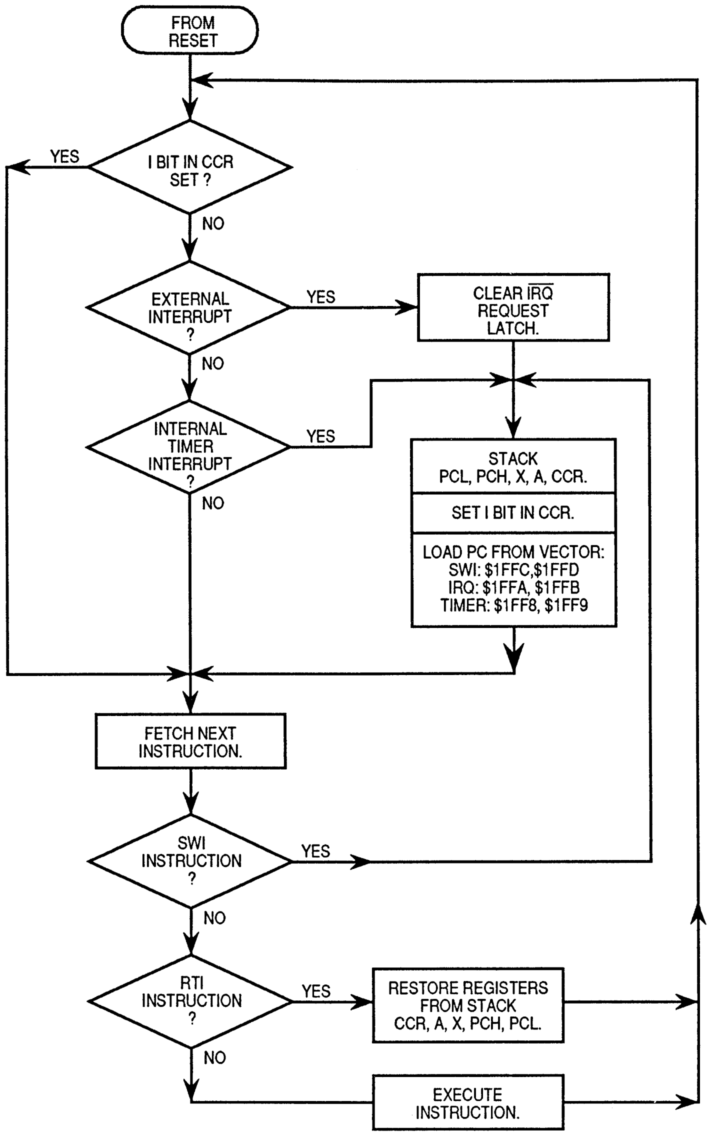


Figure 4-2. Reset and Interrupt Flowchart

4.2.1 External Interrupt

The CPU recognizes an external interrupt when the external interrupt pin ($\overline{IRQ}$) goes to a logical zero while the interrupt mask (I-bit) is at logical zero. At this time, a small synchronization delay occurs, and a logical one is latched internally to signify that an external interrupt has been requested. When the CPU completes its current instruction, the interrupt latch is tested. If the interrupt latch contains a logical one, and the interrupt mask is a logical zero, the CPU then begins the interrupt sequence. The current state of the CPU is pushed onto the stack, and the interrupt mask is set to inhibit further interrupts until the present one is serviced. The address of the interrupt service routine is contained in memory locations $1FFA and $1FFB.

An edge sensitive and level sensitive external interrupt trigger, and an edge sensitive only external interrupt trigger are available as factory-set mask options. Figure 4-3 shows the internal logic of with this mask option. The interrupt latch is cleared while the interrupt vector is being fetched. During the interrupt service routine, a new external interrupt request can be initiated and latched. As soon as the interrupt mask is cleared (usually during the return from interrupt), the latched request is recognized and serviced.

The level sensitive trigger option allows multiple interrupt sources to be wire-ORed to the $\overline{IRQ}$ pin. As long as any source is holding the $\overline{IRQ}$ pin at logical zero, an external interrupt request is considered to be pending.

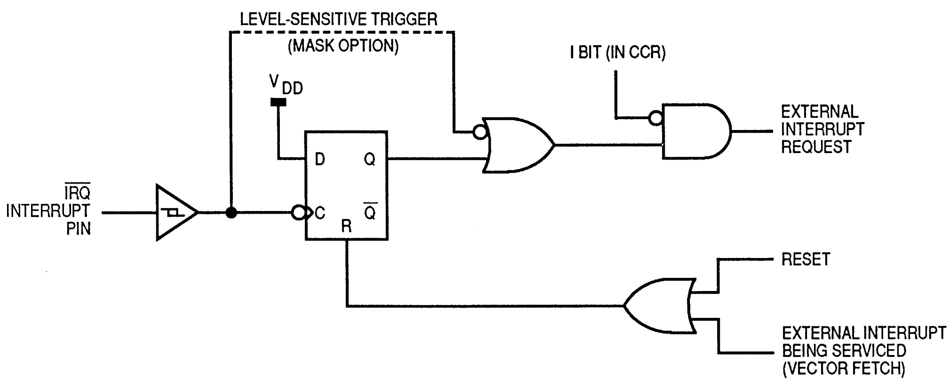


Figure 4-3. External Interrupt Logic

4.2.2 Software Interrupt (SWI)

The SWI is an executable instruction. The SWI instruction is executed regardless of the state of the interrupt mask (I-bit) in the condition code register. The address of the SWI interrupt service routine is in memory locations $1FFC and $1FFD.

4.2.3 Capture/Compare Timer Interrupt

The capture/compare timer can generate three interrupts when the interrupt mask is a logical zero. When one of the three timer interrupt flags in the timer status register is at logical one, and the corresponding interrupt enable flag in the timer control register is at logical one, the CPU recognizes a timer interrupt. (Refer to **SECTION 6 CAPTURE/COMPARE TIMER** for more information.) All three timer interrupts use the same interrupt vector at $1FF8 and $1FF9.

SECTION 5
MEMORY

Section 5 describes the organization of the on-chip memory. The CPU of the MC68HC05P1 MCU can address 8K bytes of memory space.

5.1 Memory Map

The program counter normally advances one address at a time through the on-chip memory, reading the instructions and data necessary to execute the program. The ROM portion of memory holds the program instructions, user-defined vectors, and service routines. The RAM portion of memory holds variable data. I/O, control, and status registers are memory-mapped so that the CPU can access their locations the same way it accesses any other memory location.

On-chip ROM includes 2112 bytes of factory-programmed memory for storage of application program instructions and fixed data. The last eight ROM addresses ($1FF8–$1FFF) contain user-defined vectors for servicing interrupts and resets. When ordering the MCU, the user specifies the instructions and data to be programmed into the user ROM.

The 240 bytes between $1F00 and $1FEF are reserved ROM addresses that contain the instructions for a series of self-check tests.

The MCU has 128 bytes of fully static read-write memory for storage of variable and temporary data during program execution. The CPU uses the top 64 RAM addresses ($00C0–$00FF) for the stack. The CPU uses the stack to save CPU register contents before processing an interrupt or subroutine call. The stack pointer decrements during pushes and increments during pulls.

The first 32 bytes of the memory space contain port data registers, port data direction registers, and timer control, status, and counter registers.

Figure 5-1 is a memory map of the MCU. Refer to Figure 5-2 for a more detailed memory map of the 32-byte I/O register area.

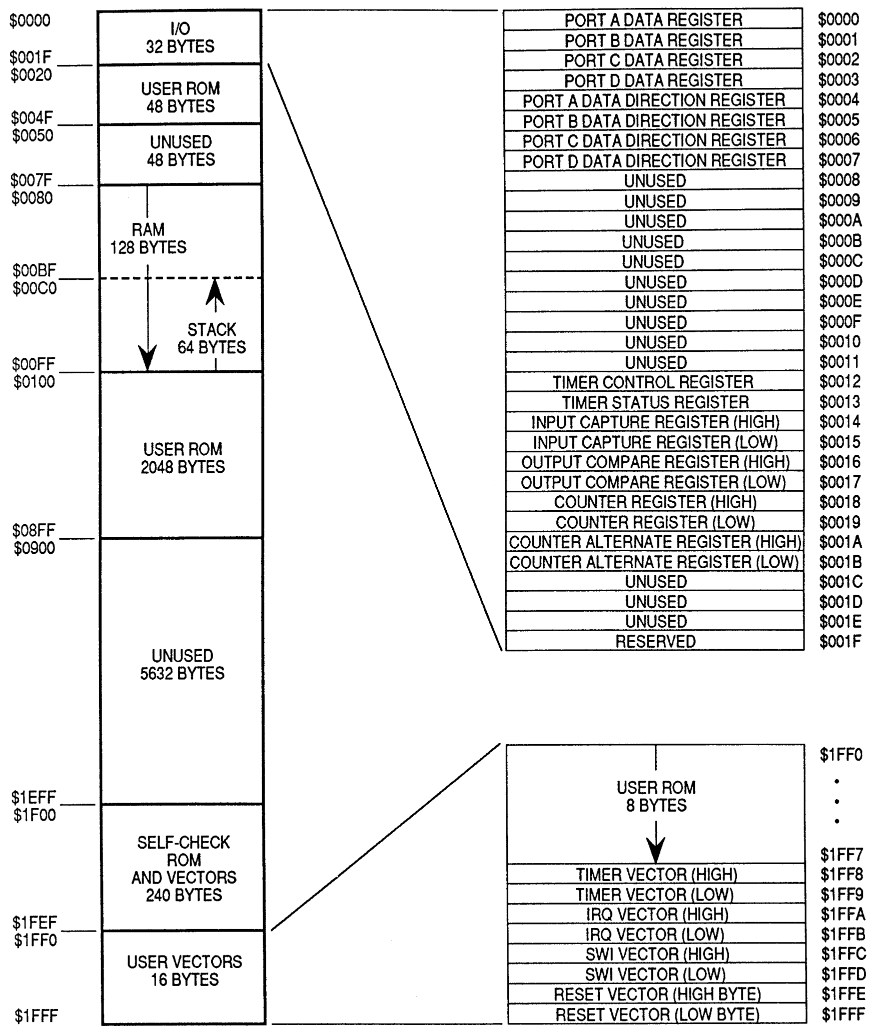


Figure 5-1. MC68HC05P1 Memory Map

NOTE

Using the stack area for data storage or as a temporary work area requires care to prevent data from being overwritten during stacking from an interrupt or subroutine call.

Address	Bit 7	6	5	4	3	2	1	Bit 0	Register
$0000	I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O	PORTA
	3	4	5	6	7	8	9	10	PORT A PIN NUMBERS (REF.)
	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORT A PIN NAMES (REF.)
$0001	I/O	I/O	I/O	0	0	0	0	0	PORTB
	13	12	11	–	–	–	–	–	PORT B PIN NUMBERS (REF.)
	PB7	PB6	PB5	–	–	–	–	–	PORT B PIN NAMES (REF.)
$0002	I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O	PORTC
	15	16	17	18	19	20	21	22	PORT C PIN NUMBERS (REF.)
	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORT C PIN NAMES (REF.)
$0003	I only	0	I/O	1	0	0	0	0	PORTD
	25	–	23	–	–	–	–	–	PORT D PIN NUMBERS (REF.)
	PD7/TCAP	–	PD5	–	–	–	–	–	PORT D PIN NAMES (REF.)
$0004	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0	DDRA
$0005	DDRB7	DDRB6	DDRB5	1	1	1	1	1	DDRB
$0006	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0	DDRC
$0007	0	0	DDRD5	0	0	0	0	0	DDRD
$0008									Unused
$0009									Unused
$000A									Unused
$000B									Unused
$000C									Unused
$000D									Unused
$000E									Unused
$000F									Unused
$0010									Unused
$0011									Unused
$0012	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL	TCR
$0013	ICF	OCF	TOF	0	0	0	0	0	TSR
$0014	Bit 15							Bit 8	TCAP (HIGH)
$0015	Bit 7							Bit 0	TCAP (LOW)
$0016	Bit 15							Bit 8	TCMP (HIGH)
$0017	Bit 7							Bit 0	TCMP (LOW)
$0018	Bit 15							Bit 8	TCNT (HIGH)
$0019	Bit 7							Bit 0	TCNT (LOW)
$001A	Bit 15							Bit 8	ALTCNT (HIGH)
$001B	Bit 7							Bit 0	ALTCNT (LOW)
$001C									Unused
$001D									Unused
$001E									Unused
$001F									RESERVED
	Bit 7	6	5	4	3	2	1	Bit 0	

Figure 5-2. I/O and Control Register Summary

5.2 Data-Retention Mode

In data-retention mode, the MCU retains RAM contents and CPU register contents at V_{DD} voltages as low as 2.0 Vdc. Before the V_{DD} voltage is lowered, drive the $\overline{RESET}$ pin to logical zero. During data-retention mode, $\overline{RESET}$ must remain low continuously. The data-retention mode allows the MCU to be left in a low power-consumption mode during which data is held, but the CPU cannot execute instructions. To exit the data-retention mode, V_{DD} must be returned to its normal operating voltage before $\overline{RESET}$ is returned to logical one.

SECTION 6
CAPTURE/COMPARE TIMER

This section describes the operation of the capture/compare timer. Figure 6-1 shows the structure of the capture/compare timer system.

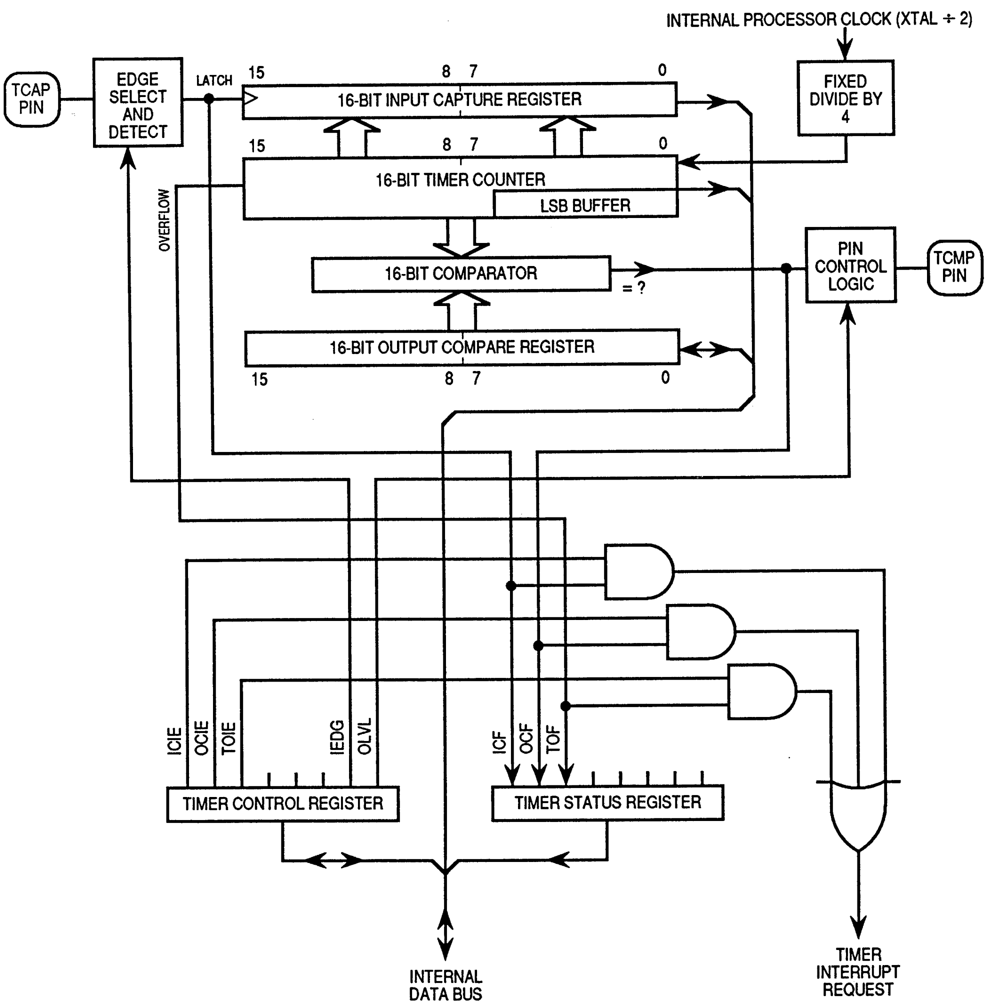


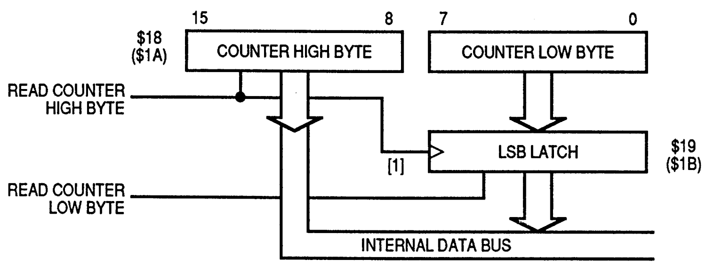
Figure 6-1. Capture/Compare Timer Block Diagram

6.1 Timer Counter

The key element in the programmable capture/compare timer is a 16-bit, free-running counter, preceded by a prescaler that divides the internal clock by four. The counter provides the timing reference for the input capture and output compare functions. Software can read the counter value at any time from either of the following two registers with no effect on the counter sequence:

- Timer Counter Register (TCNT)
- Alternate Counter Register (ALTCNT)

Reading the high byte of the timer counter register or alternate counter register accesses the high byte value at the time of the read and causes the low byte to be latched into a buffer. (Refer to Figure 6-2.) This buffer value remains fixed after the first high-byte read, even if the high byte is read more than once. The buffer is accessed when the read sequence is completed by reading the low byte of the timer counter register or the alternate counter register. If the high byte is read, the low byte must also be read to complete the read sequence.



[1] The LSB latch is normally transparent, becomes latched when high byte of counter is read, and becomes transparent again when low byte of counter is read.

Figure 6-2. 16-Bit Counter Reads

The free-running counter is preset to $FFFC during reset. During a power-on reset, the counter is preset to $FFFC and begins running after the oscillator startup delay. Because the free-running counter is 16 bits long and preceded by a fixed divide-by-four prescaler, the value in the counter repeats every 262,144 internal clock cycles.

6.1.1 Timer Counter Register (TCNT)

The high and low bytes of the free-running counter can be read from the timer counter register at locations $0018 and $0019. (Refer to Figure 6-3.) Reading the low byte ($0018) of the timer counter register after reading the timer status register clears the timer overflow flag (TOF).

Bit 7	6	5	4	3	2	1	Bit 0		
Bit 15	14	13	12	11	10	9	Bit 8	$18	TCNT (HIGH)
Bit 7	6	5	4	3	2	1	Bit 0	$19	TCNT (LOW)
RESET TO $FFFC								RESET CONDITION	

Figure 6-3. Timer Counter Register (TCNT)

6.1.2 Alternate Counter Register (ALTCNT)

The high and low bytes of the free-running counter can be read from the alternate counter register at locations $001A and $001B. (Refer to Figure 6-4.) Reading the alternate counter register does not affect the timer overflow flag (TOF). The alternate counter register can be read at any time without risk of clearing TOF inadvertently. Normally, the timer value is read from the alternate counter register unless the read sequence is intended to clear TOF.

Bit 7	6	5	4	3	2	1	Bit 0		
Bit 15	14	13	12	11	10	9	Bit 8	$1A	ALTCNT (HIGH)
Bit 7	6	5	4	3	2	1	Bit 0	$1B	ALTCNT (LOW)
RESET TO $FFFC								RESET CONDITION	

Figure 6-4. Alternate Counter Register (ALTCNT)

NOTE

To prevent interrupts from occurring between readings of the high and low bytes of the timer counter register or the alternate counter register, the interrupt mask (I-bit) in the condition code register can be set before reading the high byte and cleared after reading the low byte.

6.2 Timer Functions

The input capture and output compare functions provide a means to latch the times at which external events occur, to measure input waveforms, and to generate output waveforms and timing delays.

6.2.1 Input Capture

The input capture feature provides a means to record the time at which an external event occurs. When the timer detects a selected (negative-going or positive-going) edge on the TCAP pin, it latches the contents of the timer counter register into the input capture register. The IEDG bit in the timer control register allows software to select the edge polarity that triggers the input capture function. The ICIE bit in the timer control register allows software to determine whether or not the input capture function generates a hardware interrupt request. (Refer to Figure 6-5.)

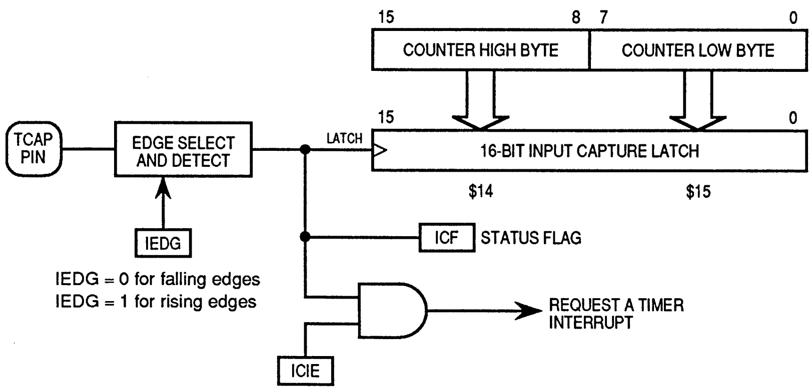


Figure 6-5. Input Capture Operation

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

6.2.2 Output Compare

The output compare feature provides a means of generating an output signal when the timer counter register reaches a selected value. The selected value is written into the output compare register. On every fourth internal clock cycle the capture/compare timer compares the value of the timer counter register to the contents of the output compare register. When a match occurs, the timer transfers the output level bit (OLVL) from the timer control register to the TCMP output pin. (Refer to Figure 6-6.)

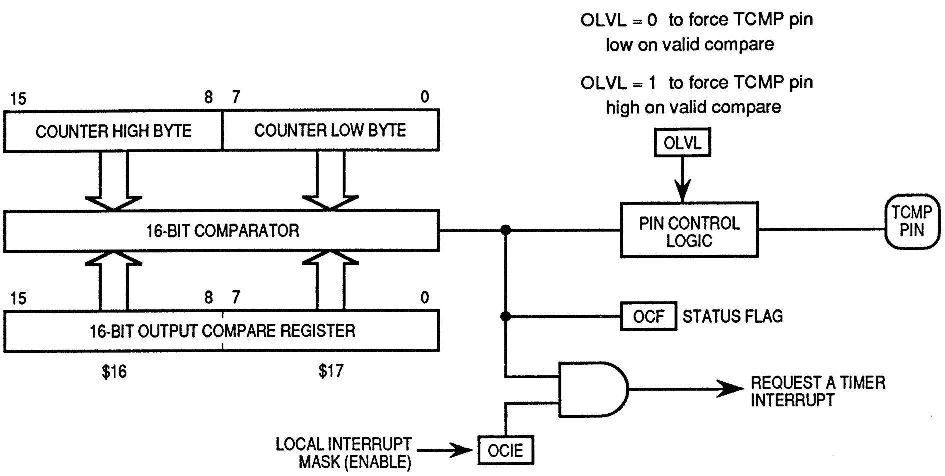


Figure 6-6. Output Compare Operation

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

6.3 Input Capture Register (ICR)

The high and low bytes of the input capture register are at memory locations $0014 and $0015. (Refer to Figure 6-7.) The input capture register is read-only and is not affected by a reset.

Bit 7	6	5	4	3	2	1	Bit 0	
Bit 15	14	13	12	11	10	9	Bit 8	$14 TCAP (HIGH)
Bit 7	6	5	4	3	2	1	Bit 0	$15 TCAP (LOW)
INPUT CAPTURE REGISTER NOT AFFECTED BY RESET								RESET CONDITION

Figure 6-7. Input Capture Register (ICR)

When the input capture edge detector senses a defined transition on the TCAP pin, the input capture flag (ICF) is set, and the input capture register latches the value of the timer counter register. The contents of the timer counter register are transferred to the input capture register on every defined signal transition whether or not ICF was previously set. The input capture register always contains the value in the timer counter register at the time of the most recent input capture. The polarity of the level transition that triggers the counter capture is defined by the input edge bit (IEDG).

The timer counter register increments every fourth cycle of the internal clock. The counter value latched into the input capture register is one count more than the count at the time of the last rising edge of the clock before the defined transition on the TCAP pin occurred. This delay is required for internal synchronization.

Reading the high byte of the input capture register inhibits the input capture function until the low byte is also read. If the high byte is read first, both bytes must be read. Reading only the low byte does not inhibit the input capture function.

NOTE

To prevent interrupts from occurring between readings of the high and low bytes of the input capture register, set the interrupt flag before reading the high byte and clear the flag after reading the low byte.

6.4 Output Compare Register (OCR)

The high and low bytes of the output compare register are at memory locations $0016 and $0017. (Refer to Figure 6-8.) All bits are readable and writable and are not altered by the capture/compare timer hardware or by a reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

Bit 7	6	5	4	3	2	1	Bit 0	
Bit 15	14	13	12	11	10	9	Bit 8	$0016 OCR (HIGH)
Bit 7	6	5	4	3	2	1	Bit 0	$0017 OCR (LOW)
OUTPUT COMPARE REGISTER NOT AFFECTED BY RESET								RESET CONDITION

Figure 6-8. Output Compare Register (OCR)

Output compare register contents are continually compared with the contents of the timer counter register. When a match occurs, the output compare flag (OCF) is set, and the OLVL bit is clocked to the TCMP output pin. OLVL appears on TCMP whether or not OCF was previously set. An output compare interrupt is enabled if the output compare interrupt enable bit (OCIE) is set. The output compare register values and the output level bit are typically changed after each successful comparison to establish a new timeout period. Writing to either byte of the output compare register does not affect the other byte.

Writing the high byte of the output compare register inhibits the output compare function until the low byte is also written. If the high byte is written first, both bytes must be written. Writing only the low byte does not inhibit the output compare function.

6.5 Timer Status Register (TSR)

The read-only timer status register shown in Figure 6-9 has status flags to indicate the following conditions:

- A selected transition occurred at the TCAP pin, and the contents of the timer counter register were transferred to the input capture register.
- A match occurred between the timer counter register and the output compare register, and the OLVL bit was transferred to the TCMP pin.
- A timer counter register transition from $FFFF to $0000 occurred.

Bit 7	6	5	4	3	2	1	Bit 0	
ICF	OCF	TOF	0	0	0	0	0	$0013 TSR
U	U	U	0	0	0	0	0	RESET CONDITION

(U = UNAFFECTED)

Figure 6-9. Timer Status Register (TSR)

ICF — Input Capture Flag

ICF is automatically set when an edge of the selected polarity occurs on TCAP. Clear ICF by reading the timer status register with ICF set, and then reading the low byte ($0015) of the input capture register.

OCF — Output Compare Flag

OCF is automatically set when the value of the timer counter register matches the contents of the output compare register. Clear OCF by reading the timer status register with OCF set, and then accessing the low byte ($0017) of the output compare register.

TOF — Timer Overflow Flag

TOF is automatically set when the timer counter register changes from $FFFF to $0000. Clear TOF by reading the timer status register with TOF set, and then accessing the low byte ($0019) of the timer counter register.

Bits 4–0 — Not used; always read zero

To clear a status bit, read the timer status register. Then access the low byte of the register associated with the status bit.

When using the timer overflow function and reading the timer counter register at random times to measure elapsed time, TOF could unintentionally be cleared. This problem can occur when reading the timer status register and the low byte of the timer counter register, but not for the purpose of servicing the TOF flag.

The alternate counter register at locations $001A and $001B contains the same value as the timer counter register at locations $0018 and $0019. Because reading the alternate counter register has no effect on the timer status register, the alternate counter register can be read at any time without clearing TOF.

6.6 Timer Control Register (TCR)

The read/write timer control register has five control bits. (Refer to Figure 6-10.) Three bits control interrupts associated with the timer status register flags ICF, OCF, and TOF. Another bit determines the edge polarity (positive-going or negative-going) that activates the input capture edge detector. Another bit determines the output level clocked onto TCMP when a successful output compare occurs.

	Bit 7	6	5	4	3	2	1	Bit 0	
	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL	$0012 TCR
	0	0	0	0	0	0	U	0	RESET CONDITION

(U = UNAFFECTED)

Figure 6-10. Timer Control Register (TCR)

ICIE — Input Capture Interrupt Enable

1 = ICF interrupt enabled

0 = ICF interrupt disabled

OCIE — Output Compare Interrupt Enable
1 = OCF interrupt enabled
0 = OCF interrupt disabled

TOIE — Timer Overflow Interrupt Enable
1 = TOF interrupt enabled
0 = TOF interrupt disabled

IEDG — Input Edge
This bit determines which transition on the TCAP pin triggers a transfer of the contents of the timer counter register to the input capture register.
1 = Positive edge (low level to high level)
0 = Negative edge (high level to low level)

OLVL — Output Level
This bit determines the output level on the TCMP pin when a successful output compare occurs.
1 = High output
0 = Low output

Bits 4, 3, and 2 — Not used; always read zero

6.7 Timer during WAIT Mode

The internal clock halts during WAIT mode, but the capture/compare timer and COP counter remain active. An interrupt from the capture/compare timer causes the processor to exit WAIT mode.

6.8 Timer during STOP Mode

In STOP mode, the capture/compare timer stops counting and holds the last count value. If $\overline{IRQ}/V_{PP}$ is used to exit STOP mode, the timer resumes counting from the count value that was present when STOP mode was entered. If $\overline{RESET}$ is used, the counter is forced to $FFFC.

If a defined transition occurs on the TCAP pin during STOP mode, ICF goes high as soon as an external interrupt brings the MCU out of STOP mode. If a power-on reset or a logical zero on the $\overline{RESET}$ pin brings the MCU out of STOP mode, all timer interrupt enable bits are cleared.

SECTION 7
SELF-CHECK MODE

The self-check mode described in this section tests the operation of the MCU.

7.1 Self-Check Circuit

The self-check function determines if the MCU is functioning properly. The self-check circuit is shown in Figure 7-1. If 9 Vdc is applied to the $\overline{IRQ}$ pin, and a logical one is applied to the PD7/TCAP pin, the MCU enters the self-check mode when reset. Port C pins PC3–PC0 are monitored for the self-check results. After a reset in self-check mode, the following self-check tests are performed automatically in self-check mode:

- I/O — Functional test of ports A, B, and C
- RAM — Counter test for each RAM byte
- ROM — Checksum of entire ROM pattern
- Capture/compare timer — Test of counter register and OCF bit
- Interrupts — Test of external and capture/compare timer interrupts

7.2 Self-Check Results

Table 7-1 lists the codes displayed by the light-emitting diodes (LEDs) to indicate the self-check results.

Table 7-1. Self-Check Results

PC3	PC2	PC1	PC0	Remarks
1	0	0	1	Bad I/O
1	0	1	0	Bad RAM
1	0	1	1	Bad Capture/Compare Timer
1	1	0	0	Bad ROM
1	1	0	1	Bad Interrupts or $\overline{IRQ}$ Request
Flashing				Good Device
All Others				Bad Device, Bad Port C, etc.

NOTE: Zero indicates LED is on; 1 indicates LED is off.

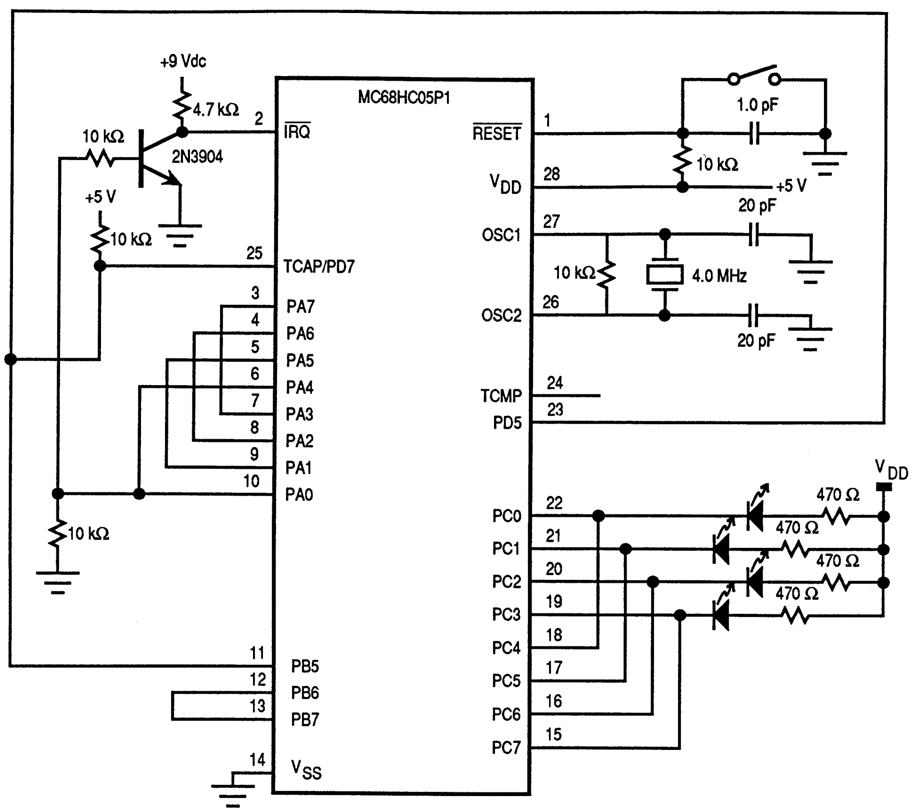


Figure 7-1. Self-Check Circuit

SECTION 8
ELECTRICAL SPECIFICATIONS

This section contains MCU electrical specifications and timing information.

8.1 Maximum Ratings

The MCU contains circuitry that protects the inputs against damage from high static voltages; however, take precautions to avoid applying voltages higher than those shown in Table 8-1. V_{in} and V_{out} should be kept within the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$. Connect unused inputs to an appropriate logical voltage level (e.g., either V_{SS} or V_{DD}).

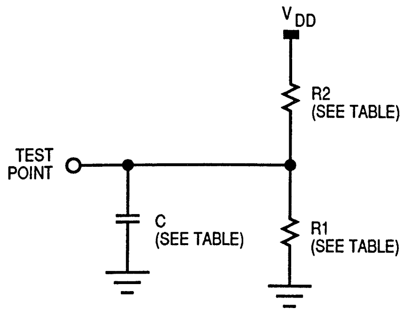
Table 8-1. Maximum Ratings

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	–0.3 to +7.0	V
Input Voltage	V_{in}	V_{SS} – 0.3 to V_{DD} + 0.3	V
Self-Check Mode ($\overline{IRQ}$ pin only)	V_{in}	V_{SS} – 0.3 to 2 × V_{DD} + 0.3	V
Current Drain per Pin (excluding V_{DD} and V_{SS})	I	25	mA
Operating Temperature Range	T_A	T_L to T_H	°C
MC68HC05P1P, DW		0 to +70	
MC68HC05P1CP, CDW		–40°C to +85°C	
MC68HC05P1VP		–40°C to +105°C	
Storage Temperature Range	T_{stg}	–65 to +150	°C

8.2 Thermal Characteristics

Table 8-2. Thermal Characteristics

Characteristic	Symbol	Value	Unit
Thermal Resistance	$R_{\theta JA}$		°C/W
Plastic		60	
SOIC		60	



V_{DD} = 4.5 V

Pins	R1	R2	C
PA7–PA0 PB5–PB0 PC7–PC0 PD5, TCMP	3.26 kΩ	2.38 kΩ	50 pF

V_{DD} = 3.0 V

Pins	R1	R2	C
PA7–PA0 PB7–PB5 PC7–PC0 PD5, TCMP	10.91 kΩ	6.32 kΩ	50 pF

Figure 8-1. Test Load

8.3 Power Considerations

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

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

where:

T_A = Ambient temperature in °C
$R_{\theta JA}$ = Package thermal resistance, junction to ambient in °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 (if $P_{I/O}$ is neglected):

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

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

$$K = P_D \times (T_A + 273°C) + R_{\theta JA} \times P_D \quad (3)$$

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.

8.4 DC Electrical Characteristics (V_{DD} = 5.0 Vdc)

Table 8-3. DC Electrical Characteristics (V_{DD} = 5.0 Vdc)

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage (I_{Load} ≤ 10.0 µA)	V_{OL} V_{OH}	– V_{DD} – 0.1	– –	0.1 –	V
Output High Voltage (I_{Load} = 0.8 mA) PA7–PA0, PB7–PB5, PC7–PC0, PD5, TCMP	V_{OH}	V_{DD} – 0.8	–	–	V
Output Low Voltage (I_{Load} = 1.6 mA) PA7–PA0, PB7–PB5, PC7–PC0, PD5, TCMP	V_{OL}	–	–	0.4	V
Input High Voltage PA7–PA0, PB7–PB5, PC7–PC0, PD5, PD7/TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 × V_{DD}	–	V_{DD}	V
Input Low Voltage PA7–PA0, PB7–PB5, PC7–PC0, PD5, PD7/TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	–	0.2 × V_{DD}	V
Data-Retention Mode Supply Voltage (0 to 70°C)	V_{RM}	2	–	–	V
Supply Current (refer to NOTES) RUN WAIT STOP 25°C 0 to 70°C (standard)	I_{DD}	 – – – –	 3.5 1.6 2.0 –	 7.0 4.0 50 140	 mA mA µA µA
I/O Ports Hi-Z Leakage Current PA7–PA0, PB7–PB5, PC7–PC0, PD5	I_{IL}	–	–	±10	µA
Input Current $\overline{RESET}$,$\overline{IRQ}$, OSC1,PD5, PD7/TCAP	I_{in}	–	–	±1	µA
Capacitance Ports (as input or output) $\overline{RESET}$, $\overline{IRQ}$, PD5,PD7/TCAP	 C_{out} C_{in}	 – –	 – –	 12 8	 pF pF

NOTES:

1. Typical values at midpoint of voltage range, 25°C only.
2. RUN (operating) I_{DD}, WAIT I_{DD} measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
3. WAIT I_{DD}, STOP I_{DD}: all ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD}$ – 0.2 V.
4. STOP I_{DD} measured with OSC1 = V_{SS}.
5. Standard temperature range is 0 to 70°C.
6. WAIT I_{DD} is affected linearly by the OSC2 capacitance.

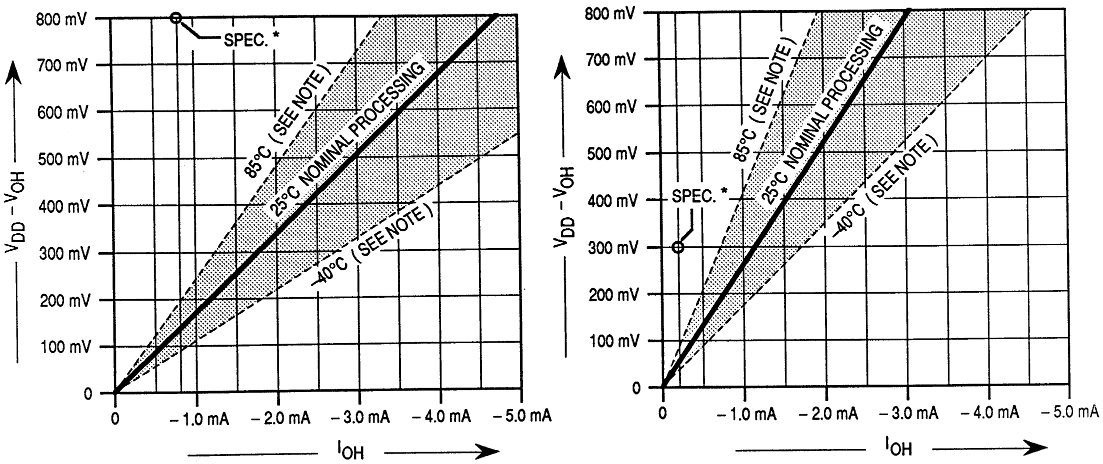
8.5 DC Electrical Characteristics (V_{DD} = 3.3 Vdc)

Table 8-4. DC Electrical Characteristics (V_{DD} = 3.3 Vdc)

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage (I_{Load} ≤ 10.0 µA)	V_{OL} V_{OH}	– $V_{DD}-0.1$	– –	0.1 –	V
Output High Voltage (I_{Load} = 0.2 mA) PA7–PA0, PB7–PB5, PC7–PC0, PD5, TCMP	V_{OH}	$V_{DD}-0.3$	–	–	V
Output Low Voltage (I_{Load} = 0.4 mA) PA7–PA0, PB7–PB5, PC7–PC0, PD5, TCMP	V_{OL}	–	–	0.3	V
Input High Voltage PA7–PA0, PB7–PB5, PC7–PC0, PD5, PD7/TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 × V_{DD}	–	V_{DD}	V
Input Low Voltage PA7–PA0, PB7–PB5, PC7–PC0, PD5, PD7/TCAP, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	–	0.2 × V_{DD}	V
Data-Retention Mode Supply Voltage (0 to 70°C)	V_{RM}	2.0	–	–	V
Supply Current (refer to NOTES) RUN WAIT STOP 25°C 0 to 70°C (Standard)	I_{DD}	 – – – –	 1.0 0.5 2.0 –	 2.5 1.4 30 80	 mA mA µA µA
I/O Ports Hi-Z Leakage Current PA7–PA0, PB7–PB5, PC7–PC0, PD5	I_{IL}	–	–	±10	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, OSC1, PD5, PD7/TCAP	I_{in}	–	–	±1	µA
Capacitance Ports (As Input or Output) $\overline{RESET}$, $\overline{IRQ}$, PD5, PD7/TCAP	 C_{out} C_{in}	 – –	 – –	 12 8	 pF pF

NOTES:
1. Typical values at midpoint of voltage range, 25°C only.
2. RUN (operating) I_{DD}, WAIT I_{DD} measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
3. WAIT I_{DD}, STOP I_{DD}: all ports configured as inputs, V_{IL} = 0.2 V, $V_{IH} = V_{DD} - 0.2$ V.
4. STOP I_{DD} measured with OSC1 = V_{SS}.
5. Standard temperature range is 0 to 70°C.
6. WAIT I_{DD} is affected linearly by the OSC2 capacitance.

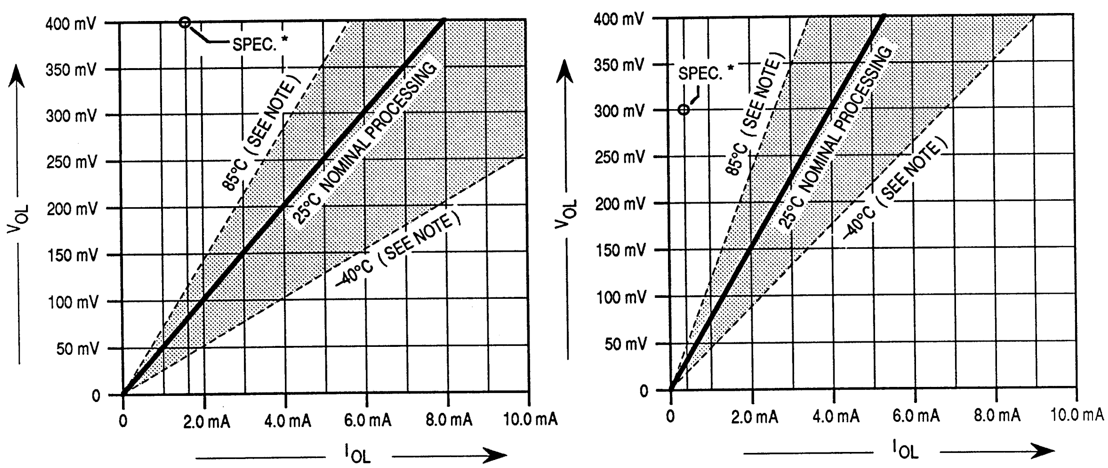


* At V_{DD} = 5.0 V, devices are specified and tested for (V_{DD} – V_{OH}) ≤ 800 mV @ I_{OH} = – 0.8 mA.

* At V_{DD} = 3.3 V, devices are specified and tested for (V_{DD} – V_{OH}) ≤ 300 mV @ I_{OH} = – 0.2 mA.

Shaded area indicates variation in driver characteristics caused by changes in temperature and for normal processing tolerances. Within the limited range of values shown, V vs. I curves are approximately straight lines.

Figure 8-2. Typical High-Side Driver Characteristics

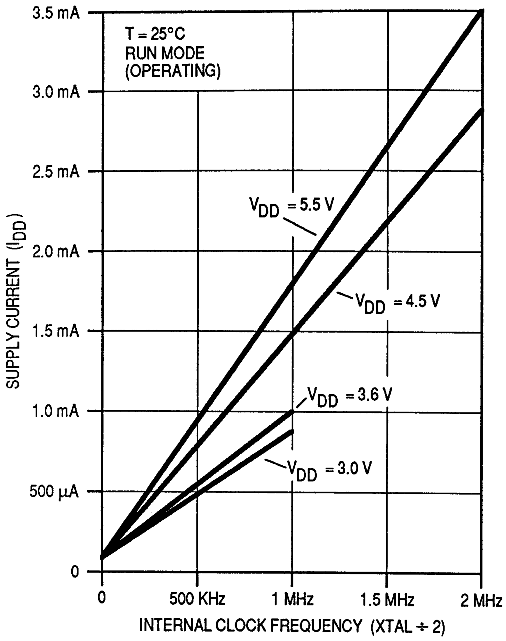


* At V_{DD} = 5.0 V, devices are specified and tested for V_{OL} ≤ 400 mV @ I_{OL} = 1.6 mA.

* At V_{DD} = 3.3 V, devices are specified and tested for V_{OL} ≤ 300 mV @ I_{OL} = 0.4 mA.

Shaded area indicates variation in driver characteristics caused by changes in temperature and for normal processing tolerances. Within the limited range of values shown, V vs. I curves are approximately straight lines.

Figure 8-3. Typical Low-Side Driver Characteristics



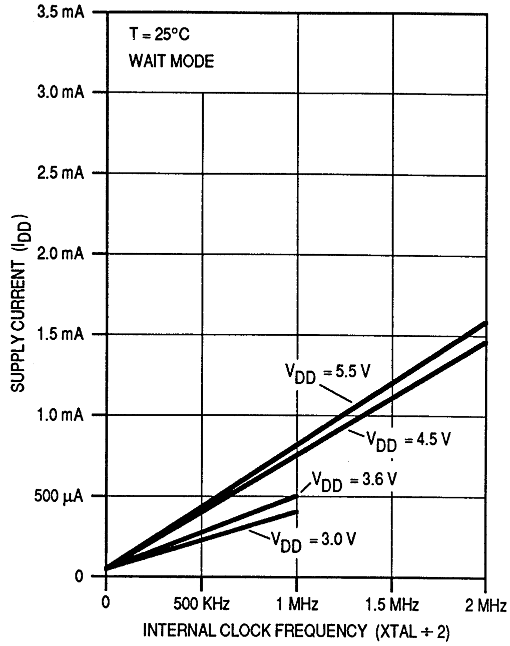
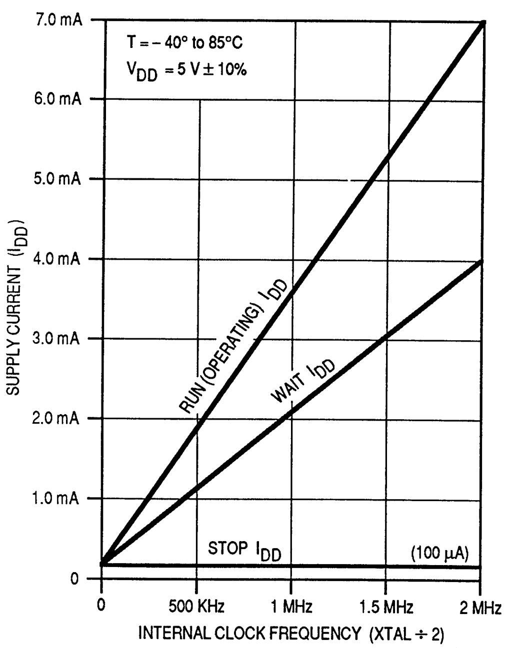


Figure 8-4. Typical Supply Current vs Clock Frequency



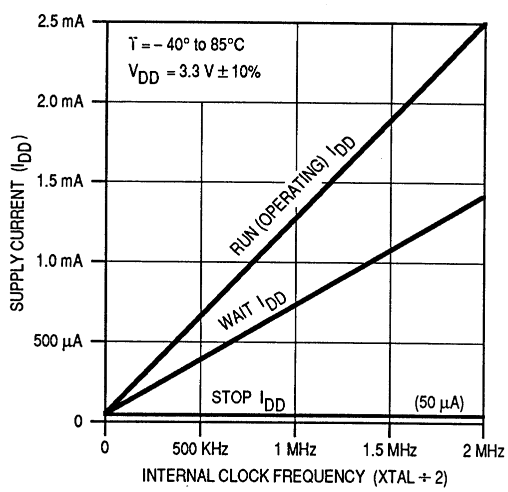


Figure 8-5. Maximum Supply Current vs Clock Frequency

8.6 Control Timing (V_{DD} = 5.0 Vdc)

Table 8-5. Control Timing (V_{DD} = 5.0 Vdc)

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Oscillator Frequency	f_{osc}			
Crystal Option		–	4.2	MHz
External Clock Option		dc	4.2	MHz
Internal Operating Frequency	f_{op}			
Crystal (f_{osc} ÷ 2)		–	2.1	MHz
External clock (f_{osc} ÷ 2)		dc	2.1	MHz
Internal clock cycle time	t_{cyc}	480	–	ns
$\overline{\text{RESET}}$ Pulse Width	t_{RL}	1.5	–	t_{cyc}
Capture/Compare Timer				
Resolution (refer to NOTE 1)	t_{RESL}	4.0	–	t_{cyc}
Input Capture Pulse Width	t_{TH}, t_{TL}	125	–	ns
Input Capture Pulse Period	t_{TLTL}	(refer to NOTE 2)	–	t_{cyc}
Interrupt Pulse Width Low (Edge Triggered)	t_{ILIH}	125	–	ns
Interrupt Pulse Period	t_{ILIL}	(refer to NOTE 3)	–	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	90	–	ns

NOTES:

1. Because a 2-bit prescaler in the capture/compare timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.
2. The period t_{TLTL} should not be less than the number of cycles it takes to execute the capture interrupt service routine plus 24 t_{cyc}.
3. The period t_{ILIL} should not be less than the number of cycles it takes to execute the interrupt service routine plus 21 t_{cyc}.

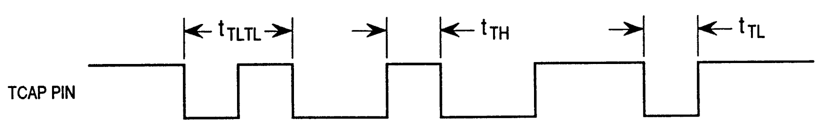
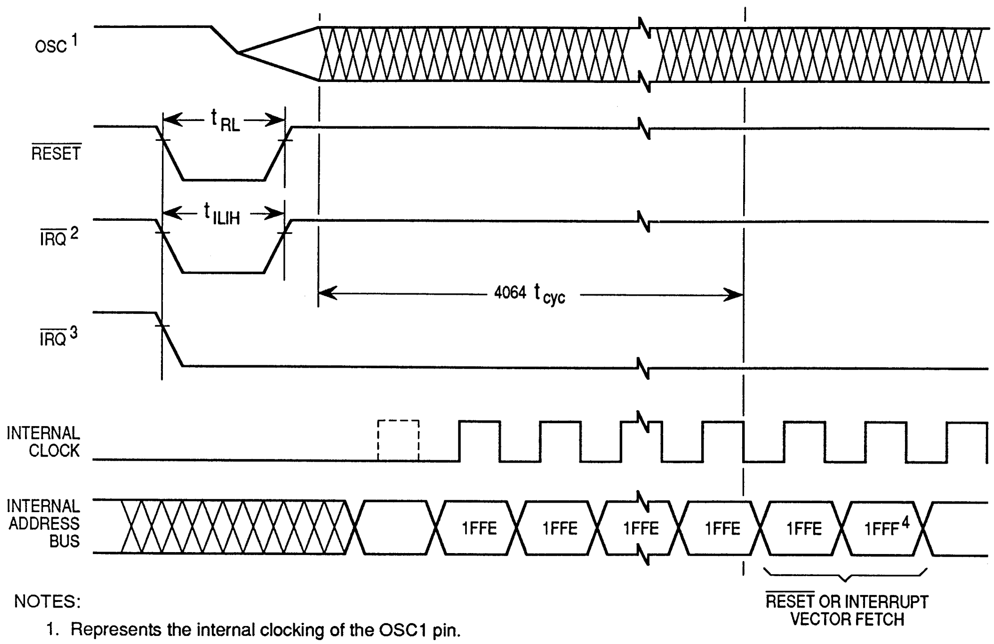


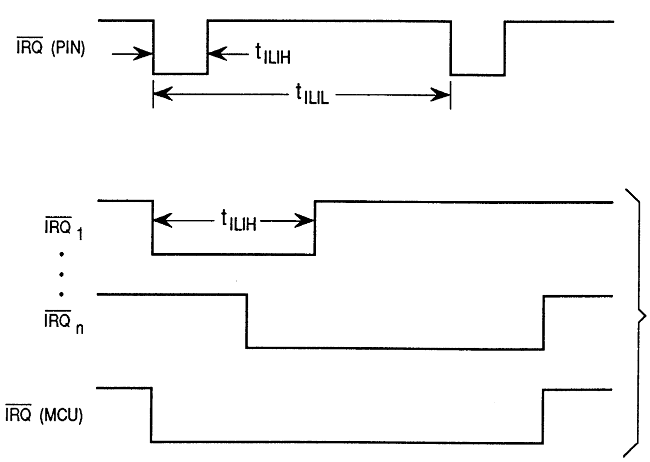
Figure 8-6. TCAP Timing



NOTES:

1. Represents the internal clocking of the OSC1 pin.
2. $\overline{IRQ}$ pin edge-sensitive mask option.
3. $\overline{IRQ}$ pin level- and edge-sensitive mask option.
4. $\overline{RESET}$ vector address shown for timing example.

Figure 8-7. STOP Recovery Timing



Edge-Sensitive Trigger Condition
The minimum pulse width (t_{ILIH}) is either 125 ns (V_{DD} = 5 V) or 250 ns (V_{DD} = 3 V). The period t_{ILIL} should not be less than the number of t_{cyc} cycles it takes to execute the interrupt service routine plus 19 t_{cyc} cycles.

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

NORMALLY USED WITH WIRE–ORed CONNECTION

Figure 8-8. External Interrupt Timing

8.7 Control Timing (V_{DD} = 3.3 Vdc)

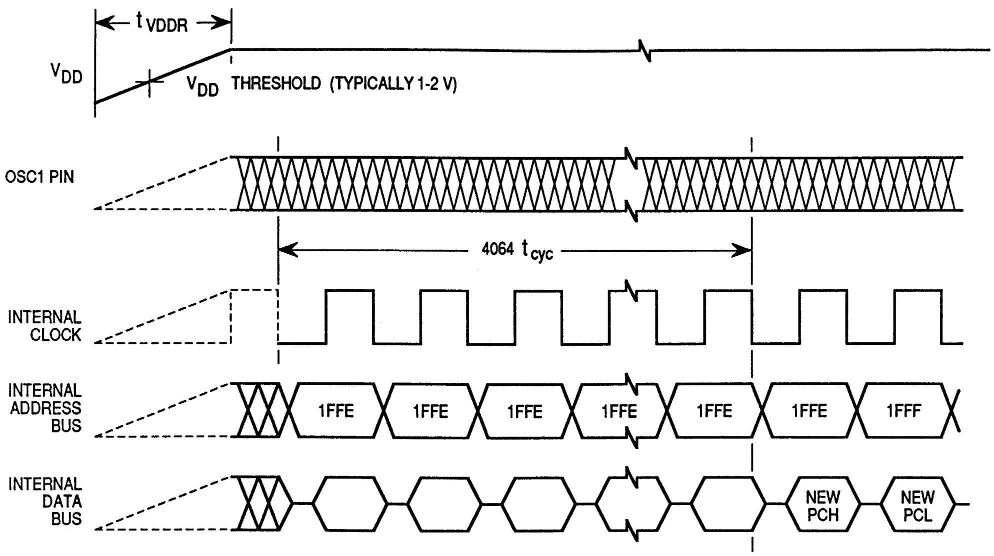
Table 8-6. Control Timing (V_{DD} = 3.3 Vdc)

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H.)

Characteristic	Symbol	Min	Max	Unit
Oscillator Frequency	f_{osc}			
Crystal Option		–	2.0	MHz
External Clock Option		dc	2.0	MHz
Internal Operating Frequency	f_{op}			
Crystal (f_{osc} ÷ 2)		–	1.0	MHz
External clock (f_{osc} ÷ 2)		dc	1.0	MHz
Cycle Time	t_{cyc}	1000	–	ns
STOP Recovery Startup Time (Crystal Oscillator)	t_{ILCH}	–	100	ms
$\overline{\text{RESET}}$ Pulse Width , Excluding Power-Up	t_{RL}	1.5	–	t_{cyc}
Capture/Compare Timer				
Resolution (refer to NOTE 1)	t_{RESL}	4.0	–	t_{cyc}
Input Capture Pulse Width	t_{TH}, t_{TL}	250	–	ns
Input Capture Pulse Period	t_{TLTL}	(refer to NOTE 2)	–	t_{cyc}
Interrupt Pulse Width Low (Edge Triggered)	t_{ILIH}	250	–	ns
Interrupt Pulse Period	t_{ILIL}	(refer to NOTE 3)	–	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	–	ns

NOTES:

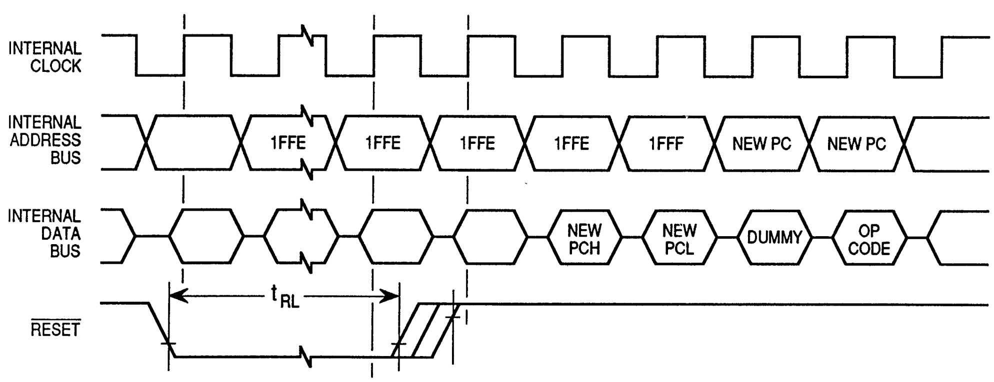
1. Because a 2-bit prescaler in the capture/compare timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.
2. The period t_{TLTL} should not be less than the number of cycles it takes to execute the capture interrupt service routine plus 24 t_{cyc}.
3. The period t_{ILIL} should not be less than the number of cycles it takes to execute the interrupt service routine plus 21 t_{cyc}.



NOTES:

1. Internal clock, internal address bus, and internal data bus signals are not available externally.
2. An internal POR reset is triggered as V_{DD} rises through a threshold (typically 1-2 V).

Figure 8-9. Power-On Reset Timing



NOTES:

1. Internal clock, internal address bus, and internal data bus signals are not available externally.
2. The next rising edge of the internal processor clock after the rising edge of $\overline{RESET}$ initiates the reset sequence.

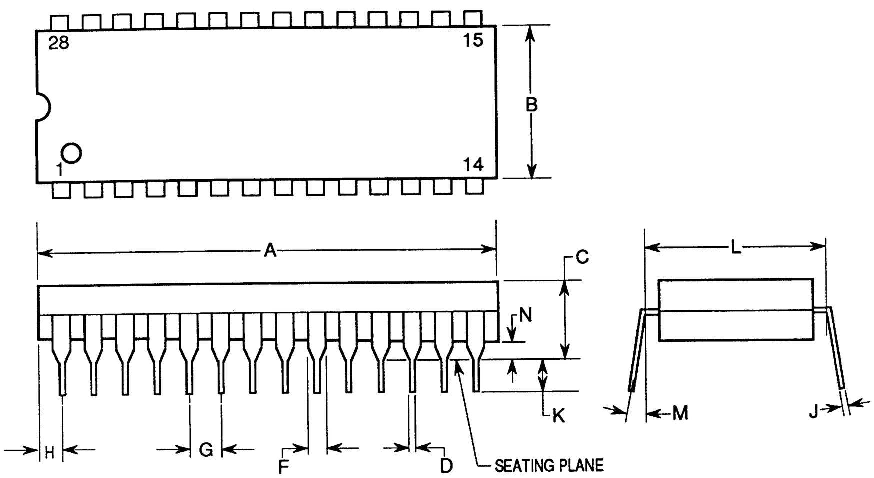
Figure 8-10. External Reset Timing

SECTION 9
MECHANICAL SPECIFICATIONS

This section describes the dimensions of the DIP (Dual-In-line Package) and SOIC (Small Outline Integrated Circuit) MCU packages.

9.1 Dual-In-Line Package (DIP)

P SUFFIX
PLASTIC PACKAGE
CASE 710-02



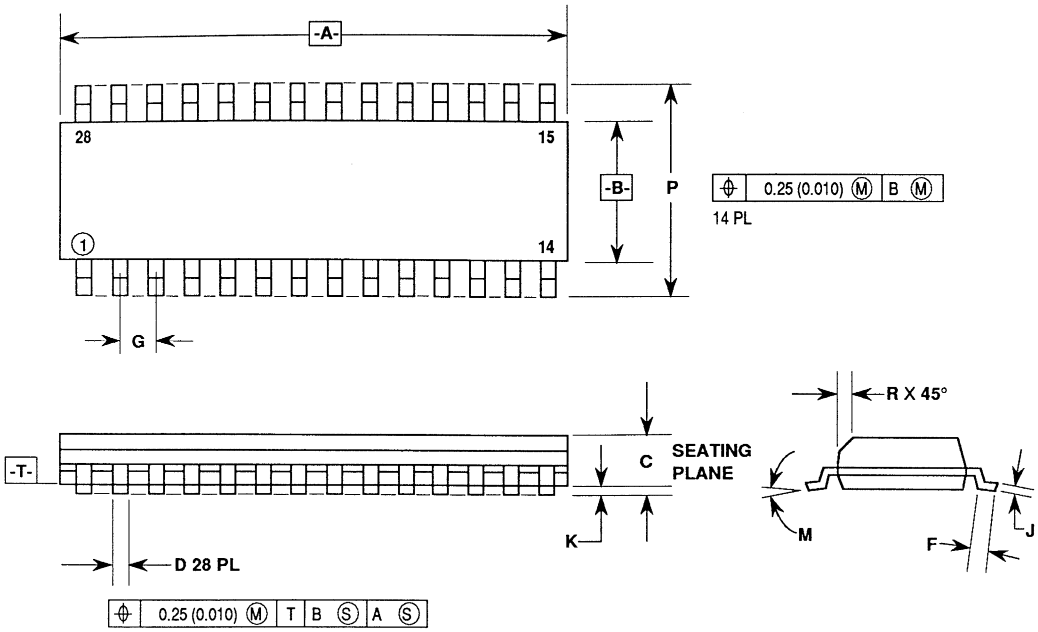
NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	36.45	37.21	1.435	1.465
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

Figure 9-1. Case 710-02 Dimensions

9.2 Small Outline Integrated Circuit (SOIC)

DW SUFFIX
SMALL OUTLINE INTEGRATED CIRCUIT
CASE 751F-02



NOTES:
1. DIMENSIONS A AND B ARE DATUMS AND T IS A DATUM SURFACE.
2. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
3. CONTROLLING DIMENSION: MILLIMETER.
4. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION.
5. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	17.80	18.05	0.701	0.710
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	0.50	0.90	0.020	0.035
G	1.27 BSC		0.050 BSC	
J	0.25	0.32	0.010	0.012
K	0.10	0.25	0.004	0.009
M	0°	7°	0°	7°
P	10.05	10.55	0.395	0.415
R	0.25	0.75	0.010	0.029

Figure 9-2. Case 751F-02 Dimensions

SECTION 10
ORDERING INFORMATION

Use the information contained in this section to order the MCU.

10.1 ROM Pattern Media

Ordering information can be delivered to Motorola in the following media:

- MS™-DOS[1] or PC-DOS flexible disk (360K)
- EPROM(s) 2764, MCM68764, MCM68766

To initiate a ROM pattern for the MCU, first contact the local field service office, a sales person, or a Motorola representative.

10.1.1 Flexible Disks

A flexible disk containing the customer's program (using positive logic for address and data), can be submitted for pattern generation. Clearly label the disk with the customer's name, data, project or product name, and the name of the file containing the pattern.

In addition to the program pattern, a file containing the program source code list can be included. This data is kept confidential and used to expedite the process in case of any difficulty with the pattern file.

MS-DOS is the Microsoft Disk Operating System. PC-DOS is the IBM®[2] Personal Computer (PC) Disk Operating System. Disks submitted must be standard density (360K) double-sided 5-1/4 in. The disks must contain object file code in Motorola's S-record format, a character-based object file format generated by M6805 cross assemblers and linkers on IBM PC-style machines.

---

[1]MS-DOS is a trademark of Microsoft, Inc.
[2]IBM is a registered trademark of International Business Machines Corporation.

10.1.2 EPROMs

A type 2764, 68764, or 68766 EPROM containing the customer's program (using positive logic for address and data), can be submitted for pattern generation. User ROM is programmed at EPROM addresses $0020 through $004F (page zero) and $0100 through $08FF with vectors at addresses $1FF0 to $1FFF. Set at logical zero all unused bytes, including those in the user's space. For shipment to Motorola, pack EPROMs securely in a conductive IC carrier. Do not package the EPROMs in a Styrofoam container.

10.2 ROM Pattern Verification

10.2.1 Verification Media

All original pattern media are filed for contractual purposes and are not returned. A computer list of the ROM code is generated and returned along with a listing verification form. Thoroughly check the list, and complete, sign and return the verification form to Motorola. The signed verification form constitutes the contractual agreement for the creation of the customer mask. To aid in the verification process, Motorola programs the *customer-supplied* blank EPROMs or DOS disks from the data file used to create the custom mask.

10.2.2 ROM Verification Units (RVUs)

Ten RVUs containing the customer's ROM pattern are sent for program verification. These units are made using the custom mask, but are for the purpose of ROM verification only. For expediency, the RVUs are unmarked, packaged in ceramic, and tested with 5 V at room temperature. These RVUs are free of charge with the minimum order quantity, but are not production parts. RVUs are not backed or guaranteed by Motorola Quality Assurance.

10.3 MC Order Numbers

Table 10-1 provides ordering information for available package types.

Table 10-1. MC Order Numbers

Package Type	Temperature	MC Order Number
Plastic DIP	0°C to + 70°C	MC68HC05P1P
SOIC	0°C to + 70°C	MC68HC05P1DW

MC68HC05P1 MCU ORDERING FORM

Date ________________ Customer PO Number ____________________________

Customer Company ____

Address ____

City ____________________ State ________________ Zip ________

Country ____

Phone _________________________________ Extension ____________

Customer Contact Person ______________________________________

Customer Part Number (if applicable - 12 characters maximum) ______________________

Application ____

Internal Oscillator Input:

- ☐ Crystal/Resonator
- ☐ Resistor

Interrupt Trigger:

- ☐ Edge-Sensitive
- ☐ Edge- and Level-Sensitive

Temperature Range:

- ☐ 0 to 70°C (Standard)
- ☐ –40 to +85°C

Special Electrical Provisions: ____________________________________
(Customer specifications required)

Pattern Media:

- ☐ MS-DOS Disk File
- ☐ 2764 EPROM
- ☐ MCM68764 EPROM
- ☐ PC-DOS Disk File
- ☐ MCM68766 EPROM
- ☐ Other ____
(Requires prior factory approval)

Device Marking:

- ☐ Motorola Standard
  - Motorola Logo
  - Motorola Part Number
  - Mask and Datecode
- ☐ Standard with Customer Part Number
  - Motorola Logo
  - Motorola Part Number
  - Customer Part Number
  - Mask and Datecode
- ☐ Other ____________________

Device marking other than the two standard forms requires prior factory approval.

____
(SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

____
(SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.

A23036-1 PRINTED IN USA 1/91 IMPERIAL LITHO C76732 18,000 MCU YGACAA

Literature Distribution Centers:

USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Center; 88 Tanners Drive, Blakelands, Milton Keynes, MK14 5BP, England.
JAPAN: Nippon Motorola Ltd.; 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141 Japan.
ASIA-PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Center, No. 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong.

MOTOROLA