MC68HC05J1/D
REV 1

HC05

MC68HC05J1

TECHNICAL
DATA

MOTOROLA

MC68HC05J1

HCMOS MICROCONTROLLER UNIT

Motorola reserves the right to make changes without further notice to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Employment Opportunity/Affirmative Action Employer.

© MOTOROLA INC., 1991

TABLE OF CONTENTS

Paragraph Number | Title | Page Number

Section 1
Introduction

Section 2
Signal Description, Input/Output Programming, Memory, and CPU Registers

Section 3
Self-Check, Resets, Interrupts, Low-Power and Data-Retention Modes

TABLE OF CONTENTS (Continued)

Section 4
Timer

Section 5
Instruction Set and Addressing Modes

TABLE OF CONTENTS (Concluded)

Section 6
Electrical Specifications

Section 7
Mechanical Data

Section 8
Ordering Information

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1
INTRODUCTION

The MC68HC05J1 high-density complementary metal-oxide semiconductor (HCMOS) microcontroller unit (MCU) is a member of the M68HC05 Family of microcontrollers. This high-performance, low-cost MCU has parallel I/O capability with pins programmable as input or output.

Figure 1-1 depicts the hardware features; additional features available on the MCU are as follows:

- On-Chip Oscillator with RC or Crystal/Ceramic Resonator Mask Options
- Memory-Mapped I/O
- 64 Bytes of On-Chip RAM
- 1040 Bytes of User ROM Including 16 Bytes of User Vectors
- 14 Bidirectional I/O Lines
- 15 Stage Multifunctional Timer
- Self-Check Mode
- Power-Saving STOP, WAIT, and Data-Retention Modes (STOP is a Mask Option)
- Single 3.0–5.5-Volt Supply (2-Volt Data Retention Mode)
- Fully Static Operation
- 8 × 8 Unsigned Multiply Instruction
- COP Watchdog System Enabled by Mask Option
- Illegal Address Reset

Four mask options are available: 1) clock (RC or crystal), 2) STOP instruction (enable/disable), 3) $\overline{\text{IRQ}}$ (edge sensitive only or edge sensitive and level sensitive), and 4) COP watchdog timer (enable/disable).

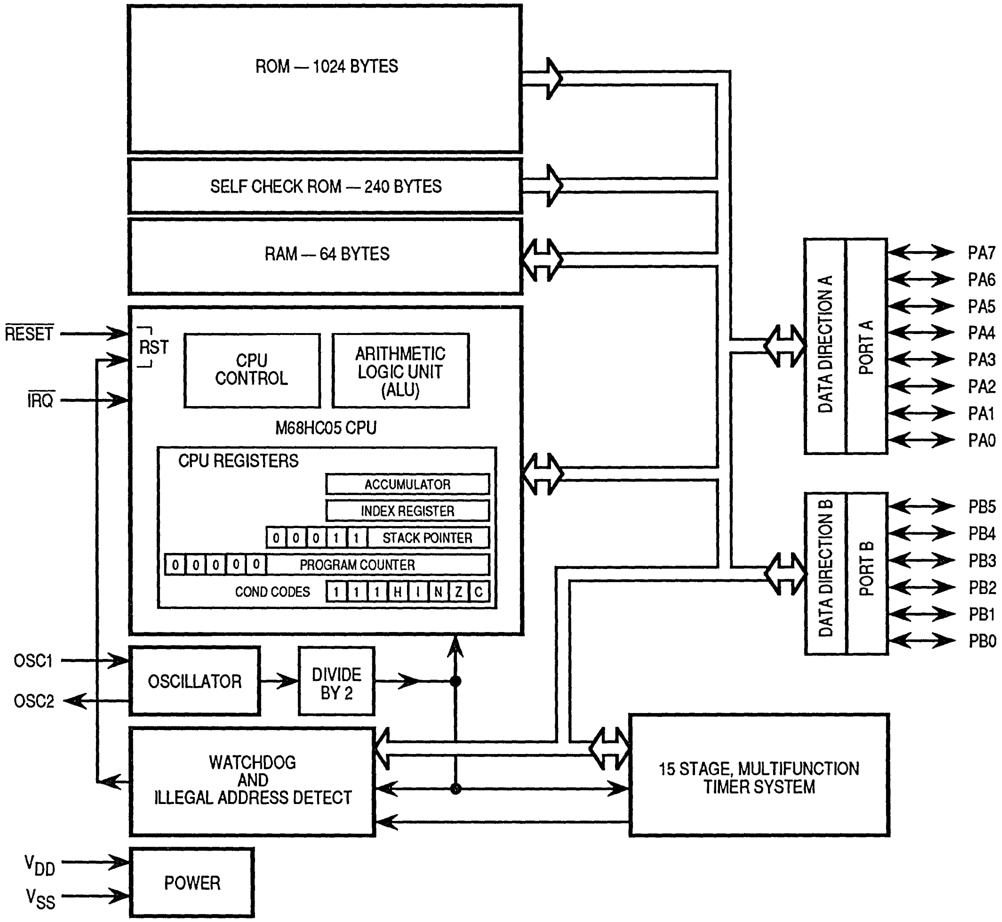


Figure 1-1. MC68HC05J1 Block Diagram

SECTION 2
SIGNAL DESCRIPTION, INPUT/OUTPUT PROGRAMMING, MEMORY, AND CPU REGISTERS

This section provides a description of the signals, input/output programming, memory, and CPU registers.

2.1 SIGNAL DESCRIPTION

The following paragraphs provide a description of the MC68HC05J1 signals. Reference is made, where applicable, to other sections that contain more detail about the function being performed.

2.1.1 V_{DD} and V_{SS}

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

2.1.2 $\overline{\text{IRQ}}$

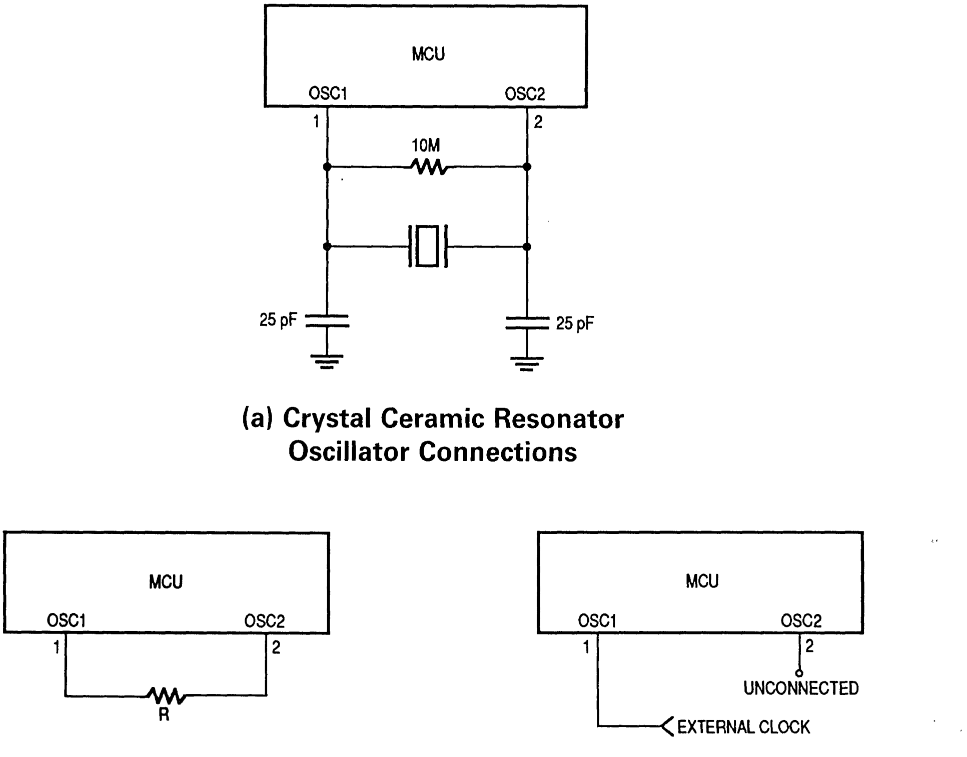
This pin provides the capability for applying an asynchronous external interrupt to the MCU. It has a mask option that provides two different triggering sensitivity choices. Refer to **3.3.1 External Interrupts** for more detail.

2.1.3 OSC1, OSC2

These two pins provide connections for an on-chip clock oscillator circuit. A crystal, a ceramic resonator, a resistor, or an external signal connects to these pins to provide a system clock. Each alternative is discussed in subsequent paragraphs. In all cases, particular care should be taken in the circuit board layout around the oscillator pins. The oscillator frequency (f_{OSC}) connected to these pins is two times the internal bus operating frequency (f_{op}).

2.1.3.1 CRYSTAL. The circuit in Figure 2-1(a) shows a typical crystal oscillator circuit for an AT-cut, parallel resonant crystal. The crystal supplier's recommendations should be followed, since the crystal parameters determine the external component values required to provide maximum stability and reliable starting. The load capacitance values shown in the oscillator circuit include all stray layout capacitances.

2.1.3.2 CERAMIC RESONATOR. A ceramic resonator can be used in place of the crystal in cost-sensitive applications. The circuit in Figure 2-1(a) can be used for a ceramic resonator, but the manufacturer of the resonator should be consulted for specific information on external component values, since the ceramic resonator characteristics determine the external component values required.



(a) Crystal Ceramic Resonator Oscillator Connections

(b) RC Oscillator Connections

(c) External Clock Source Connections (For Crystal Mask Option Only)

Figure 2-1. Oscillator Connections

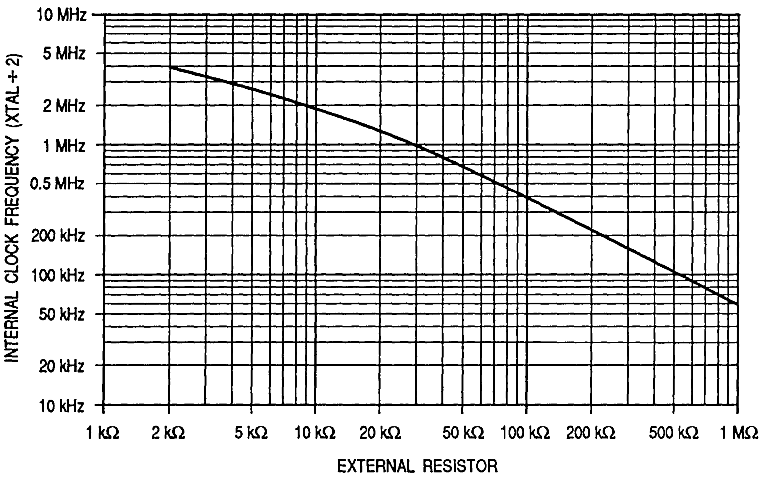


Figure 2-2. Typical Frequency vs Resistance for RC Oscillator Option

2.1.3.3 RC OSCILLATOR. With this option, a resistor is connected to the oscillator pins as shown in Figure 2-1(b). The relationship between R and the internal clock (f_{op}) is shown in Figure 2-2. Consult factory for tolerance limits and design specifications.

2.1.3.4 EXTERNAL CLOCK. An external clock from another CMOS-compatible device can be connected to the OSC1 input, with the OSC2 input not connected, as shown in Figure 2-1(c).

2.1.4 $\overline{\text{RESET}}$

A logic zero level on this pin forces the MCU to a known startup state. Refer to **3.2 RESETS** for additional information.

2.1.5 Input/Output Ports (PA7–PA0, PB5–PB0)

These 14 lines are arranged as one 8-bit port (A) and one 6-bit port (B). All 14 lines are independently programmable as either inputs or outputs under software control of the data direction registers. Refer to **2.2 I/O PORT PROGRAMMING** for additional information.

NOTE

> Any unused inputs and I/O ports should be tied to an appropriate logic level (e.g., either V_{DD} or V_{SS}). Although the I/O ports of the MC68HC05J1 do not specifically require termination, it is recommended, to reduce the possibility of static damage.

2.2 I/O PORT PROGRAMMING

Any port pin is programmable as either an input or an output under software control of the corresponding data direction register (DDR). Each port bit can be selected as output or input by writing the corresponding bit in the port DDR to a logic one for output or logic zero for input. On reset, all DDRs are initialized to logic zero to put the ports in the input mode. The port data registers are not initialized on reset, but may be written to before setting the DDR bits to avoid undefined levels.

When programmed as outputs, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading. The latched output data bit may always be written. Therefore, any write to a port writes all of its data bits, even though the port DDR is set to input. This port write may be used to initialize the data registers and avoid undefined outputs. Refer to Figure 2-3 for typical port circuitry.

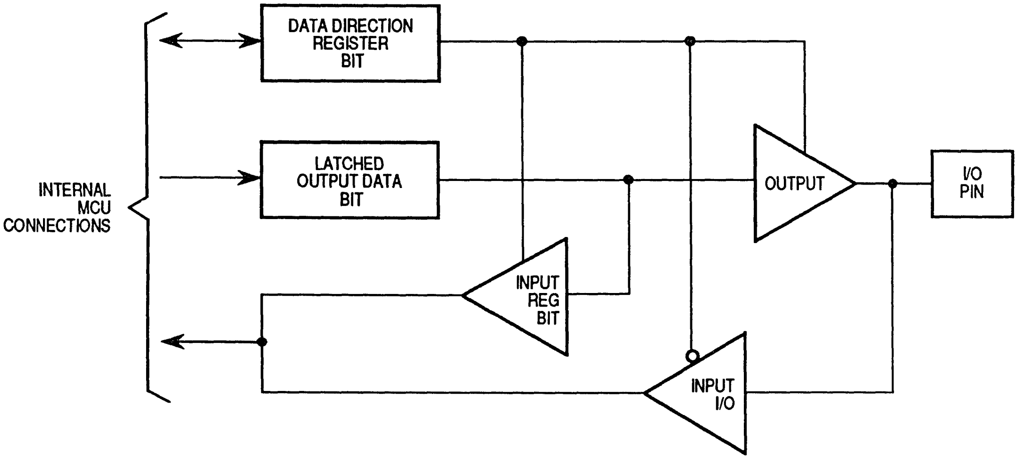


Figure 2-3. Typical Port I/O Circuit

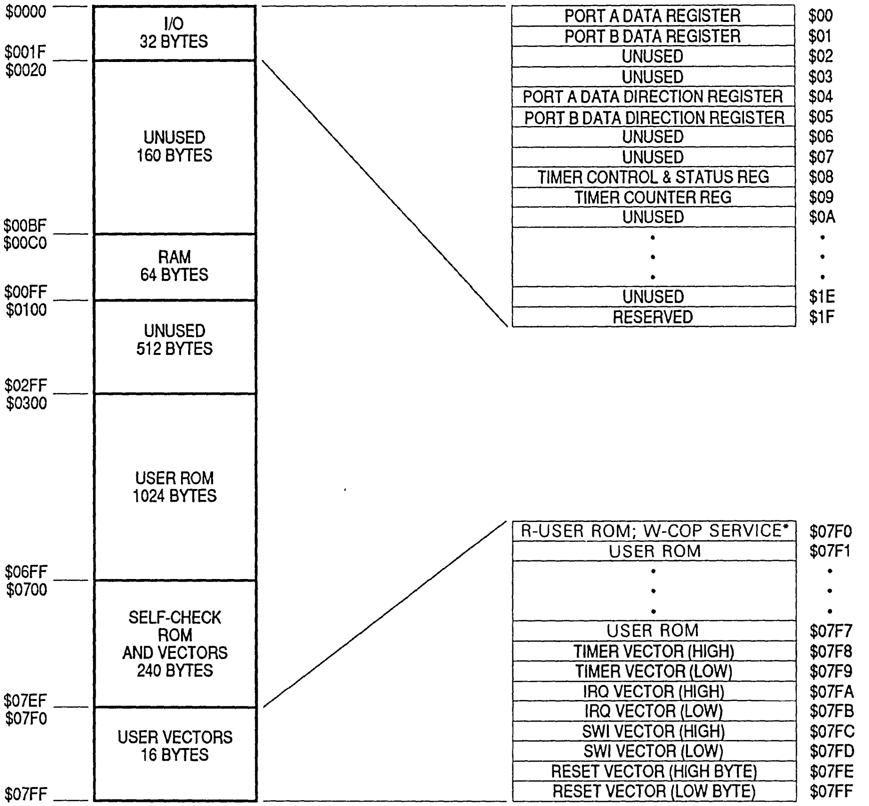
2.3 MEMORY

The MCU is capable of addressing 2048 bytes of memory and I/O registers, as shown in Figure 2-4. The locations consist of user ROM, user RAM, self-check ROM, control registers, and I/O. The user-defined reset and interrupt vectors are located from $07F0–$07FF.

The shared stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls. Refer to **3.3 INTERRUPTS** for additional information.

NOTE

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



*Writing 0 to bit 0 of $07F0 clears COP timer. Reading $07F0 returns user ROM data.

Figure 2-4. MC68HC05J1 Memory Map

2.4 CPU REGISTERS

The MCU contains the registers described in the following paragraphs.

2.4.1 Accumulator (A)

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

7	0
A	

2.4.2 Index Register (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to a 0-, 8-, or 16-bit immediate value to create an effective address. The index register may also be used as a temporary storage area.

7	0
X	

2.4.3 Program Counter (PC)

The program counter is an 11-bit register that contains the address of the next byte to be fetched. Since addresses are often expressed as 16-bit values, the PC may be thought of as having five imaginary upper bits, which are always zeros.

15	14	13	12	11	10	0
0	0	0	0	0	PC	

2.4.4 Stack Pointer (SP)

The stack pointer is an 11-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset-stack-pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the five most significant bits are permanently set to 00011. These five 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 (decimal) locations. If the total of 64 locations is exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.

10			7		0
0	0	0	1	1	SP

2.4.5 Condition Code Register (CCR)

The CCR is a 5-bit register in which the H, N, Z, and C bits are used to indicate the results of the instruction just executed, and the I bit is used to enable interrupts. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. The CCR should be thought of as having three additional upper bits that are always ones. The function of each of the lower five bits is explained in the following paragraphs.

7	6	5	4				0
1	1	1	H	I	N	Z	C

2.4.5.1 HALF CARRY (H). This bit is set during add (ADD) and add with carry (ADC) operations to indicate that a carry occurred between bits 3 and 4. Operations on binary coded decimal (BCD) values require this status indicator.

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

2.4.5.3 NEGATIVE (N). When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic one).

2.4.5.4 ZERO (Z). When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

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

SECTION 3
SELF-CHECK, RESETS, INTERRUPTS, LOW-POWER AND DATA-RETENTION MODES

Self-check, resets, interrupts, low-power and data-retention modes are discussed in the following paragraphs.

3.1 SELF-CHECK

The self-check capability provides the ability to determine if the device is functional. Self-check is performed using the circuit shown in Figure 3-1. Self-check is entered on the rising edge of $\overline{RESET}$ if $\overline{IRQ}$ is at $2 \times V_{DD}$ and PB1 is at logic one. $\overline{RESET}$ must held low for at least 4064 f_{op} cycles after power-on reset or one and one-half t_{cyc} after any other reset. Port A pins PA3–PA0

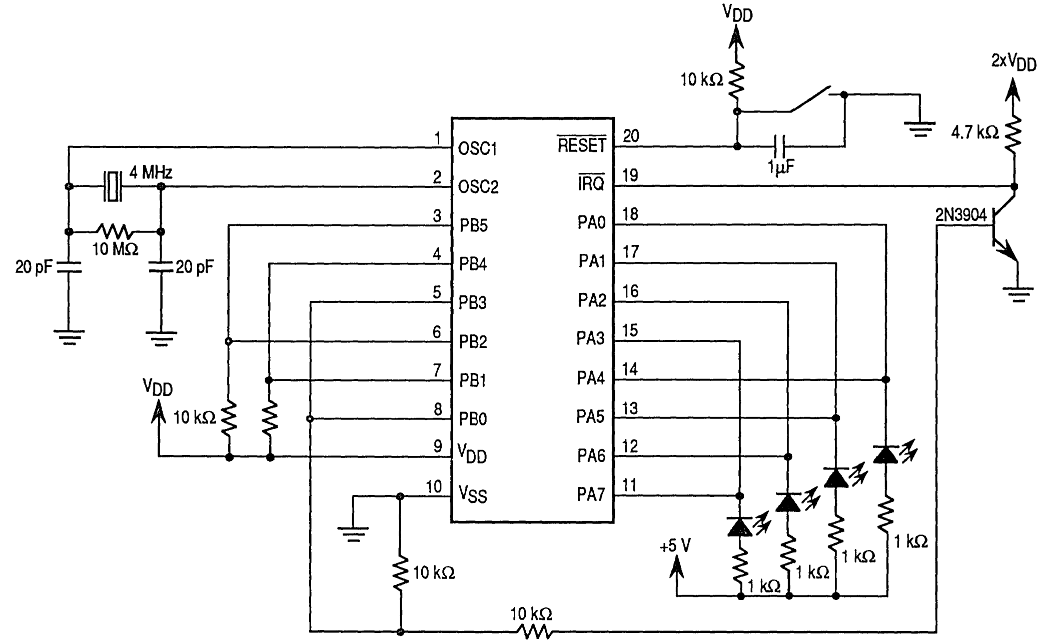


Figure 3-1. Self-Check Circuit Schematic Diagram

are monitored for the self-check results. After reset, the following five tests are performed automatically:

I/O — Exercise of ports A and B
RAM — Counter test for each RAM byte
Timer — Tracks counter register and checks timer status flag bits
ROM — Exclusive OR with odd ones parity result
Interrupt — Tests external and timer interrupts

Self-check results (using the LEDs as monitors) are shown in Table 3-1.

Table 3-1. Self-Check Results

PA3	PA2	PA1	PA0	Remarks
1	0	0	1	Bad I/O
1	0	1	0	Bad RAM
1	0	1	1	Bad Timer
1	1	0	0	Bad ROM
1	1	0	1	Bad Interrupts or $\overline{\text{IRQ}}$ Request
Flashing				Good Device
All Others				Bad Device

0 indicates LED is on; 1 indicates LED is off.

3.2 RESETS

Reset halts all MCU operation immediately. Reset forces the MCU to assume a set of initial conditions and begin executing instructions from a predetermined starting address. The MCU can be reset four ways: by initial power-up, by the external $\overline{\text{RESET}}$ input, by a COP timeout, and by the illegal address reset.

Reset forces the following conditions:

• PORTA, PORTB Data Direction	Cleared
• Timer Divide-by-Four Prescaler	Reset
• Timer Counter	Cleared
• RTI Rate	Set for Lowest Rate
• Timer Interrupt Flags	Cleared
• Timer Interrupt Enable Bits	Cleared
• Stack Pointer	$0FF
• Internal Address Bus	$7FE
• I Bit in Condition Code Register	Set
• $\overline{\text{IRQ}}$ Request Latch	Cleared

3.2.1 Power-On Reset (POR)

An internal reset is generated on power-up to allow the internal clock generator to stabilize. The POR is designed for use at power turn-on only and is not intended to be used as a means to detect a drop in the supply voltage. There is a 4064 t_{CYC} (internal processor clock cycles) delay after the oscillator becomes active to allow the clock generator to stabilize. If the external $\overline{\text{RESET}}$ pin is still being held low at the end of 4064 t_{CYC}, the MCU will remain in reset until $\overline{\text{RESET}}$ goes high.

3.2.2 External Reset Input

The MCU reset sequence is initiated whenever a logic zero is applied to the $\overline{\text{RESET}}$ input for a period of one and one-half t_{CYC}. The $\overline{\text{RESET}}$ input consists primarily of a Schmitt trigger that senses the logic level on the $\overline{\text{RESET}}$ line. $\overline{\text{RESET}}$ is an input-only pin and will not indicate when an internal reset has occurred.

3.2.3 COP Timeout Reset

This MCU offers a computer operating properly (COP) watchdog system as a mask option to protect against software failures. When the COP is enabled, a COP reset sequence must be issued on a regular basis so that the COP does not time out and initiate an MCU reset. When the COP circuit times out, an internal reset is generated and the normal reset procedure is initiated. A timeout is prevented by periodically clearing the COP bit (writing a zero to bit 0 of location $07F0) at a time interval that is smaller than the COP timeout interval. The COP watchdog system is implemented on this MCU by using the output of the real-time interrupt circuit and further dividing it by eight. Refer to Table 4-1.

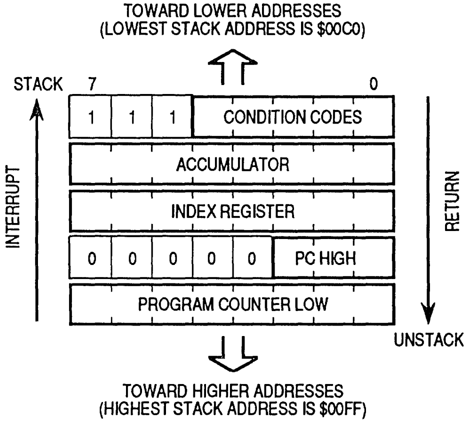
3.2.4 Illegal Address Reset

If an opcode fetch occurs from an address that is not implemented in ROM or RAM, the MCU is automatically reset.

3.3 INTERRUPTS

The MCU can be interrupted three different ways: the two maskable hardware interrupts ($\overline{\text{IRQ}}$ and timer) and the nonmaskable software interrupt instruction (SWI).

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume. The stacking order is shown in Figure 3-2.



NOTE: When an interrupt occurs, CPU registers are saved on the stack in the order PCL, PCH, X, A, CC. On a return from interrupt registers are recovered from the stack in reverse order.

Figure 3-2. Interrupt Stack Order

Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted but are considered pending until the current instruction is complete. (The current instruction is the one already fetched and being operated on.)

When the current instruction is complete, the processor checks all pending hardware interrupts. If unmasked (I bit clear) and if the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction, regardless of the I bit state. Refer to Figure 3-3 for the reset and interrupt instruction processing sequence.

3.3.1 External Interrupt

If the I bit is set, all interrupts are disabled. Clearing the I bit enables the external interrupt. The external interrupt is latched and then internally synchronized on the falling edge of $\overline{\text{IRQ}}$. The action of the external interrupt is identical to the timer interrupt, except that the interrupt request input at $\overline{\text{IRQ}}$ is latched internally and the service routine address is specified by the contents of $07FA and $07FB.

Either a level-sensitive and edge-sensitive trigger or an edge-sensitive-only trigger are available as a mask option. Figure 3-4 shows a functional internal diagram for the interrupt line. Refer to Figure 6-7 for a mode timing diagram for the interrupt line. There are two treatments of the interrupt line to the processor. The first method shows two single pulses on the interrupt line spaced far enough apart to be serviced. The minimum time between pulses is a function of the length of the interrupt service. Once a pulse occurs, the next pulse should normally not occur until a return-from-interrupt (RTI) instruction occurs. This time (t_{ILIL}) is obtained by adding 19 instruction cycles to the total number of cycles it takes to complete the service routine (not including the RTI instruction). The second method shows many interrupt lines "wire-ORed" to form the interrupts at the processor. If the interrupt line remains low after servicing an interrupt, then the next interrupt is recognized.

NOTE

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

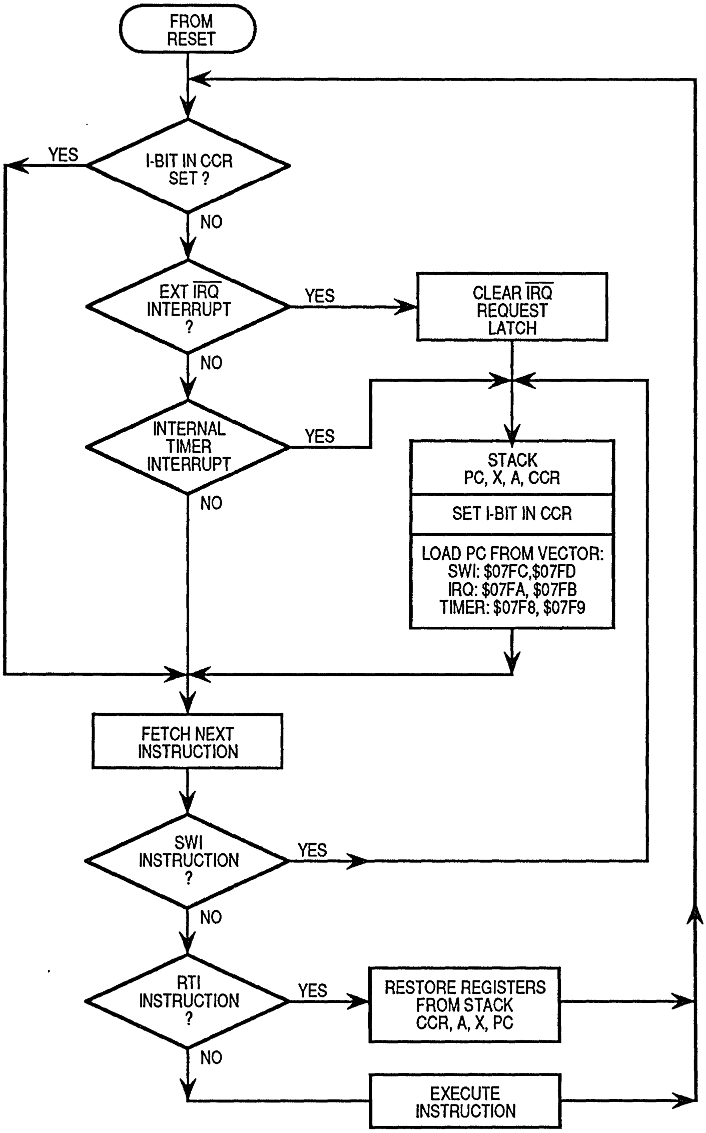


Figure 3-3. Hardware Interrupt Flowchart

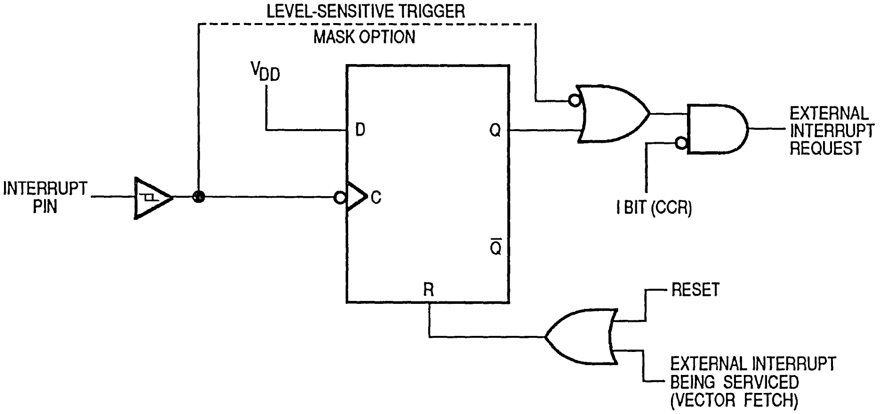


Figure 3-4. External Interrupt Internal Function Diagram

3.3.2 Timer Interrupt

There are two different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The interrupt flags and enable bits are in the timer control and status register (TCSR). Refer to **SECTION 4 TIMER** for more information.

3.3.3 Software Interrupt (SWI)

The SWI is an executable instruction that is executed regardless of the state of the I bit in the CCR. If the I bit is zero, SWI executes after the other interrupts. The SWI operation is similar to the hardware interrupts. The interrupt service routine address is specified by the contents of memory locations $07FC and $07FD.

3.4 LOW-POWER MODES

The STOP, WAIT, and data-retention modes are discussed in the following paragraphs.

3.4.1 STOP Mode

The STOP instruction places the MCU in its lowest power-consumption mode. In response to a STOP instruction, the internal oscillator is turned off, halting all internal processing including timer operation and COP watchdog system operation (refer to Figure 3-5).

During the STOP mode, the interrupt flags (TOF and RTIF) and interrupt enable bits (TOFE and RTIE) in the TCSR are cleared to remove any pending timer interrupt requests and to disable further timer interrupts. The timer prescaler is cleared. The I bit is cleared to enable external interrupts. All other registers, including the other bits in the TCSR, and memory are unaltered. All I/O lines remain unchanged. The processor can be brought out of STOP only by an external interrupt or reset.

The STOP instruction can be disabled by a mask option. When disabled, the STOP instruction is executed as a NOP.

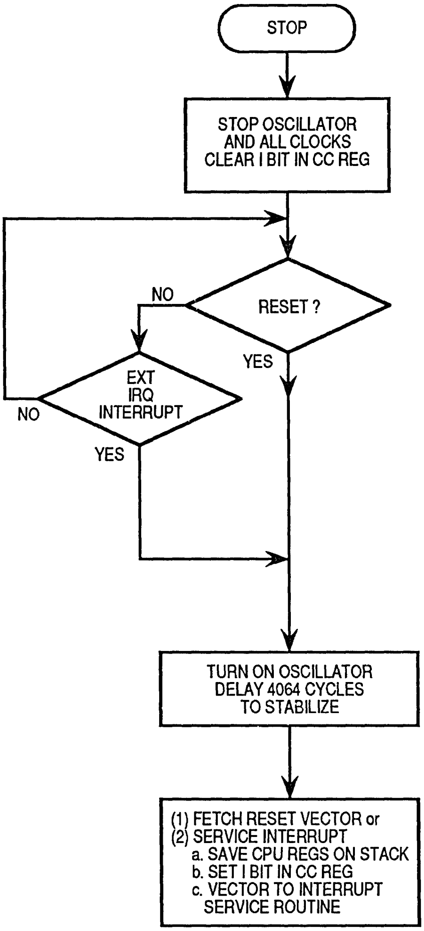


Figure 3-5. STOP Function Flowchart

3.4.2 WAIT Mode

The WAIT instruction places the MCU in a low power-consumption mode, but the WAIT mode consumes more power than the STOP mode. All CPU action is suspended, but the timer remains active (refer to Figure 3-6). An interrupt from the timer will cause the MCU to exit the WAIT mode.

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

If the COP option is selected, the COP resets the MCU when it times out, and so COP should not be selected for a system that uses WAIT mode for periods longer than the COP period.

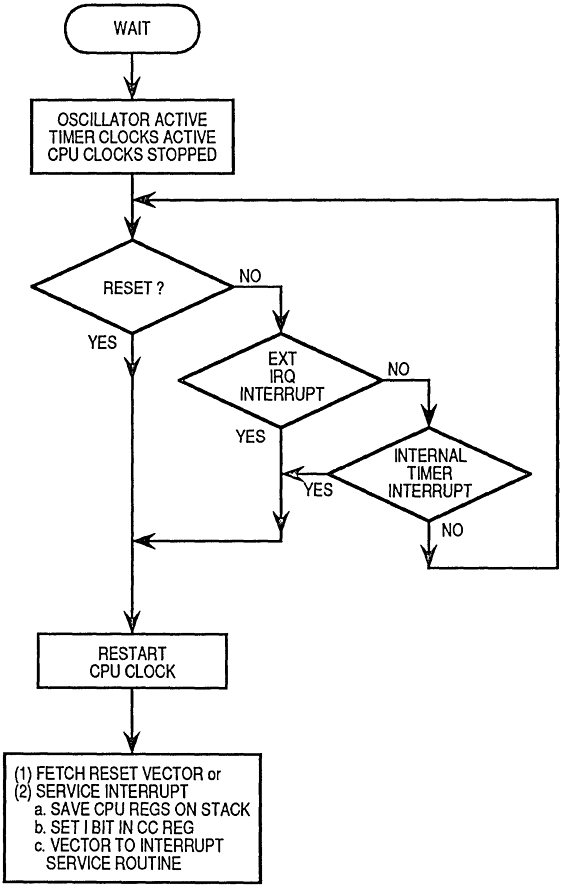


Figure 3-6. WAIT Function Flowchart

3.4.3 Data-Retention Mode

The contents of RAM and CPU registers are retained at supply voltages as low as 2.0 Vdc. This is called the data-retention mode where the data is held, but the device is not guaranteed to operate. $\overline{\text{RESET}}$ must be held low during data-retention mode.

SECTION 4
TIMER

The timer in this MCU is a 15-stage, multifunction ripple counter. As shown in Figure 4-1, the timer begins with a fixed divide-by-four prescaler which drives an 8-bit ripple counter. The value of this counter can be read at any time at the timer counter register (TCR). A timer overflow function is implemented on the last stage of this counter, allowing a potential interrupt at the rate of f_{op} (internal processor clock) divided by 1024. Following this counter are four additional divider stages leading to the real-time interrupt (RTI) circuit. The RTI circuit consists of three additional divider stages and a one-

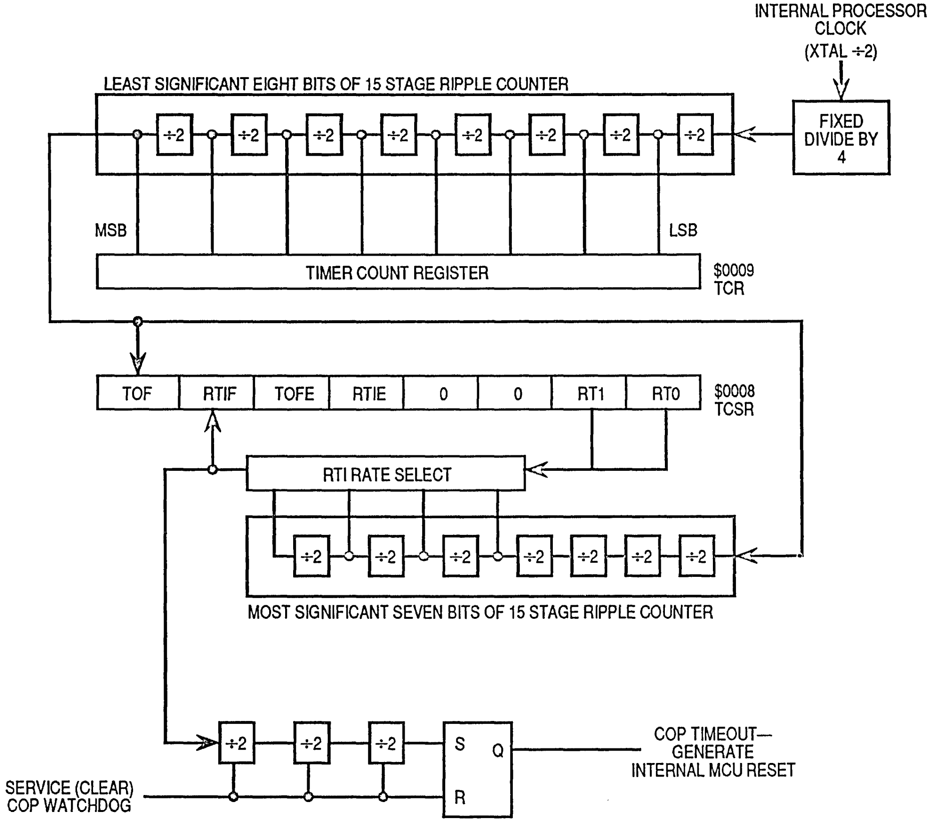

INTERNAL PROCESSOR CLOCK (XTAL ÷2)
FIXED DIVIDE BY 4
LEAST SIGNIFICANT EIGHT BITS OF 15 STAGE RIPPLE COUNTER
÷2
MSB
LSB
TIMER COUNT REGISTER
$0009 TCR
TOF
RTIF
TOFE
RTIE
0
0
RT1
RT0
$0008 TCSR
RTI RATE SELECT
MOST SIGNIFICANT SEVEN BITS OF 15 STAGE RIPPLE COUNTER
S
Q
R
COP TIMEOUT— GENERATE INTERNAL MCU RESET
SERVICE (CLEAR) COP WATCHDOG

Figure 4-1. Timer Block Diagram

of-four rate selector. The RTI output is further divided by eight to drive the COP system. The RTI rate selector bits and the RTI and TOF enable bits and flags are located in the TCSR.

4.1 TIMER CONTROL AND STATUS REGISTER

	7	6	5	4	3	2	1	0	
$0008	TOF	RTIF	TOFE	RTIE	0	0	RT1	RT0	**TCSR**
RESET:	0	0	0	0	0	0	1	1	

TOF — Timer Overflow Flag

This is a clearable, read-only status bit that is set when the 8-bit ripple counter rolls over from $FF to $00. Rollover also generates a CPU interrupt request if the timer overflow enable bit (TOFE) is set. This bit is cleared by reset or by writing a zero to it. Writing a one to this bit has no effect.

RTIF — Real-Time Interrupt Flag

This is a clearable, read-only status bit that is set when the output of the chosen divider stage (any one of the four) becomes active. RTIF also generates a CPU interrupt request if the real-time interrupt enable bit (RTIE) is set. This bit is cleared by reset or by writing a zero to it. Writing a one to this bit has no effect. At a bus rate of 2 MHz, the maximum interrupt period is 65.5 ms. Refer to Table 4-1.

TOFE — Timer Overflow Enable

When this bit is set, a CPU interrupt request is generated whenever TOF is set.

RTIE—Real-Time Interrupt Enable

When this bit is set, a CPU interrupt request is generated whenever RTIF is set.

Bits 3–2 — Not used; always read zero.

RT1–RT0 — Real-Time Interrupt Rate Selects

These two bits select one of the four taps in the real-time interrupt circuit and allow changing the real-time interrupt rate. Reset sets both bits, selecting the lowest periodic rate, which allows the maximum time to alter this setting if desired. In normal systems the RTI rate would be set one time during reset initialization software. Unpredictable results may be obtained by altering this rate when the timeout period is imminent or uncertain. If the selected tap is modified during a cycle in which the counter is switching, an RTIF could be missed or an extra one generated. Because the COP timer is derived from this rate, changing this rate will also affect the COP system. The COP should be cleared before changing RTI taps. Refer to Table 4-1.

Table 4-1. RTI and COP Rates (f_{op}=2 MHz)

RT1	RT0	RTI Rate	Minimum COP Reset
0	0	8.2 ms	57.3 ms
0	1	16.4 ms	114.7 ms
1	0	32.8 ms	229.4 ms
1	1	65.5 ms	458.8 ms

4.2 TIMER COUNTER REGISTER

	7	6	5	4	3	2	1	0	
$0009	TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0	**TCR**

This 8-bit read-only register contains the value of the 8-bit ripple counter at the beginning of the timer chain. The counter is clocked at a rate of f_{op} divided by four and can be employed for various functions, including a software input capture. Extended time periods can be measured by using the TOF feature to increment a temporary RAM storage location, thereby simulating a 16-bit (or larger) timer. Power-on clears the entire counter chain and begins clocking the counter. After 4064 cycles, the power-on reset circuit is released, which again clears the counter chain and allows the device to come out of reset. At this point, if $\overline{\text{RESET}}$ is not asserted (driven low), the timer will start counting up from zero, and normal operation will begin. When $\overline{\text{RESET}}$ is asserted anytime during operation, the counter chain is cleared.

4.3 TIMER DURING STOP AND WAIT

The CPU clock halts during WAIT, but the timer remains active. If interrupts are not masked, the occurrence of a timer interrupt will cause the processor to exit WAIT.

The timer is cleared when STOP is executed. When STOP is exited by an external interrupt or a $\overline{\text{RESET}}$, the internal oscillator will resume operation. Following the 4064-cycle internal processor delay, the timer is cleared, and normal operation continues.

SECTION 5
INSTRUCTION SET AND ADDRESSING MODES

This section provides a description of the instruction set and addressing modes.

5.1 INSTRUCTION SET

The MCU has a set of basic instructions that can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

Table 5-1 shows all the MC68HC05J1 instructions in all possible addressing modes. For each instruction, the operand construction is shown as well as the number of machine code bytes and the execution time in internal processor clock cycles (t_{cyc}). One internal processor clock cycle equals two oscillator input cycles (f_{osc}).

Table 5-1. Instructions, Addressing Modes, and Execution Times (Sheet 1 of 4)

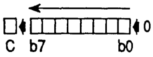
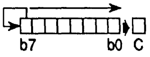
Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (hexadecimal)		Bytes	Cycles	Condition Code				
				Opcode	Operand			H	I	N	Z	C
ADC (opr)	Add with Carry	ACCA ← ACCA + M + C	IMM	A9	ii	2	2	↕	—	↕	↕	↕
			DIR	B9	dd	2	3					
			EXT	C9	hh ll	3	4					
			IX2	D9	ee ff	3	5					
			IX1	E9	ff	2	4					
			IX	F9		1	3					
ADD (opr)	Add	ACCA ← ACCA + M	IMM	AB	ii	2	2	↕	—	↕	↕	↕
			DIR	BB	dd	2	3					
			EXT	CB	hh ll	3	4					
			IX2	DB	ee ff	3	5					
			IX1	EB	ff	2	4					
			IX	FB		1	3					
AND (opr)	Logical AND	ACCA ← ACCA ● M	IMM	A4	ii	2	2	—	—	↕	↕	—
			DIR	B4	dd	2	3					
			EXT	C4	hh ll	3	4					
			IX2	D4	ee ff	3	5					
			IX1	E4	ff	2	4					
			IX	F4		1	3					
ASL (opr)	Arithmetic Shift Left		DIR	38	dd	2	5	—	—	↕	↕	↕
ASLA			INH(A)	48		1	3					
ASLX			INH(X)	58		1	3					
ASL (opr)			IX1	68	ff	2	6					
ASL (opr)			IX	78		1	5					
ASR (opr)	Arithmetic Shift Right		DIR	37	dd	2	5	—	—	↕	↕	↕
ASRA			INH(A)	47		1	3					
ASRX			INH(X)	57		1	3					
ASR (opr)			IX1	67	ff	2	6					
ASR (opr)			IX	77		1	5					
BCC (rel)	Branch if Carry Clear	? C = 0	REL	24	rr	2	3	—	—	—	—	—
BCLR n, (opr)	Clear Bit n in Memory	Mn ← 0	DIR(b0)	11	dd	2	5	—	—	—	—	—
			DIR(b1)	13	dd	2	5					
			DIR(b2)	15	dd	2	5					
			DIR(b3)	17	dd	2	5					
			DIR(b4)	19	dd	2	5					
			DIR(b5)	1B	dd	2	5					
			DIR(b6)	1D	dd	2	5					
			DIR(b7)	1F	dd	2	5					
BCS (rel)	Branch if Carry Set	? C = 1	REL	25	rr	2	3	—	—	—	—	—
BEQ (rel)	Branch if Equal	? Z = 1	REL	27	rr	2	3	—	—	—	—	—
BHCC (rel)	Branch if Half Carry Clear	? H = 0	REL	28	rr	2	3	—	—	—	—	—
BHCS (rel)	Branch if Half Carry Set	? H = 1	REL	29	rr	2	3	—	—	—	—	—
BHI (rel)	Branch if Higher	? (C + Z) = 0	REL	22	rr	2	3	—	—	—	—	—
BHS (rel)	Branch if Higher or Same	? C = 0	REL	24	rr	2	3	—	—	—	—	—
BIH (rel)	Branch if $\overline{IRQ}$ Pin is High	? $\overline{IRQ}$ Pin = 1	REL	2F	rr	2	3	—	—	—	—	—
BIL (rel)	Branch if $\overline{IRQ}$ Pin is Low	? $\overline{IRQ}$ Pin = 0	REL	2E	rr	2	3	—	—	—	—	—
BIT (rel)	Bit Test Memory with A	ACCA ● M	IMM	A5	ii	2	2	—	—	↕	↕	—
			DIR	B5	dd	2	3					
			EXT	C5	hh ll	3	4					
			IX2	D5	ee ff	3	5					
			IX1	E5	ff	2	4					
			IX	F5		1	3					
BLO (rel)	Branch if Lower	? C = 1	REL	25	rr	2	3	—	—	—	—	—
BLS (rel)	Branch if Lower or Same	?(C+Z) = 1	REL	23	rr	2	3	—	—	—	—	—
BMC (rel)	Branch if I Bit is Clear	? I = 0	REL	2C	rr	2	3	—	—	—	—	—

Table 5-1. Instructions, Addressing Modes, and Execution Times (Sheet 2 of 4)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (hexadecimal)		Bytes	Cycles	Condition Code				
				Opcode	Operand			H	I	N	Z	C
BMI (rel)	Branch if Minus	? N = 1	REL	2B	rr	2	3	—	—	—	—	—
BMS (rel)	Branch if I Bit is Set	? I = 1	REL	2D	rr	2	3	—	—	—	—	—
BNE (rel)	Branch if Not Equal	? Z = 0	REL	26	rr	2	3	—	—	—	—	—
BPL (rel)	Branch if Plus	? N = 0	REL	2A	rr	2	3	—	—	—	—	—
BRA (rel)	Branch Always	? 1 = 1	REL	20	rr	2	3	—	—	—	—	—
BRCLR n, (opr) (rel)	Branch if Bit n of M = 0	? Bit n of M = 0	DIR(b0)	01	dd rr	3	5	—	—	—	—	↕
			DIR(b1)	03	dd rr	3	5					
			DIR(b2)	05	dd rr	3	5					
			DIR(b3)	07	dd rr	3	5					
			DIR(b4)	09	dd rr	3	5					
			DIR(b5)	0B	dd rr	3	5					
			DIR(b6)	0D	dd rr	3	5					
			DIR(b7)	0F	dd rr	3	5					
BRN (rel)	Branch Never	? 1 = 0	REL	21	rr	2	3	—	—	—	—	—
BRSET n, (opr) (rel)	Branch if Bit n of M = 1	? Bit n of M = 1	DIR(b0)	00	dd rr	3	5	—	—	—	—	↕
			DIR(b1)	02	dd rr	3	5					
			DIR(b2)	04	dd rr	3	5					
			DIR(b3)	06	dd rr	3	5					
			DIR(b4)	08	dd rr	3	5					
			DIR(b5)	0A	dd rr	3	5					
			DIR(b6)	0C	dd rr	3	5					
			DIR(b7)	0E	dd rr	3	5					
BSET n, (opr)	Set Bit n in Memory	Mn ← 1	DIR(b0)	10	dd	2	5	—	—	—	—	—
			DIR(b1)	12	dd	2	5					
			DIR(b2)	14	dd	2	5					
			DIR(b3)	16	dd	2	5					
			DIR(b4)	18	dd	2	5					
			DIR(b5)	1A	dd	2	5					
			DIR(b6)	1C	dd	2	5					
			DIR(b7)	1E	dd	2	5					
BSR (rel)	Branch to Subroutine	PC ← PC + 0002 (SP) ← PCL; SP ← SP − 0001 (SP) ← PCH; SP ← SP − 0001 PC ← PC + Rel	REL	AD	rr	2	6	—	—	—	—	—
CLC	Clear C Bit	C bit ← 0	INH	98		1	2	—	—	—	—	0
CLI	Clear I Bit	I bit ← 0	INH	9A		1	2	—	0	—	—	—
CLR (opr)	Clear	M ← 00	DIR	3F	dd	2	5	—	—	0	1	—
CLRA		A ← 00	INH(A)	4F		1	3					
CLRX		X ← 00	INH(X)	5F		1	3					
CLR (opr)		M ← 00	IX1	6F	ff	2	6					
CLR (opr)		M ← 00	IX	7F		1	5					
CMP (opr)	Compare A with Memory	ACCA − M	IMM	A1	ii	2	2	—	—	↕	↕	↕
			DIR	B1	dd	2	3					
			EXT	C1	hh ll	3	4					
			IX2	D1	ee ff	3	5					
			IX1	E1	ff	2	4					
			IX	F1		1	3					
COM (opr)	1's Complement	M ← $\overline{M} = \$FF - M$	DIR	33	dd	2	5	—	—	↕	↕	1
COMA		A ← $\overline{A} = \$FF - A$	INH(A)	43		1	3					
COMX		X ← $\overline{X} = \$FF - X$	INH(X)	53		1	3					
COM (opr)		M ← $\overline{M} = \$FF - M$	IX1	63	ff	2	6					
COM (opr)		M ← $\overline{M} = \$FF - M$	IX	73		1	5					
CPX (opr)	Compare X with Memory	X − M	IMM	A3	ii	2	2	—	—	↕	↕	↕
			DIR	B3	dd	2	3					
			EXT	C3	hh ll	3	4					
			IX2	D3	ee ff	3	5					
			IX1	E3	ff	2	4					
			IX	F3		1	3					

Table 5-1. Instructions, Addressing Modes, and Execution Times (Sheet 3 of 4)

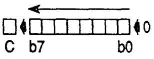
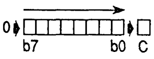
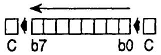
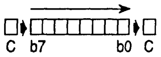
Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (hexadecimal)		Bytes	Cycles	Condition Code				
				Opcode	Operand			H	I	N	Z	C
DEC (opr)	Decrement	M ← M − 01	DIR	3A	dd	2	5	—	—	↕	↕	—
DECA		A ← A − 01	INH(A)	4A		1	3					
DECX	DEX (same as DECX)	X ← X − 01	INH(X)	5A		1	3					
DEC (opr)		M ← M − 01	IX1	6A	ff	2	6					
DEC (opr)		M ← M − 01	IX	7A		1	5					
EOR (opr)	Exclusive OR A with Memory	ACCA ← ACCA ⊕ M	IMM	A8	ii	2	2	—	—	↕	↕	—
			DIR	B8	dd	2	3					
			EXT	C8	hh ll	3	4					
			IX2	D8	ee ff	3	5					
			IX1	E8	ff	2	4					
			IX	F8		1	3					
INC (opr)	Increment	M ← M + 01	DIR	3C	dd	2	5	—	—	↕	↕	—
INCA		A ← A + 01	INH(A)	4C		1	3					
INCX	INX (same as INCX)	X ← X + 01	INH(X)	5C		1	3					
INC (opr)		M ← M + 01	IX1	6C	ff	2	6					
INC (opr)		M ← M + 01	IX	7C		1	5					
JMP (opr)	Jump	PC ← effective address	DIR	BC	dd	2	2	—	—	—	—	—
			EXT	CC	hh ll	3	3					
			IX2	DC	ee ff	3	4					
			IX1	EC	ff	2	3					
			IX	FC		1	2					
JSR (opr)	Jump to Subroutine	PC ← PC + n (n = 1, 2, or 3)	DIR	BD	dd	2	5	—	—	—	—	—
		(SP) ← PCL; SP ← SP − 0001	EXT	CD	hh ll	3	6					
		(SP) ← PCH; SP ← SP − 0001	IX2	DD	ee ff	3	7					
		PC ← effective address	IX1	ED	ff	2	6					
			IX	FD		1	5					
LDA (opr)	Load A from Memory	ACCA ← M	IMM	A6	ii	2	2	—	—	↕	↕	—
			DIR	B6	dd	2	3					
			EXT	C6	hh ll	3	4					
			IX2	D6	ee ff	3	5					
			IX1	E6	ff	2	4					
			IX	F6		1	3					
LDX (opr)	Load X from Memory	X ← M	IMM	AE	ii	2	2	—	—	↕	↕	—
			DIR	BE	dd	2	3					
			EXT	CE	hh ll	3	4					
			IX2	DE	ee ff	3	5					
			IX1	EE	ff	2	4					
			IX	FE		1	3					
LSL (opr)	Logical Shift Left		DIR	38	dd	2	5	—	—	↕	↕	↕
LSLA			INH(A)	48		1	3					
LSLX			INH(X)	58		1	3					
LSL (opr)			IX1	68	ff	2	6					
LSL (opr)			IX	78		1	5					
LSR (opr)	Logical Shift Right		DIR	34	dd	2	5	—	—	0	↕	↕
LSRA			INH(A)	44		1	3					
LSRX			INH(X)	54		1	3					
LSR (opr)			IX1	64	ff	2	6					
LSR (opr)			IX	74		1	5					
MUL	Unsigned Multiply	X:A ← X • A	INH	42		1	11	0	—	—	—	0
NEG (opr)	Negate (2's Complement)	M ← − M (i.e. 00 − M)	DIR	30	dd	2	5	—	—	↕	↕	↕
NEGA		A ← − A	INH(A)	40		1	3					
NEGX		X ← − X	INH(X)	50		1	3					
NEG (opr)		M ← − M	IX1	60	ff	2	6					
NEG (opr		M ← − M	IX	70		1	5					
NOP	No Operation		INH	9D		1	2	—	—	—	—	—
ORA (opr)	Inclusive OR	ACCA ← ACCA + M	IMM	AA	ii	2	2	—	—	↕	↕	—
			DIR	BA	dd	2	3					
			EXT	CA	hh ll	3	4					
			IX2	DA	ee ff	3	5					
			IX1	EA	ff	2	4					
			IX	FA		1	3					

Table 5-1. Instructions, Addressing Modes, and Execution Times (Sheet 4 of 4)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (hexadecimal)		Bytes	Cycles	Condition Code				
				Opcode	Operand			H	I	N	Z	C
ROL (opr) ROLA ROLX ROL (opr) ROL (opr)	Rotate Left through Carry		DIR INH(A) INH(X) IX1 IX	39 49 59 69 79	dd ff 	2 1 1 2 1	5 3 3 6 5	—	—	↕	↕	↕
ROR (opr) RORA RORX ROR (opr) ROR (opr)	Rotate Right through Carry		DIR INH(A) INH(X) IX1 IX	36 46 56 66 76	dd ff 	2 1 1 2 1	5 3 3 6 5	—	—	↕	↕	↕
RSP	Reset Stack Pointer	SP ← $00FF	INH	9C		1	2	—	—	—	—	—
RTI	Return from Interrupt	SP ← SP+0001; CC ← (SP) SP ←SP+0001; ACCA← (SP) SP ← SP+0001; X ← (SP) SP ← SP+0001; PCH ← (SP) SP ← SP+0001; PCL ← (SP)	INH	80		1	9	(From Stack) ↕	 ↕	 ↕	 ↕	 ↕
RTS	Return from Subroutine	SP ← SP+0001; PCH ← (SP) SP ← SP+0001; PCL ← (SP)	INH	81		1	6	—	—	—	—	—
SBC (opr)	Subtract with Carry	ACCA ← ACCA − M − C	IMM DIR EXT IX2 IX1 IX	A2 B2 C2 D2 E2 F2	ii dd hh ll ee ff ff 	2 2 3 3 2 1	2 3 4 5 4 3	—	—	↕	↕	↕
SEC	Set C Bit	C bit ← 1	INH	99		1	2	—	—	—	—	1
SEI	Set I Bit	I bit ← 1	INH	9B		1	2	—	1	—	—	—
STA (opr)	Store A in Memory	M ← ACCA	DIR EXT IX2 IX1 IX	B7 C7 D7 E7 F7	dd hh ll ee ff ff 	2 3 3 2 1	4 5 6 5 4	—	—	↕	↕	—
STOP	Enable $\overline{IRQ}$, Stop Oscillator		INH	8E		1	2	—	0	—	—	—
STX (opr)	Store X in Memory	M ← X	DIR EXT IX2 IX1 IX	BF CF DF EF FF	dd hh ll ee ff ff 	2 3 3 2 1	4 5 6 5 4	—	—	↕	↕	0
SUB (opr)	Subtract	ACCA ← ACCA − M	IMM DIR EXT IX2 IX1 IX	A0 B0 C0 D0 E0 F0	ii dd hh ll ee ff ff 	2 2 3 3 2 1	2 3 4 5 4 3	—	—	↕	↕	↕
SWI	Software Interrupt	PC ← PC+0001 (SP) ← PCL; SP ← SP−0001 (SP) ← PCH; SP ← SP−0001 (SP) ← X; SP ← SP−0001 (SP) ← ACCA; SP ← SP−0001 (SP) ← CC; SP ← SP−0001 I bit ← 1 PCH ← S07FC (vector PCL ← S07FD fetch)	INH	83		1	10	—	1	—	—	—
TAX	Transfer A to X	X ← ACCA	INH	97		1	2	—	—	—	—	—
TST (opr) TSTA TSTX TST (opr) TST (opr)	Test for Negative or Zero	M − 0	DIR INH(A) INH(X) IX1 IX	3D 4D 5D 6D 7D	dd ff 	2 1 1 2 1	4 3 3 5 4	—	—	↕	↕	0
TXA	Transfer X to A	ACCA ← X	INH	9F		1	2	—	—	—	—	—
WAIT	Enable Interrupts, Halt CPU		INH	8F		1	2	—	0	—	—	—

5.1.1 Register/Memory Instructions

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction list.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic Compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR
Multiply	MUL

5.1.2 Read-Modify-Write Instructions

These instructions read a memory location or a register, modify or test 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 since it does not modify the value. Refer to the following list of instructions.

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

5.1.3 Branch Instructions

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
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 Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

5.1.4 Bit Manipulation Instructions

The MCU is capable of setting or clearing any writable bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear, and bit test branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n=0 . . . 7)
Branch if Bit n is Clear	BRCLR n (n=0 . . . 7)
Set Bit n	BSET n (n=0 . . . 7)
Clear Bit n	BCLR n (n=0 . . . 7)

5.1.5 Control Instructions

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

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

5.1.6 Opcode Map Summary

Table 5-2 is an opcode map for the instructions used on the MCU.

5.2 ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single-byte instructions; the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

5.2.1 Immediate

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

5.2.2 Direct

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

5.2.3 Extended

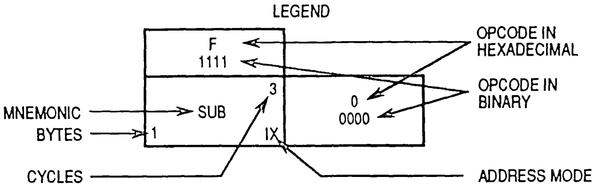
In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

Table 5-2. Opcode Map

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
HI LOW	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 LOW
0 0000	5 BRSET0 3 BTB	5 BSET0 2 BSC	3 BRA 2 REL	5 NEG 2 DIR	3 NEGA 1 INH	3 NEGX 1 INH	6 NEG 2 IX1	5 NEG 1 IX	9 RTI 1 INH		2 SUB 2 IMM	3 SUB 2 DIR	4 SUB 3 EXT	5 SUB 3 IX2	4 SUB 2 IX1	3 SUB 1 IX	0 0000
1 0001	5 BRCLR0 3 BTB	5 BCLR0 2 BSC	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 BTB	5 BSET1 2 BSC	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 BTB	5 BCLR1 2 BSC	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 BTB	5 BSET2 2 BSC	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 BTB	5 BCLR2 2 BSC	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 BTB	5 BSET3 2 BSC	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 BTB	5 BCLR3 2 BSC	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 BTB	5 BSET4 2 BSC	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 BTB	5 BCLR4 2 BSC	3 BHCS 2 REL	5 ROL 2 DIR	3 ROLA 1 INH	3 ROLX 1 INH	6 ROL 2 IX1	5 ROL 1 IX		2 SEC 1 INH	2 ADC 2 IMM	3 ADC 2 DIR	4 ADC 3 EXT	5 ADC 3 IX2	4 ADC 2 IX1	3 ADC 1 IX	9 1001
A 1010	5 BRSET5 3 BTB	5 BSET5 2 BSC	3 BPL 2 REL	5 DEC 2 DIR	3 DECA 1 INH	3 DECX 1 INH	6 DEC 2 IX1	5 DEC 1 IX		2 CLI 1 INH	2 ORA 2 IMM	3 ORA 2 DIR	4 ORA 3 EXT	5 ORA 3 IX2	4 ORA 2 IX1	3 ORA 1 IX	A 1010
B 1011	5 BRCLR5 3 BTB	5 BCLR5 2 BSC	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 BTB	5 BSET6 2 BSC	3 BMC 2 REL	5 INC 2 DIR	3 INCA 1 INH	3 INCX 1 INH	6 INC 2 IX1	5 INC 1 IX		2 RSP 1 INH		2 JMP 2 DIR	3 JMP 3 EXT	4 JMP 3 IX2	3 JMP 2 IX1	2 JMP 1 IX	C 1100
D 1101	5 BRCLR6 3 BTB	5 BCLR6 2 BSC	3 BMS 2 REL	4 TST 2 DIR	3 TSTA 1 INH	3 TSTX 1 INH	5 TST 2 IX1	4 TST 1 IX		2 NOP 1 INH	6 BSR 2 REL	5 JSR 2 DIR	6 JSR 3 EXT	7 JSR 3 IX2	6 JSR 2 IX1	5 JSR 1 IX	D 1101
E 1110	5 BRSET7 3 BTB	5 BSET7 2 BSC	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 BTB	5 BCLR7 2 BSC	3 BIH 2 REL	5 CLR 2 DIR	3 CLRA 1 INH	3 CLRX 1 INH	6 CLR 2 IX1	5 CLR 1 IX	2 WAIT 1 INH	2 TXA 1 INH		4 STX 2 DIR	5 STX 3 EXT	6 STX 3 IX2	5 STX 2 IX1	4 STX 1 IX	F 1111

Abbreviations for Address Modes:

INH	Inherent	REL	Relative
A	Accumulator	BSC	Bit Set/Clear
X	Index Register	BTB	Bit Test and Branch
IMM	Immediate	IX	Indexed (No Offset)
DIR	Direct	IX1	Indexed, 1 Byte (8-Bit) Offset
EXT	Extended	IX2	Indexed, 2 Byte (16-Bit) Offset



5.2.4 Relative

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed offset byte (which is the last byte of the instruction) is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from −128 to +124 from the address of the next opcode. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

5.2.5 Indexed, No Offset

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This 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.

5.2.6 Indexed, 8-Bit Offset

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this two-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510. $1FE is the last location which can be accessed in this way.

5.2.7 Indexed, 16-Bit Offset

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

5.2.8 Bit Set/Clear

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct address of the byte in which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single two-byte instruction.

5.2.9 Bit Test and Branch

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear), is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from −128 to +127 from the address of the next opcode. The state of the tested bit is also transferred to the carry bit of the condition code register.

5.2.10 Inherent

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register and/or accumulator as well as the control instructions with no other arguments are included in this mode. These instructions are one byte long.

SECTION 6
ELECTRICAL SPECIFICATIONS

This section contains the electrical specifications and associated timing information for the MC68HC05J1.

6.1 MAXIMUM RATINGS (Voltages referenced to V_{SS})

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 $2 \times V_{DD}+0.3$	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range MC68HC05J1P, DW MC68HC05J1CP, CDW	T_A	T_L to T_H 0 to +70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C

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

6.2 THERMAL CHARACTERISTICS

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

$V_{DD}=4.5$ V

Pins	R1	R2	C
PA7–PA0, PB5–PB0	3.26 kΩ	2.38 kΩ	50 pF

$V_{DD}=3.0$ V

Pins	R1	R2	C
PA7–PA0, PB5–PB0	10.91 kΩ	6.32 kΩ	50 pF

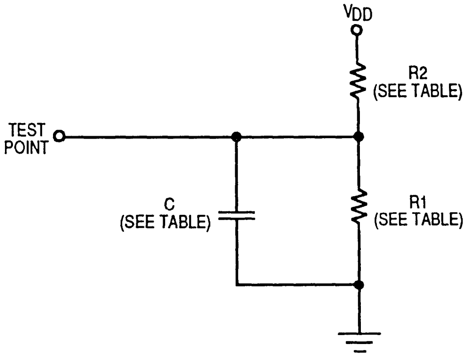


Figure 6-1. Equivalent Test Load

6.3 POWER CONSIDERATIONS

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

$$T_J = T_A + (P_D \bullet R_{\theta JA}) \qquad (1)$$

where:

T_A = Ambient Temperature, °C
$R_{\theta JA}$ = Package Thermal Resistance,
Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{I/O}$
P_{INT} = $I_{DD} \times V_{DD}$, Watts — Chip Internal Power
$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

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

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

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

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

$$K = P_D \bullet (T_A + 273°C) + R_{\theta JA} \bullet P_D^2 \qquad (3)$$

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.

6.4 DC ELECTRICAL CHARACTERISTICS

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

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage, $I_{Load} \leq 10.0\ \mu A$	V_{OL}	—	—	0.1	V
	V_{OH}	$V_{DD}-0.1$	—	—	
Output High Voltage (see Figure 6-2) ($I_{Load}=-0.8$ mA) PA7–PA0, PB5–PB0	V_{OH}	$V_{DD}-0.8$	—	—	V
Output Low Voltage (see Figure 6-3) ($I_{Load}=1.6$ mA) PA7–PA0, PB5–PB0	V_{OL}	—	—	0.4	V
Input High Voltage PA7–PA0, PB5–PB0, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA7–PA0, PB5–PB0, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes)	I_{DD}				
Run (see Figures 6-4 and 6-5)		—	2.5	5.0	mA
Wait (see Figures 6-4 and 6-5)		—	1.2	2.75	mA
Stop (see Figure 6-5)					
25°C		—	2.0	30	μA
−40° to 85°C		—	—	100	μA
I/O Ports Hi-Z Leakage Current PA7–PA0, PB5–PB0	I_{IL}	—	—	±10	μA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, OSC1	I_{in}	—	—	±1	μA
Capacitance					pF
Ports (as Input or Output)	C_{out}	—	—	12	
$\overline{\text{RESET}}$, $\overline{\text{IRQ}}$	C_{in}	—	—	8	

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, 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.

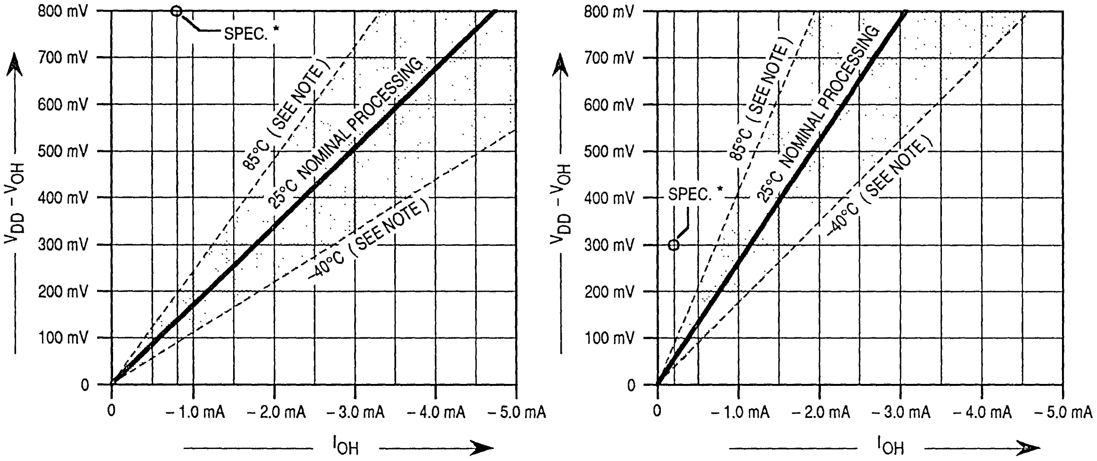
6.5 DC ELECTRICAL CHARACTERISTICS

(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, $I_{Load} \leq 10.0\ \mu A$	V_{OL} V_{OH}	— $V_{DD} - 0.1$	— —	0.1 —	V
Output High Voltage (see Figure 6-2) ($I_{Load} = -0.2$ mA) PA7–PA0, PB5–PB0	V_{OH}	$V_{DD} - 0.3$	—	—	V
Output Low Voltage (see Figure 6-3) ($I_{Load} = 0.4$ mA) PA7–PA0, PB5–PB0	V_{OL}	—	—	0.3	V
Input High Voltage PA7–PA0, PB5–PB0, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input Low Voltage PA7–PA0, PB5–PB0, $\overline{\text{IRQ}}$, $\overline{\text{RESET}}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
Data Retention Mode (0° to 70°C)	V_{RM}	2.0	—	—	V
Supply Current (see Notes) Run (see Figures 6-4 and 6-6) Wait (see Figures 6-4 and 6-6) Stop (see Figure 6-6) 25°C −40° to +85°C	I_{DD}	 — — — —	 0.7 0.5 1.0 —	 2.5 1.0 20 50	 mA mA μA μA
I/O Ports Hi-Z Leakage Current PA7–PA0, PB5–PB0	I_{IL}	—	—	±10	μA
Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$, OSC1	I_{in}	—	—	±1	μA
Capacitance Ports (as Input or Output) $\overline{\text{RESET}}$, $\overline{\text{IRQ}}$	 C_{out} C_{in}	 — —	 — —	 12 8	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} = 2.0 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, 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.

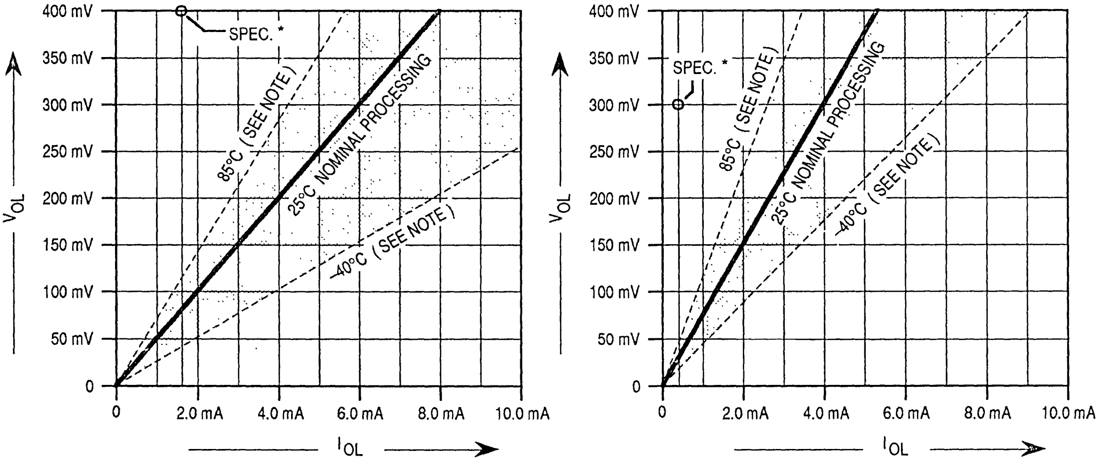


* At V_{DD} = 5.0 V, devices are specified and tested for $(V_{DD} - V_{OH}) \leq 800$ mV @ $I_{OH} = -0.8$ mA.

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

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

Figure 6-2. Typical High Side Driver Characteristics



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

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

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

Figure 6-3. Typical Low Side Driver Characteristics

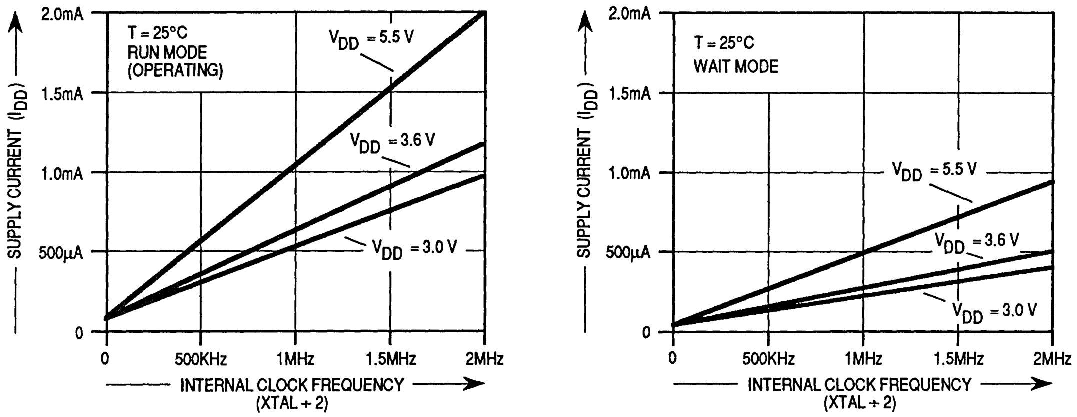
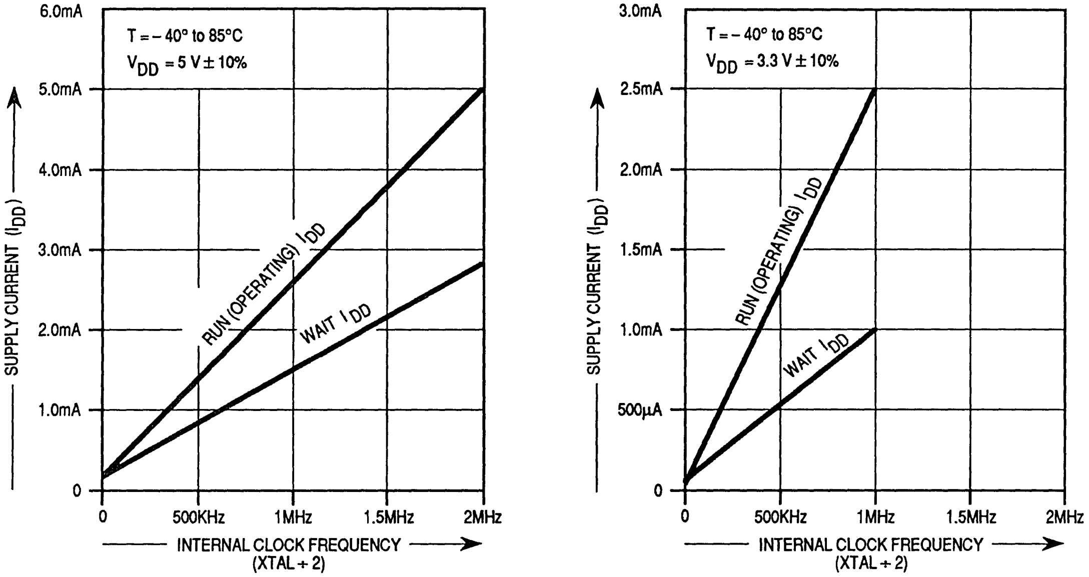


Figure 6-4. Typical Supply Current vs. Internal Clock Frequency



NOTE: Maximum STOP I_{DD} = 100 μA when V_{DD} = 5 V.

NOTE: Maximum STOP I_{DD} = 50 μA when V_{DD} = 3.3 V.

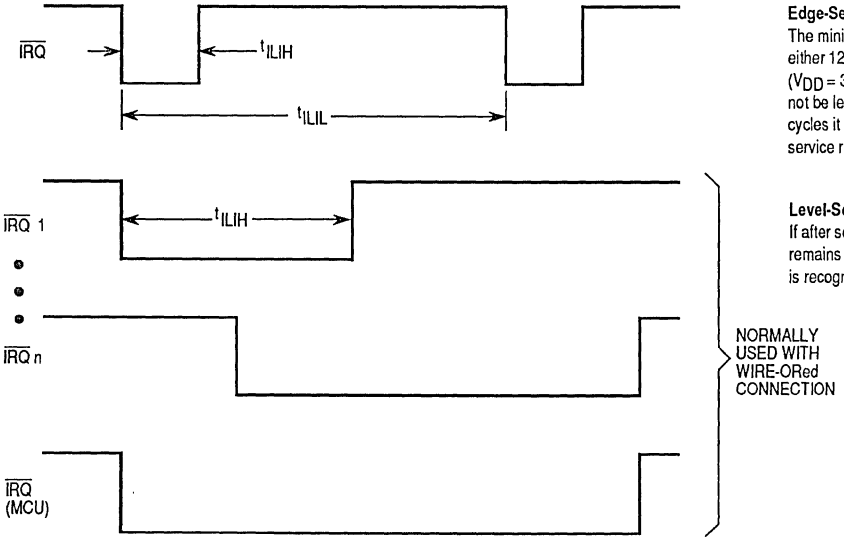
Figure 6-5. Maximum Supply Current vs. Internal Clock Frequency

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

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option RC Option	f_{osc}	 — dc —	 4.2 4.2 4.2	MHz
Internal Operating Frequency Crystal (f_{osc} ÷ 2) External Clock (f_{osc} ÷ 2) RC Option (f_{osc} ÷ 2)	f_{op}	 — dc —	 2.1 2.1 2.1	MHz
Cycle Time (see Figure 6-8)	t_{cyc}	476	—	ns
$\overline{\text{RESET}}$ Pulse Width (see Figure 6-8)	t_{RL}	1.5	—	t_{cyc}
Timer Resolution**	t_{RESL}	4.0	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 3-4)	t_{ILIH}	125	—	ns
Interrupt Pulse Period (see Figure 3-4)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

*The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 19 t_{cyc}.

**Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.


$\overline{\text{IRQ}}$
t_{ILIH}
t_{ILIL}
$\overline{\text{IRQ}}$ 1
t_{ILIH}
$\overline{\text{IRQ}}$ n
$\overline{\text{IRQ}}$ (MCU)
NORMALLY USED WITH WIRE-ORed CONNECTION

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 $\overline{\text{IRQ}}$ remains low, then the next interrupt is recognized.

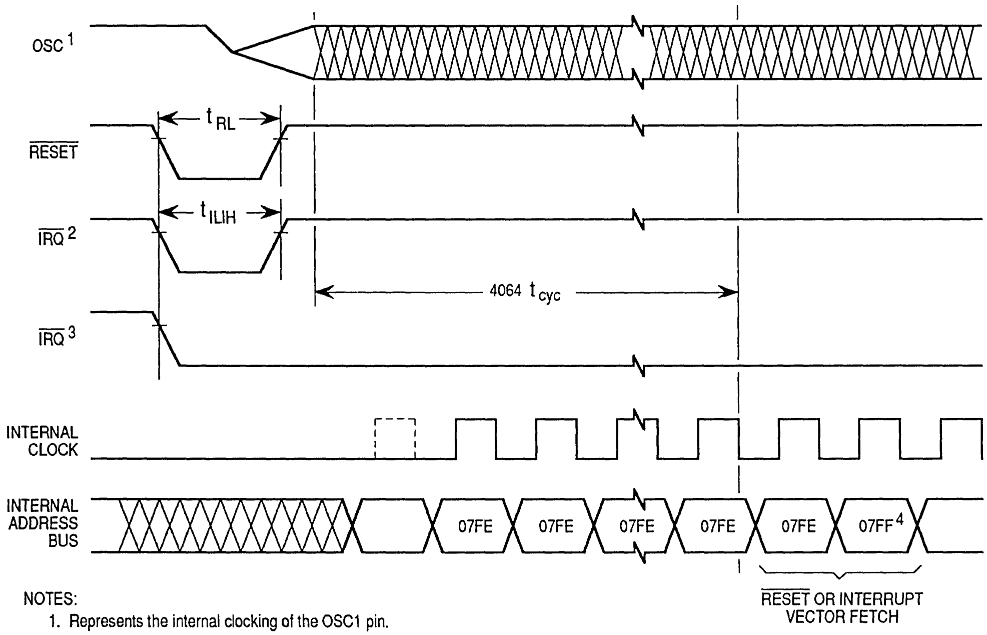
Figure 6-6. External Interrupt Mode Diagram

6.7 CONTROL TIMING (V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option RC Option	f_{osc}	 — dc —	 2.0 2.0 2.0	MHz
Internal Operating Frequency Crystal (f_{osc} ÷ 2) External Clock (f_{osc} ÷ 2) RC Option (f_{osc} ÷ 2)	f_{op}	 — dc —	 1.0 1.0 1.0	MHz
Cycle Time (see Figure 6-8)	t_{cyc}	1000	—	ns
$\overline{\text{RESET}}$ Pulse Width — Excluding Powerup (see Figure 6-8)	t_{RL}	1.5	—	t_{cyc}
Timer Resolution**	t_{RESL}	4.0	—	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered) (see Figure 3-3)	t_{ILIH}	250	—	ns
Interrupt Pulse Period (see Figure 3-3)	t_{ILIL}	*	—	t_{cyc}
OSC1 Pulse Width	t_{OH}, t_{OL}	400	—	ns

* The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 19 t_{cyc}.

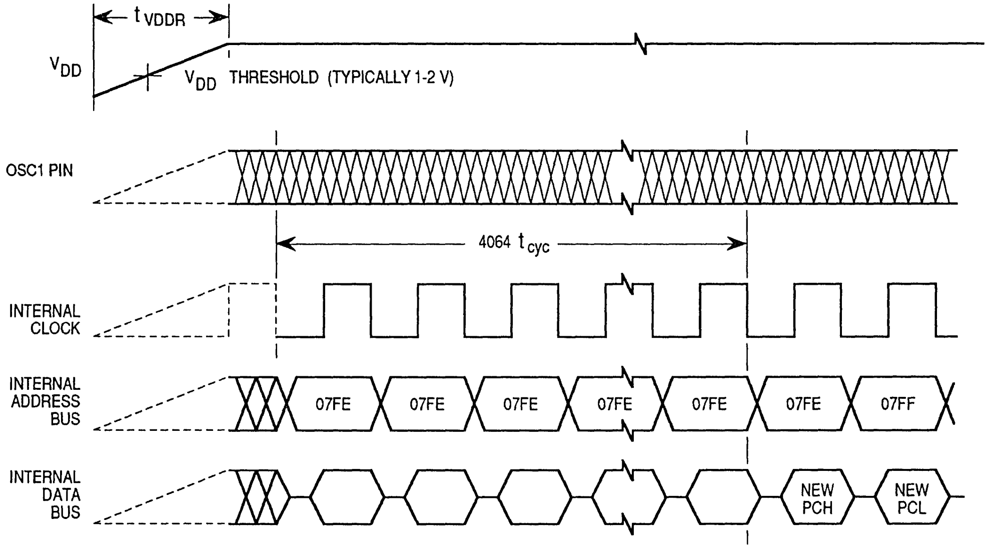
** Since a 2-bit prescaler in the timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.



NOTES:

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

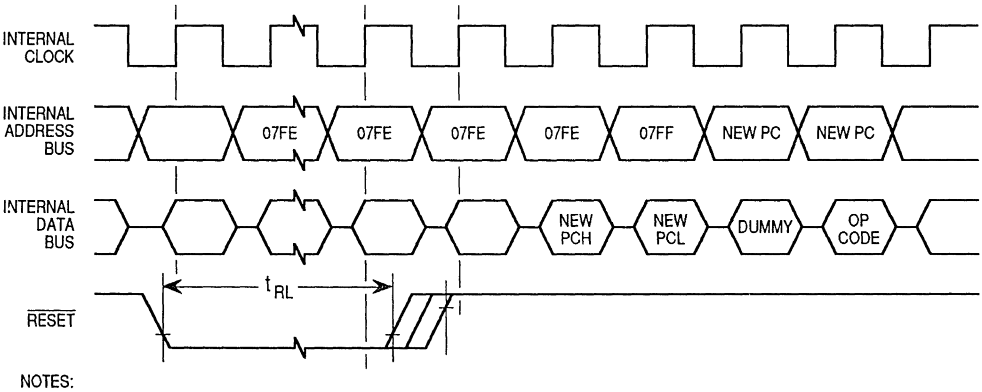
Figure 6-7. Stop Recovery Timing Diagram



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 6-8. Power-On Reset



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 $\overline{\text{RESET}}$ initiates the reset sequence.

Figure 6-9. External Reset Sequence

SECTION 7
MECHANICAL DATA

This section provides package dimensions for the MC68HC05J1.

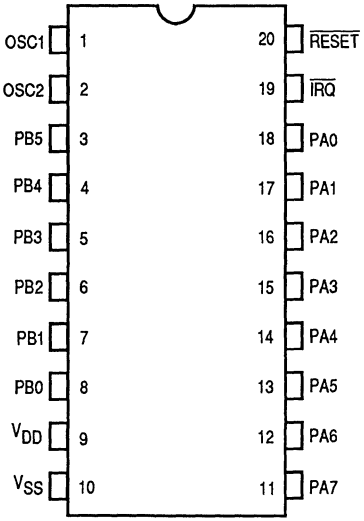
7.1 MECHANICAL OPTION INFORMATION

The following table provides ordering information pertaining to the package type, temperature, and MC order numbers for the MC68HC05J1 device.

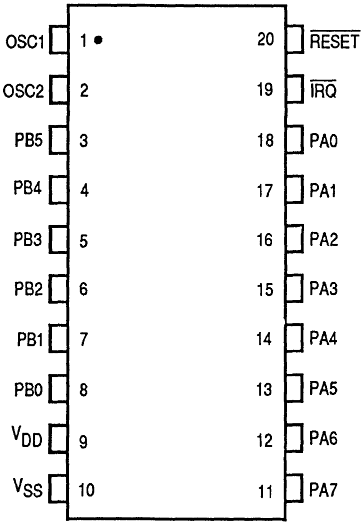
Package Type	Temperature	MC Order Number
Plastic		
(P Suffix)	0°C to +70°C	MC68HC05J1P
(CP Suffix)	−40°C to +85°C	MC68HC05J1CP
SOIC		
(DW Suffix)	0°C to +70°C	MC68HC05J1DW
(CDW Suffix)	−40°C to +85°C	MC68HC05J1CDW

7.2 PIN ASSIGNMENTS

7.2.1 20-Pin Dual-in-Line Package

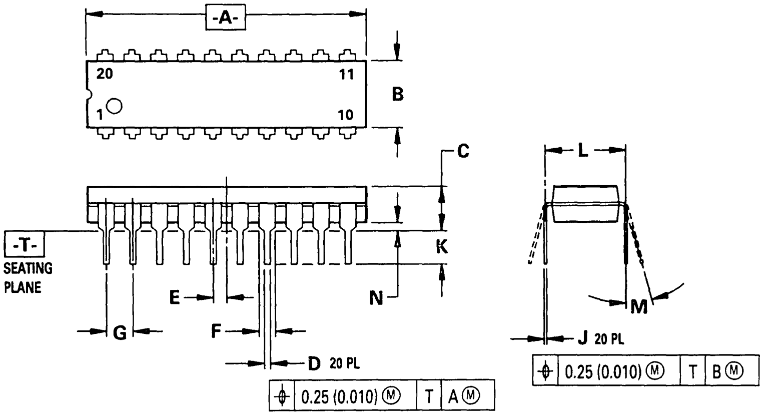


7.2.2 20-Pin SOIC Package



7.3 PACKAGE DIMENSIONS

PLASTIC DIP
CASE 738-03

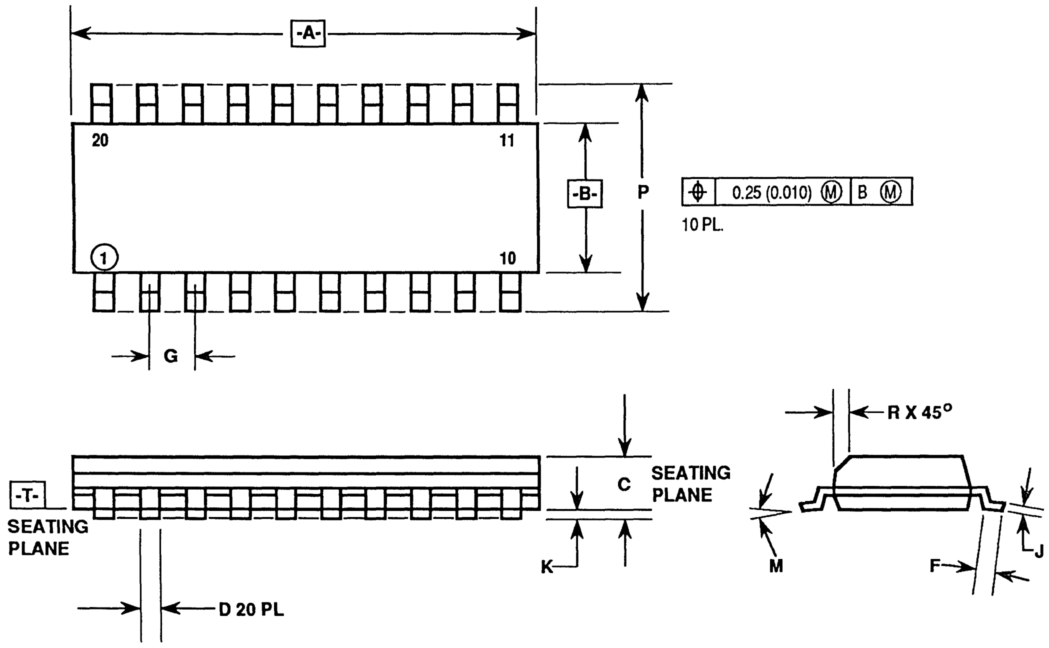


NOTES:

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION "L" TO CENTER OF LEAD WHEN FORMED PARALLEL.
4. DIMENSION "B" DOES NOT INCLUDE MOLD FLASH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	25.66	27.17	1.010	1.070
B	6.10	6.60	0.240	0.260
C	3.81	4.57	0.150	0.180
D	0.39	0.55	0.015	0.022
E	1.27 BSC		0.050 BSC	
F	1.27	1.77	0.050	0.070
G	2.54 BSC		0.100 BSC	
J	0.21	0.38	0.008	0.015
K	2.80	3.55	0.110	0.140
L	7.62 BSC		0.300 BSC	
M	0°	15°	0°	15°
N	0.51	1.01	0.020	0.040

SOIC
CASE 751D-03



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 DIM: MILLIMETER.
4. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION.
5. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
6. 751D-01, AND 02 OBSOLETE, NEW STANDARD 751D-03.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	12.65	12.95	0.499	0.510
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

SECTION 8
ORDERING INFORMATION

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

8.1 MCU ORDERING FORMS

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

- A current MCU ordering form that is **completely filled out** (Contact your Motorola sales office for assistance.)
- A copy of the customer specification if the customer specification deviates from the Motorola specification for the MCU
- Customer's application program on one of the media listed in **8.2 Application Program Media**

The current MCU ordering from is also available through the Motorola Freeware Bulletin Board Service (BBS). The number is (512) 891-FREE. After making the connection, type bbs in lowercase letters and press the return key to start the BBS software.

8.2 APPLICATION PROGRAM MEDIA

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

- Macintosh®[1] 3-1/2" diskette (double-sided 800K or double-sided high-density 1.4M)
- MS-DOS®[2] or PC-DOS®[3] 3-1/2" (double-sided 720K or double-sided high-density 1.44M) diskette
- MS-DOS® or PC-DOS® 5-1/4" (double-sided double-density 360K or double-sided high-density 1.2M) diskette
- EPROM(s) 2716, 2732, 2764, 27128, 27256, 27512 (depending on the size of the memory map of the MCU)

Use positive logic for data and addresses.

8.2.1 Diskettes

If submitting the applications program on a diskette, clearly label the diskette with the following information:

- Customer name
- Customer part number
- Project or product name
- Filename of object code
- Date
- Operating system that formatted diskette
- Formatted capacity of diskette

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

[1] Macintosh is a registered trademark of Apple Computer, Inc.

[2] MS-DOS is a registered trademark of Microsoft, Inc.

[3] PC-DOS is a registered trademark of International Business Machines Corporation.

NOTE

Begin the application program at the first user ROM location. Program addresses must correspond exactly to the available on-chip user ROM addresses as shown in the memory map. **Write $00 in all non-user ROM locations or leave all non-user ROM locations blank.** See the current MCU ordering form for additional requirements.

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

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

8.2.2 EPROMs

If submitting the application program in an EPROM, clearly label the EPROM with the following information:

- Customer name
- Customer part number
- Checksum
- Project or product name
- Date

NOTE

Begin the application program at the first user ROM location. Program addresses must correspond exactly to the available on-chip user ROM addresses as shown in the memory map. **Write $00 in all non-user ROM locations.** See the current MCU ordering form for additional requirements.

Submit the application program in one EPROM large enough to contain the entire memory map. If the memory map has two user ROM areas with the same addresses, then write the two areas on separate EPROMs. Label the EPROMs with the addresses they contain.

Pack EPROMs securely in a conductive IC carrier for shipment. Do not use Styrofoam.

8.3 ROM PROGRAM VERIFICATION

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

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

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

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

8.4 RVUs (ROM VERIFICATION UNITS)

After receiving the signed listing verify form, Motorola manufactures a custom photographic mask. The mask contains the customer's application program, and is used to process silicon wafers. The application program cannot be changed after the manufacture of the mask begins. Motorola then produces ten MCUs, called RVUs, and sends the RVUs to the customer. RVUs are usually packaged in unmarked ceramic and tested with 5 Vdc at room temperature. RVUs are not tested to environmental extremes because their sole purpose is to demonstrate that the customer's user ROM pattern was properly implemented.

The ten RVUs are free of charge with the minimum order quantity but are not production parts. RVUs are not guaranteed by Motorola Quality Assurance.

Literature Distribution Centers:

USA: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Centre; 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

A25283-1 PRINTED IN USA 12/91 IMPERIAL LITHO 83170 10,000 MCU YGACAA