68HC05J5

SPECIFICATION
(General Release)

Technical Operations Taiwan
CSIC Group

TABLE OF CONTENTS

Section **Page**

SECTION 1
GENERAL DESCRIPTION

SECTION 2
MEMORY

SECTION 3
CENTRAL PROCESSING UNIT

SECTION 4
INTERRUPTS

TABLE OF CONTENTS

Section **Page**

SECTION 5
RESETS

SECTION 6
MODES OF OPERATION

SECTION 7
INPUT/OUTPUT PORTS

TABLE OF CONTENTS

Section **Page**

SECTION 8
MULTI-FUNCTION TIMER

SECTION 9
INSTRUCTION SET

SECTION 10
ELECTRICAL SPECIFICATIONS

SECTION 11
MECHANICAL SPECIFICATION

TABLE OF CONTENTS

Section **Page**

LIST OF FIGURES

LIST OF FIGURES

Figure	Title	Page

LIST OF TABLES

LIST OF TABLES

SECTION 1
GENERAL DESCRIPTION

The MC68HC05J5 HCMOS Microcontroller is a member of the MC68HC05 Family of low-cost single-chip 8-bit Microcontroller Units (MCUs). The MC68HC05J5 is an enhanced version of the MC68HC05J1A, which includes high sink current port pins, slow output transition port pins, an extra interrupt on a port pin, low-voltage-reset, and a tight tolerance RC oscillator option.

The MC68HC05J5 is available in 20-pin and 16-pin packages. (The 16-pin package has four less I/O port lines than the 20-pin package.)

Although the MC68HC05J5 is an enhanced version of the MC68HC05J1A, their pin assignments are different.

1.1 FEATURES

- Industry standard M68HC05 CPU core
- Fully static operation with no minimum clock speed
- 1240 bytes of user ROM
- 64 bytes of user RAM
- 14 bidirectional I/O pins
- On-chip Oscillator: Crystal/Resonator Oscillator or RC Oscillator with only one external resistor required
- Low Voltage Reset (LVR)
- Hardware mask and flag for external interrupts
- 15-bit Multi-Function Timer
- Computer Operating Properly (COP) watchdog
- Power saving STOP and WAIT modes
- Illegal Address Reset (ILADR)
- Available in 16-pin PDIP, 16-pin SOIC, 20-pin PDIP, and 20-pin SOIC packages

1.2 MASK OPTIONS

The following mask options are available:

MASK	OPTION
On-chip oscillator	[Crystal/Resonator] or [RC]
Crystal/resonator feedback resistor	[Connected] or [Disconnected]
STOP instruction convert to WAIT	[Enabled] or [Disabled]
PA0-PA3 external interrupt capability	[Enabled] or [Disabled]
External interrupt pins ($\overline{IRQ}$, PA0-PA3)	[Edge-triggered] or [Edge and level triggered]
Port A and Port B pull-down/pull-up resistors	[Connected] or [Disconnected]
COP Watchdog Timer	[Enabled] or [Disabled]
Low Voltage Reset	[Enabled] or [Disabled]

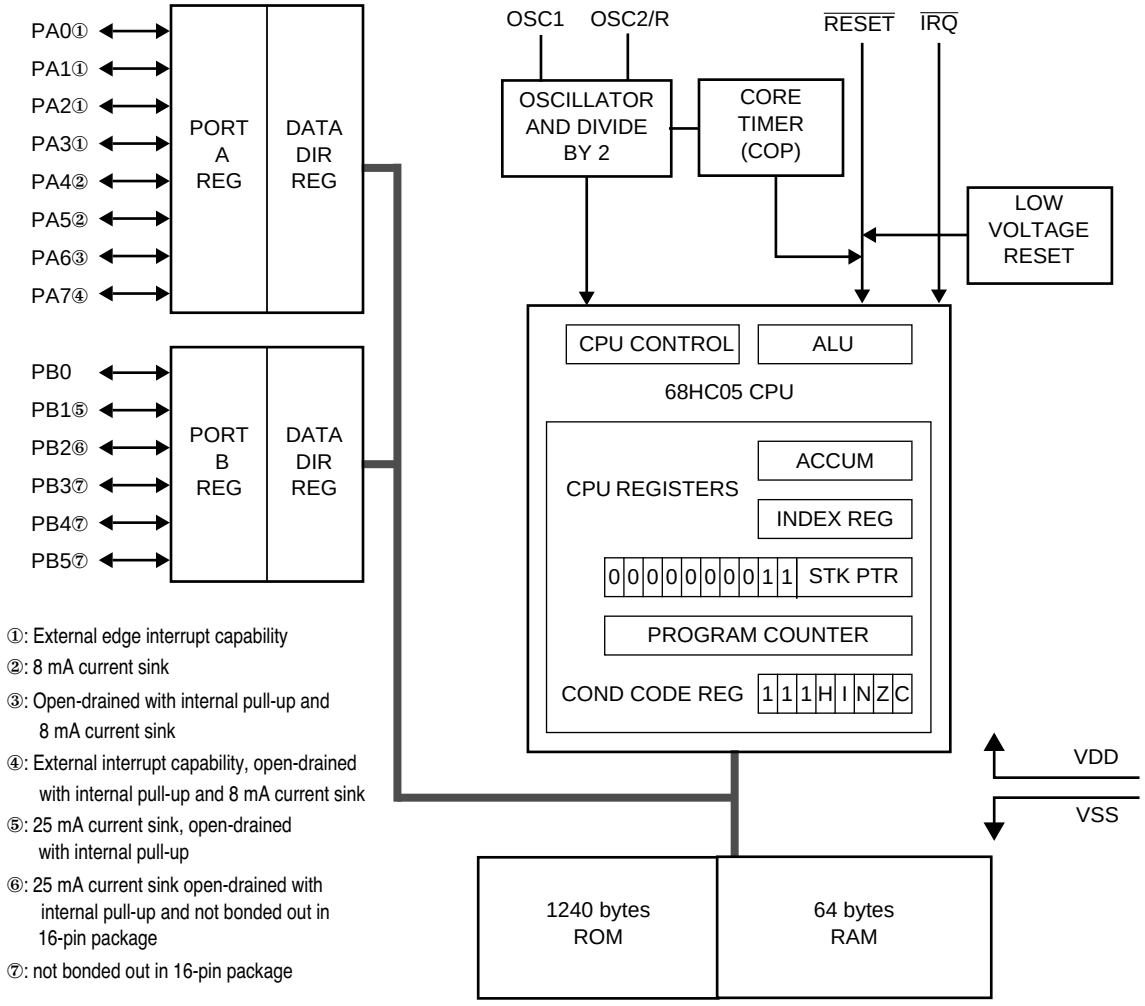


Figure 1-1. MC68HC05J5 Block Diagram

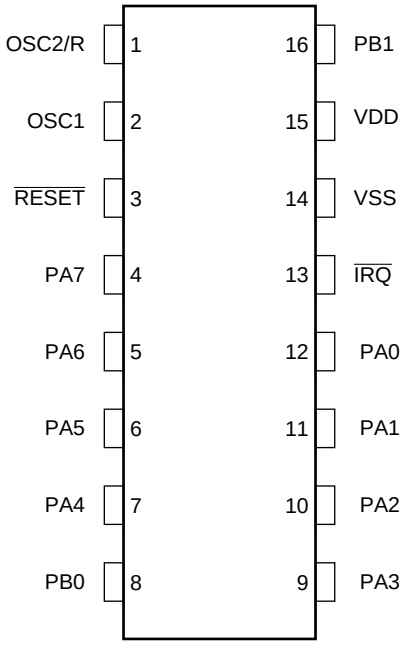


Figure 1-2. Pin Assignments for 16-Pin Package

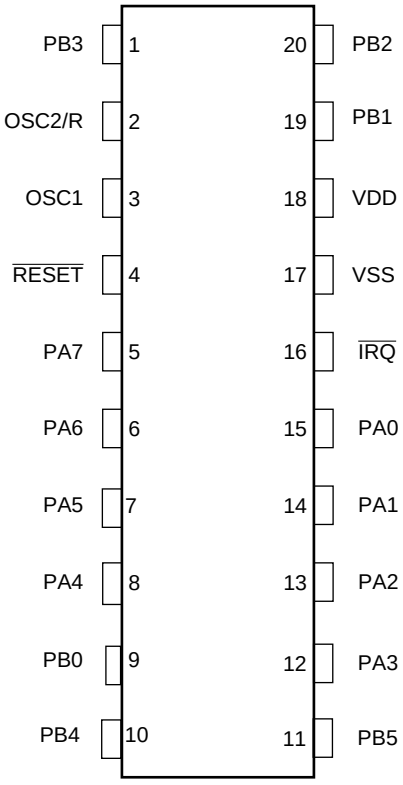


Figure 1-3. Pin Assignments for 20-Pin Package

1.3 FUNCTIONAL PIN DESCRIPTION

The following paragraphs give a description of the general function of each pin assigned in **Figure 1-2** and **Figure 1-3**.

1.3.1 V_{DD} AND V_{SS}

Power is supplied to the MCU through V_{DD} and V_{SS}. V_{DD} is the positive supply, and V_{SS} is ground. The MCU operates from a single 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 should be taken to provide good power supply bypassing at the MCU by using bypass capacitors with good high-frequency characteristics that are positioned as close to the MCU as possible. Bypassing requirements vary, depending on how heavily the MCU pins are loaded.

1.3.2 OSC1, OSC2/R

The OSC1 and OSC2/R pins are the connections for the on-chip oscillator. The OSC1 and OSC2/R pins can accept the following sets of components:

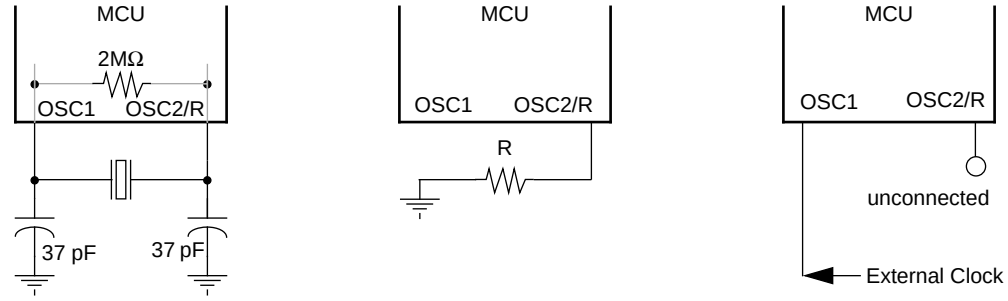
1. A crystal as shown in **Figure 1-4**(a)
2. A ceramic resonator as shown in **Figure 1-4**(a)
3. An external resistor as shown in **Figure 1-4**(b)
4. An external clock signal as shown in **Figure 1-4**(c)

The frequency, f_{OSC}, of the oscillator or external clock source is divided by two to produce the internal operating frequency, f_{OP}. The type of oscillator is selected by a mask option. An internal 2MΩ resistor may be selected between OSC1 and OSC2/R by a mask option (crystal/ceramic resonator mode only).

If the RC oscillator option is selected, OSC1 pin should be connected to a known logic level, either one or zero.

1.3.2.1 Crystal Oscillator

The circuit in **Figure 1-4**(a) shows a typical oscillator circuit for an AT-cut, parallel resonant crystal. The crystal manufacturer's recommendations should be followed, as the crystal parameters determine the external component values required to provide maximum stability and reliable start-up. The load capacitance values used in the oscillator circuit design should include all stray capacitances. The crystal and components should be mounted as close as possible to the pins for start-up stabilization and to minimize output distortion. An internal start-up resistor of approximately 2 MΩ is provided between OSC1 and OSC2/R for the crystal type oscillator as a mask option.



(a) Crystal or ceramic resonator connection

(b) RC oscillator connection

(c) External clock source connection

Figure 1-4. Oscillator Connections

1.3.2.2 Ceramic Resonator Oscillator

In cost-sensitive applications, a ceramic resonator can be used in place of the crystal. The circuit in **Figure 1-4**(a) can be used for a ceramic resonator. The resonator manufacturer's recommendations should be followed, as the resonator parameters determine the external component values required for maximum stability and reliable starting. The load capacitance values used in the oscillator circuit design should include all stray capacitances. The ceramic resonator and components should be mounted as close as possible to the pins for start-up stabilization and to minimize output distortion. An internal start-up resistor of approximately 2 MΩ is provided between OSC1 and OSC2/R for the ceramic resonator type oscillator as a mask option.

1.3.2.3 RC Oscillator

The lowest cost oscillator is the RC oscillator configuration. With this option an external resistor is connected between OSC2/R pin and the V_{SS} pin as shown in **Figure 1-4**(b). The typical operating frequency f_{OSC} is set at 4 MHz with the external R tied to V_{SS}. The internal start-up resistor of approximately 2 MΩ is not connected between OSC1 and OSC2/R for the mask option of the RC type oscillator.

The tolerance of this RC oscillator is guaranteed to be no greater than ±15% at the specified conditions of 0 °C to 40 °C and 5V ±10% V_{DD} providing that the tolerance of the external resistor R is at most ±1% and the center frequency range is from 3.8MHz to 4.2MHz. The center frequency is the nominal operating frequency of the RC oscillator and can be adjusted by adjusting the external R value to change the internal VCO charging current.

In order to obtain an oscillator clock with the best possible tolerance, the external resistor connected to the OSC2/R pin should be grounded as close to the VSS pin as possible and the other terminal of this external resistor should be connected as close to the OSC2/R pin as possible.

1.3.2.4 External Clock

An external clock from another CMOS-compatible device can be connected to the OSC1 input, with the OSC2/R input not connected, as shown in **Figure 1-4**(c). This configuration is possible only when the crystal/ceramic resonator mask option is selected.

1.3.3 $\overline{\text{RESET}}$

This is an I/O pin. This pin can be used as an input to reset the MCU to a known start-up state by pulling it to the low state. The $\overline{\text{RESET}}$ pin contains a steering diode to discharge any voltage on the pin to V_{DD}, when the power is removed. An internal pull-up is also connected between this pin and V_{DD}. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger to improve its noise immunity as an input. This pin is an output pin if LVR triggers an internal reset.

1.3.4 $\overline{\text{IRQ}}$

This input pin drives the asynchronous IRQ interrupt function of the CPU. The IRQ interrupt function has a mask option to provide either only negative edge-sensitive triggering or both negative edge-sensitive and low level-sensitive triggering. If the option is selected to include level-sensitive triggering, the $\overline{\text{IRQ}}$ input requires an external resistor to V_{DD} for "wired-OR" operation, if desired. The $\overline{\text{IRQ}}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity.

NOTE

Each of the PA0 to PA3 I/O pins may be connected as an OR function with the IRQ interrupt function by a mask option. This capability allows keyboard scan applications where the transitions or levels on the I/O pins will behave the same as the $\overline{\text{IRQ}}$ pin, except for the inverted phase. The edge or level sensitivity selected by a separate mask option for the $\overline{\text{IRQ}}$ pin also applies to the I/O pins OR'ed to create the IRQ signal. Besides, PA7 also has falling-edge only interrupt capability whose functionality is controlled by another set of register bits.

1.3.5 PA0-PA7

These eight I/O lines comprise Port A. PA6 and PA7 are open-drained pins with pull-up devices whereas PA0 to PA5 are push-pull pins with pull-down devices. PA4 to PA7 are also capable of sinking 8 mA.

The state of any pin is software programmable and all Port A lines are configured as inputs during power-on or reset. The lower four I/O pins (PA0 to PA3) can be connected via an internal OR gate to the IRQ interrupt function enabled by a mask option. Another independent interrupt source comes from the falling edge on PA7. PA7 interrupt source is associated with a second set of interrupt control/status bits. All Port A pins except PA6 and PA7 have software programmable pull-down devices also provided by a mask option. PA6 and PA7 pins have software programmable pull-up devices also provided by the same mask option. Pull-up

devices on PA6 and PA7 once enabled are always enabled regardless of pin direction configuration, unlike pull-down devices on PA0 to PA5 which are activated only when these pins are configured as input pins.

PA6 and PA7 pins, when configured as output pins, also have slow output falling-edge transition feature to reduce EMI. The falling-edge transition time is tentatively set at 250ns typical at a specified load of 500pF, assuming the bus rate is 2MHz. The slow transition output feature of PA6 and PA7, along with that of PB1 and PB2, can be enabled or disabled by software. Both PA6 and PA7 pins have Schmitt trigger input for better noise immunity. V_{IH} and V_{IL} are specified at 2.4V and 0.8V, respectively.

The slow transition feature of PA6 and PA7 pins can be enabled or disabled by software. Once enabled, slow transition feature is applied to both pins while in output mode.

1.3.6 PB0-PB5

NOTE

I/O lines PB2 to PB5 are not available on the 16-pin package.

These six I/O lines comprise Port B. PB0, PB3 to PB5 are push-pull I/O lines with pull-down resistor. PB1 and PB2 are open-drain I/O lines with pull-up resistor.

The state of any line is software programmable and is configured as an input during power-on or reset. I/O lines PB1 and PB2 have software programmable pull-up device whereas PB0, PB3 to PB5 have software programmable pull-down device, by a mask option. Pull-up devices on PB1 and PB2 lines once enabled are always enabled regardless of pin direction configuration; unlike pull-down devices on PB0, PB3-PB5 lines, which are activated only when the pin is configured as input pin.

Similar to PA6 and PA7, PB1 also has a slow output falling transition feature when configured as an output line. PB1 has 25mA sink capability at 0.5V V_{OL}.

PB2 output is one clock cycle (250ns if bus rate is 2MHz) late than other I/O pins if slow output transition feature is enabled. PB2 has 25mA sink capability at 0.5V V_{OL}.

NOTE

For the 16-pin package, PB1 and PB2 are bonded to the same pin and is labelled PB1. This PB1 has 50mA sink capability is slow transition feature is enabled and if they are written with the same value at the same write cycle. The falling transition time of PB1 is set at 250ns typical at a specified load of 50pF, assuming that the bus rate is 2MHz. The slow transition feature on this PB1 pin is longer than PB1 pin for the 20-pin package.

NOTE

If Port Data Register PB1 and PB2 are not written with the same value, PB1 pin on the 16-pin package will sink 25 mA only and the output transition time will be shorter.

SECTION 2
MEMORY

2.1 MEMORY MAP

The MC68HC05J5 has 2K-bytes of addressable memory consisting 32 bytes of I/O, 64 bytes of user RAM, and 1240 bytes of user ROM, as shown in **Figure 2-1**.

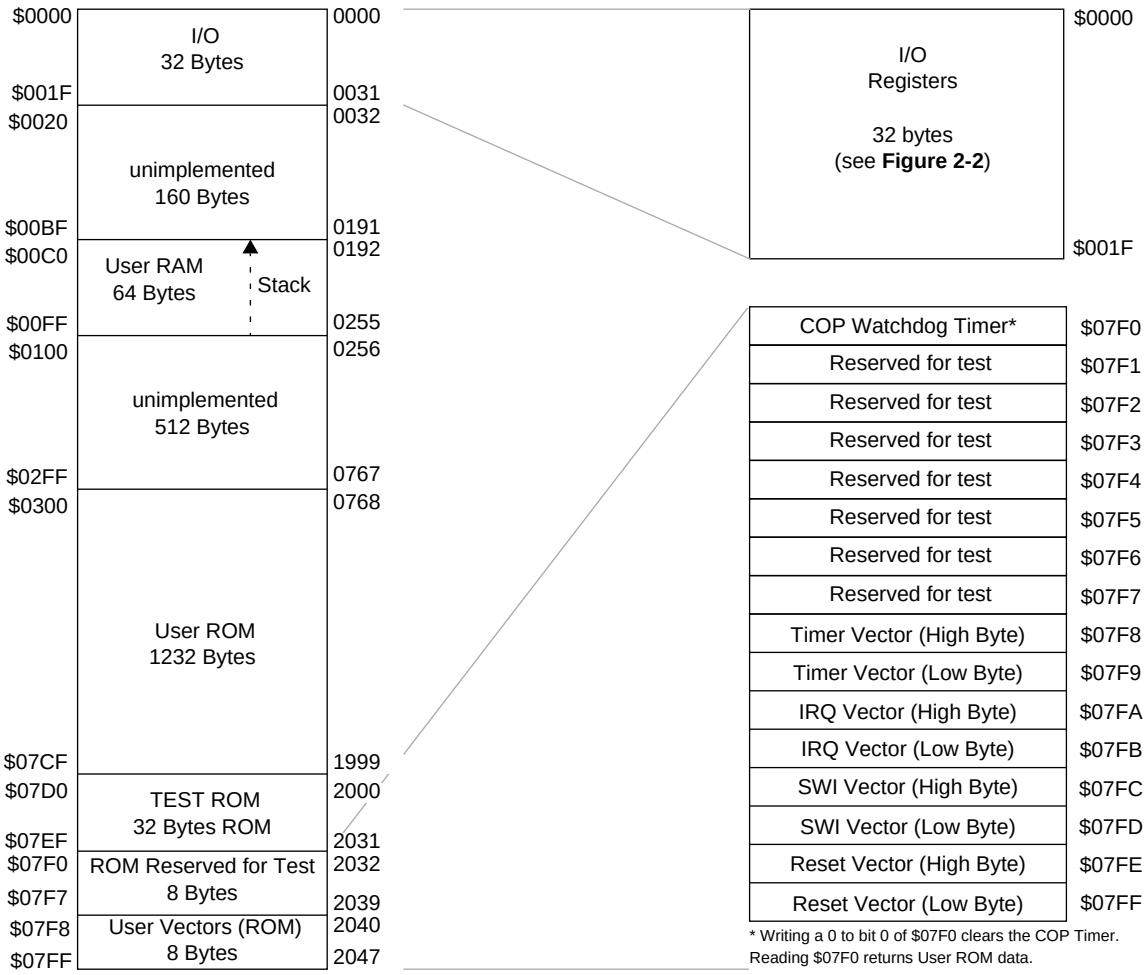


Figure 2-1. MC68HC05J5 Memory Map

2.2 I/O AND CONTROL REGISTERS

The I/O and Control Registers reside in locations $0000-$001F. The overall organization of these registers is shown in **Figure 2-2**. The bit assignments for each register are shown in **Figure 2-3** and **Figure 2-4**. Reading from unimplemented bits will return unknown states, and writing to unimplemented bits will be ignored.

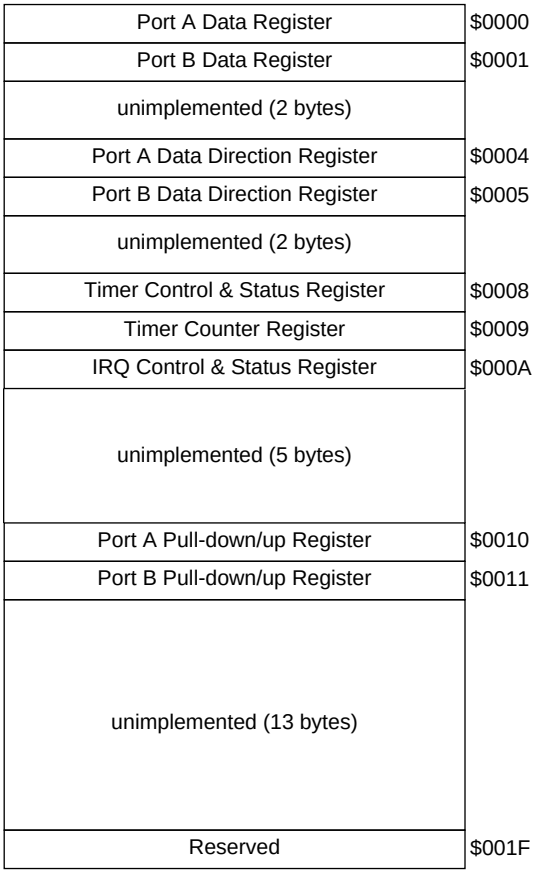


Figure 2-2. I/O Registers Memory Map

2.3 RAM

The User RAM consists of 64 bytes (including the stack), located from $00C0 to $00FF. The stack begins at address $00FF and proceeds down to $00C0. Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

2.4 ROM

There are a total of 1240 bytes of user ROM on-chip. This includes 1232 bytes of user ROM from locations $0300 to $07CF for user program storage and 8 bytes for user vectors from locations $07F8 to $07FF. There are a total of 40 bytes of Internal Test ROM on chip at locations $07D0 to $07EF and from $07F0 to $07F7.

ADDR	REGISTER	READ / WRITE	7	6	5	4	3	2	1	0
$0000	PORT A DATA PORTA	R	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
		W								
$0001	PORT B DATA PORTB	R	0	0	PB5	PB4	PB3	PB2	PB1	PB0
		W								
$0002	unimplemented	R								
		W								
$0003	unimplemented	R								
		W								
$0004	PORT A DATA DIRECTION DDRA	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		W								
$0005	PORT B DATA DIRECTION DDRB	R	SLOWE	0	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		W								
$0006	unimplemented	R								
		W								
$0007	unimplemented	R								
		W								
$0008	TIMER CONTROL & STATUS TCSR	R	TOF	RTIF	TOFE	RTIE	0	0	RT1	RT0
		W					TOFR	RTIFR		
$0009	TIMER COUNTER TCR	R	TMR7	TMR6	TMR5	TMR4	TMR3	TMR2	TMR1	TMR0
		W								
$000A	IRQ CONTROL & STATUS ICSR	R	IRQE	IRQE1	0	0	IRQF	IRQF1	0	0
		W							IRQR	IRQR1
$000B	unimplemented	R								
		W								
$000C	unimplemented	R								
		W								
$000D	unimplemented	R								
		W								
$000E	unimplemented	R								
		W								
$000F	unimplemented	R								
		W								

UNIMPLEMENTED RESERVED FOR TEST

Figure 2-3. I/O Registers $0000-$000F

ADDR	REGISTER	READ WRITE	7	6	5	4	3	2	1	0
$0010	PORT A PULLDOWN/UP REG. PDURA	R								
		W	PURA7	PURA6	PDRA5	PDRA4	PDRA3	PDRA2	PDRA1	PDRA0
$0011	PORT B PULLDOWN/UP REG. PDURB	R								
		W			PDRB5	PDRB4	PDRB3	PURB2	PURB1	PDRB0
$0012	unimplemented	R								
		W								
$0013	unimplemented	R								
		W								
$0014	unimplemented	R								
		W								
$0015	unimplemented	R								
		W								
$0016	unimplemented	R								
		W								
$0017	unimplemented	R								
		W								
$0018	unimplemented	R								
		W								
$0019	unimplemented	R								
		W								
$001A	unimplemented	R								
		W								
$001B	unimplemented	R								
		W								
$001C	unimplemented	R								
		W								
$001D	unimplemented	R								
		W								
$001E	unimplemented	R								
		W								
$001F	RESERVED FOR TEST TEST	R								
		W								

UNIMPLEMENTED RESERVED FOR TEST

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

SECTION 3
CENTRAL PROCESSING UNIT

The MC68HC05J5 has a 2K memory map. The stack has only 64 bytes. Therefore, the stack pointer has been reduced to only 6 bits and will only decrement down to $00C0 and then wrap-around to $00FF. All other instructions and registers behave as described in this chapter.

3.1 REGISTERS

The MCU contains five registers which are hard-wired within the CPU and are not part of the memory map. These five registers are shown in **Figure 3-1** and are described in the following paragraphs.

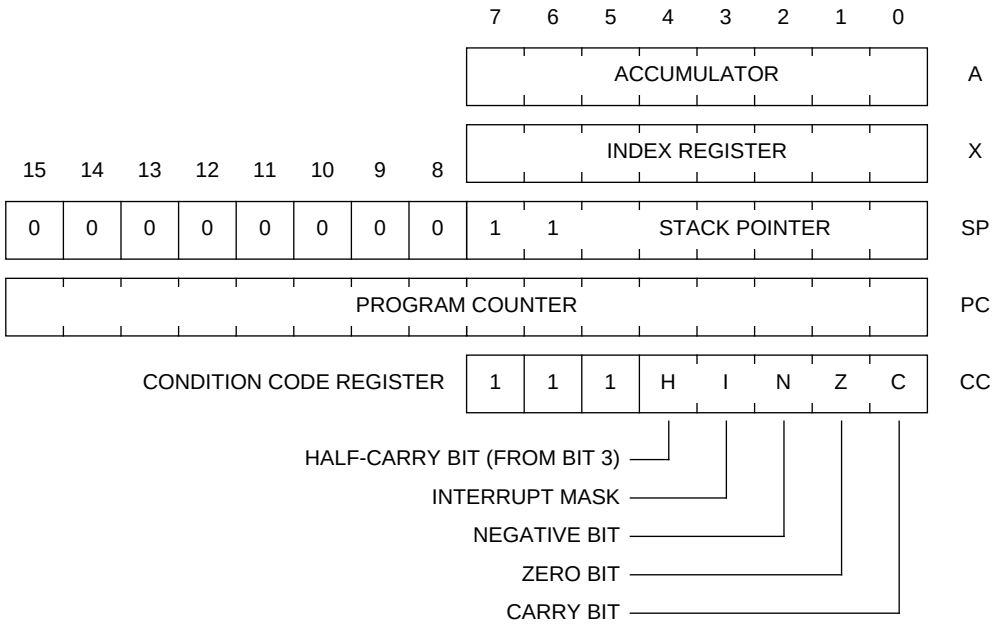


Figure 3-1. MC68HC05 Programming Model

3.1.1 Accumulator (A)

The accumulator is a general purpose 8-bit register as shown in **Figure 3-1**. The CPU uses the accumulator to hold operands and results of arithmetic calculations or non-arithmetic operations. The accumulator is not affected by a reset of the device.

3.1.2 Index Register (X)

The index register shown in **Figure 3-1** is an 8-bit register that 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 content 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 content to a 16-bit immediate value.

The index register can also serve as an auxiliary accumulator for temporary storage. The index register is not affected by a reset of the device.

3.1.3 Stack Pointer (SP)

The stack pointer shown in **Figure 3-1** is a 16-bit register. In MCU devices with memory space less than 64K-bytes the unimplemented upper address lines are ignored. The stack pointer contains the address of the next free location on the stack. During a reset or the reset stack pointer (RSP) instruction, the stack pointer is set to $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled off the stack.

When accessing memory, the ten most significant bits are permanently set to 0000000011. The six least significant register bits are appended to these ten fixed bits to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 ($40) locations. If 64 locations are exceeded, the stack pointer wraps around and overwrites the previously stored information. A subroutine call occupies two locations on the stack and an interrupt uses five locations.

3.1.4 Program Counter (PC)

The program counter shown in **Figure 3-1** is a 16-bit register. In MCU devices with memory space less than 64K-bytes the unimplemented upper address lines are ignored. The program counter contains the address of the next instruction or operand to be fetched.

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 CCR shown in **Figure 3-1** is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. The fifth bit is the interrupt mask. These bits can be individually tested by a program, and specific actions can be taken as a result of their states. The condition code register should be thought of as having three additional upper bits that are always ones. Only the interrupt mask is affected by a reset of the device. 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 (add with carry) 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, the internal 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 request occurs while the interrupt mask is set, the interrupt request 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 state before the interrupt was encountered. After any reset, the interrupt mask is set and can only be cleared by the Clear I-Bit (CLI), or WAIT instructions.

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, data manipulation, or data load operation 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. This bit is neither set by an INC nor by a DEC instruction.

SECTION 4
INTERRUPTS

The CPU can be interrupted in five different ways:

- Non-maskable Software Interrupt Instruction (SWI)
- External Asynchronous Interrupt ($\overline{IRQ}$)
- Optional External Interrupt on PA0-PA3 (mask option)
- External Interrupt on PA7
- Internal Timer Interrupt

4.1 CPU INTERRUPT PROCESSING

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

If interrupts are not masked (I-bit in the CCR is clear) and the corresponding interrupt enable bit is set the processor will proceed with interrupt processing. Otherwise, the next instruction is fetched and executed. If an interrupt occurs the processor completes the current instruction, then stacks the current CPU register states, sets the I-bit to inhibit further interrupts, and finally checks the pending hardware interrupts. If more than one interrupt is pending following the stacking operation, the interrupt with the highest vector location shown in **Table 4-1** will be serviced first. The SWI is executed the same as any other instruction, regardless of the I-bit state.

When an interrupt is to be processed the CPU fetches the address of the appropriate interrupt software service routine from the vector table at locations $07F8 to $07FF as defined in **Table 4-1**.

Table 4-1. Vector Address for Interrupts and Reset

Register	Flag Name	Interrupts	CPU Interrupt	Vector Address
N/A	N/A	Reset	$\overline{RESET}$	$07FE-$07FF
N/A	N/A	Software	SWI	$07FC-$07FD
ICSR	IRQF/IRQF1	External Interrupt	$\overline{IRQ}$	$07FA-$07FB
TCSR	TOF	Timer Overflow	TIMER	$07F8-$07F9
TCSR	RTIF	Real Time Interrupt	TIMER	$07F8-$07F9

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

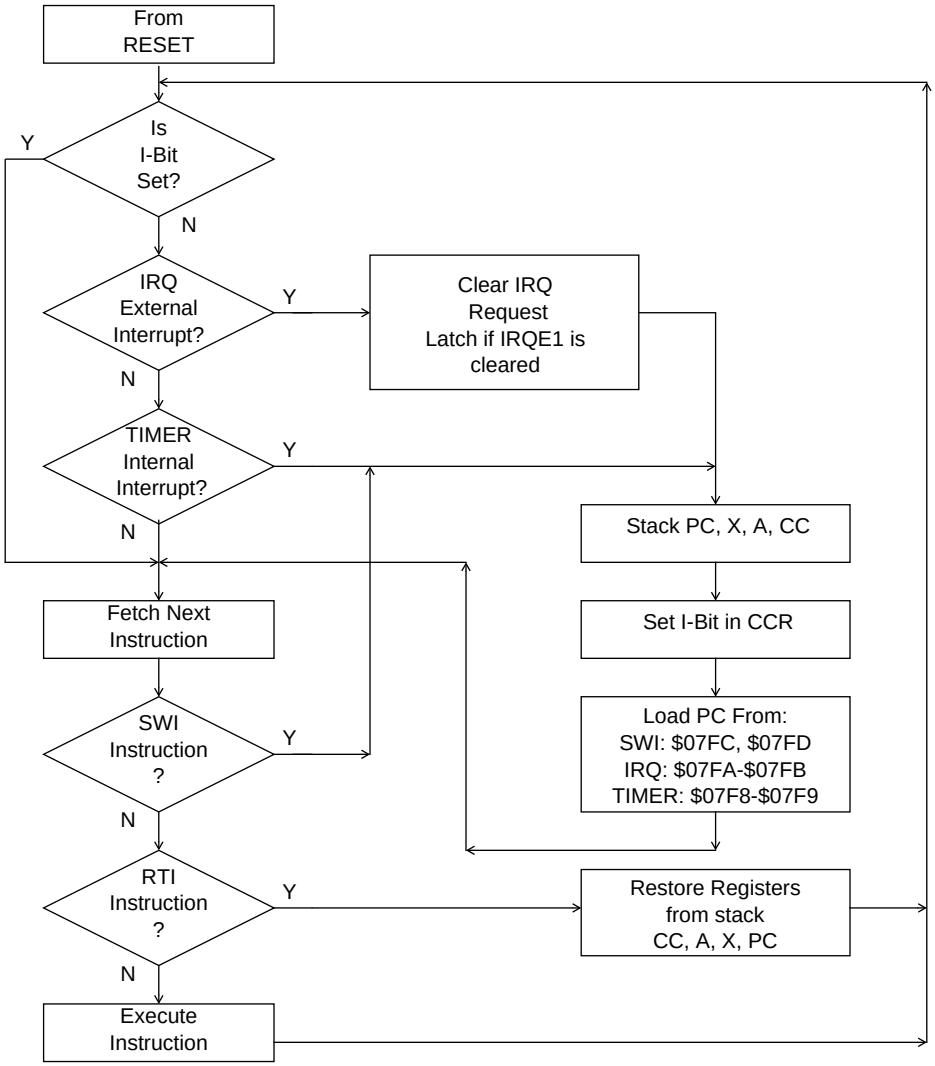


Figure 4-1. Interrupt Processing Flowchart

4.2 RESET INTERRUPT SEQUENCE

The RESET function is not in the strictest sense an interrupt; however, it is acted upon in a similar manner as shown in **Figure 4-1**. A low level input on the $\overline{\text{RESET}}$ pin or an internally generated RST signal causes the program to vector to its starting address which is specified by the contents of memory locations $07FE and $07FF. The I-bit in the condition code register is also set.

4.3 SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction and a non-maskable interrupt since it is executed regardless of the state of the I-bit in the CCR. As with any instruction, interrupts pending during the previous instruction will be serviced before the SWI opcode is fetched. The interrupt service routine address is specified by the contents of memory locations $07FC and $07FD.

4.4 HARDWARE INTERRUPTS

All hardware interrupts except RESET are maskable by the I-bit in the CCR. If the I-bit is set, all hardware interrupts (internal and external) are disabled. Clearing the I-bit enables the hardware interrupts. There are two types of hardware interrupts which are explained in the following sections.

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

Interrupts from external pins are available on:

- $\overline{\text{IRQ}}$ pin
- PA0 to PA3 pins (enabled by mask option)
- PA7 pin

4.4.1.1 $\overline{\text{IRQ}}$, PA0, PA1, PA2, and PA3 Pins

If “edge-only” sensitivity is chosen by mask option, the IRQ interrupt is sensitive to the following cases:

1. Falling edge on the $\overline{\text{IRQ}}$ pin.
2. Rising edge on any PA0-PA3 pin with IRQ enabled (via mask option).

If “edge-and-level” sensitivity is chosen, the IRQ interrupt is sensitive to the following cases:

1. Low level on the $\overline{\text{IRQ}}$ pin.
2. Falling edge on the $\overline{\text{IRQ}}$ pin.
3. High level on any PA0-PA3 pin with IRQ enabled (via mask option).
4. Rising edge on any PA0-PA3 pin with IRQ enabled (via mask option).

The IRQE enable bit controls whether an active IRQF flag can generate an IRQ interrupt sequence. This interrupt is serviced by the interrupt service routine located at the address specified by the contents of $07FA and $07FB.

The IRQ latch is automatically cleared by entering the interrupt service routine if IRQE1 enable bit is cleared. If IRQE1 enable bit is also set, the only way of clearing IRQF is by writing a logic one to the IRQR acknowledge bit. Writing a logic one to the IRQR acknowledge bit in the ICSR is the other way of clearing IRQF flag, regardless of the status of the IRQE1 bit, besides IRQ vector fetch. This conditional reset of IRQF flag provides a way for the user to differentiate the

interrupt sources from IRQ and IRQ1 latches and also to make it HC05J1A compatible if PA7 interrupt is not used. As long as the output state of the IRQF flag bit is active the CPU will continuously re-enter the IRQ interrupt sequence until the active state is removed or the IRQE enable bit is cleared.

4.4.1.2 PA7 Pin

PA7 interrupt source, if enabled by IRQE1 enable bit, triggers IRQ interrupt on PA7 falling edge only. The IRQ1 latch (IRQF1 flag) can ONLY be cleared by writing a logic one to the IRQR1 acknowledge bit in the ICSR. IRQ vector fetch can NOT clear IRQF1 flag. IRQ interrupt caused by PA7 falling edge also vectors to $07FA and $07FB.

4.4.2 IRQ Control/Status Register (ICSR), $0A

The IRQ interrupt function is controlled by the ICSR located at $000A. All unused bits in the ICSR will read as logic zeros. The IRQF, IRQF1, IRQE1 bits are cleared and IRQE bit is set by reset.

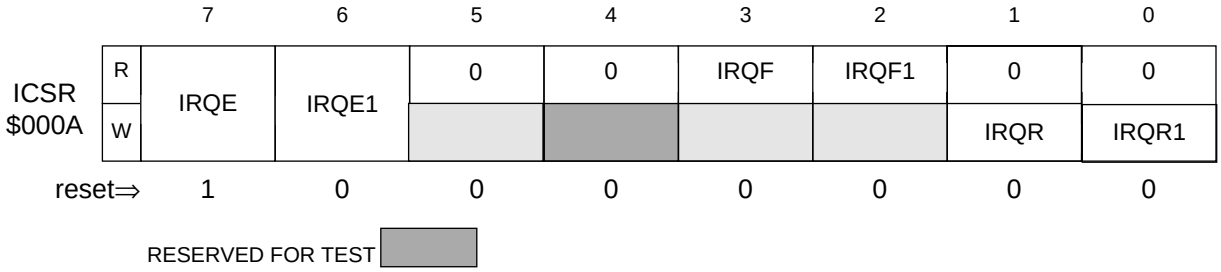


Figure 4-2. IRQ Status & Control Register

IRQR 1 - PA7 Interrupt Acknowledge

The IRQR1 acknowledge bit clears an IRQ interrupt triggered by a falling edge on PA7 by clearing the IRQ1 latch. The IRQR1 acknowledge bit will always read as a logic zero.

1 = Writing a logic one to the IRQR1 acknowledge bit will clear the IRQ1 latch.

0 = Writing a logic zero to the IRQR1 acknowledge bit will have no effect on the IRQ1 latch.

IRQR - IRQ Interrupt Acknowledge

The IRQR acknowledge bit clears an IRQ interrupt by clearing the IRQ latch. The IRQR acknowledge bit will always read as a logic zero.

1 = Writing a logic one to the IRQR acknowledge bit will clear the IRQ latch.

0 = Writing a logic zero to the IRQR acknowledge bit will have no effect on the IRQ latch.

IRQF1 - PA7 Interrupt Request Flag

Writing to the IRQF1 flag bit will have no effect on it. If the additional setting of IRQF1 flag bit is not cleared in the IRQ service routine and the IRQE1 enable bit remains set the CPU will re-enter the IRQ interrupt sequence continuously until either the IRQF1 flag bit or the IRQE1 enable bit is cleared. The IRQF1 latch is cleared by reset.

1 = Indicates that an IRQ request triggered by a falling edge on PA7 is pending.

0 = Indicates that no IRQ request triggered by a falling edge on PA7 is pending. The IRQF1 flag bit can ONLY be cleared by writing a logic one to the IRQR1 acknowledge bit. Doing so before exiting the service routine will mask out additional occurrences of the IRQF1.

IRQF - IRQ Interrupt Request Flag

Writing to the IRQF flag bit will have no effect on it. If the additional setting of IRQF flag bit is not cleared in the IRQ service routine and the IRQE enable bit remains set the CPU will re-enter the IRQ interrupt sequence continuously until either the IRQF flag bit or the IRQE enable bit is clear. The IRQF latch is cleared by reset.

1 = Indicates that an IRQ request is pending.

0 = Indicates that no IRQ request triggered by pins PA0-3 or $\overline{\text{IRQ}}$ is pending. The IRQF flag bit is cleared once the IRQ vector is fetched AND if IRQE1 is also cleared. If IRQE1 is set, then the only way of clearing IRQF flag is by writing a logic one to IRQR bit. The IRQF flag bit can be cleared, regardless of the status of the IRQE1 bit, by writing a logic one to the IRQR acknowledge bit to clear the IRQ latch and also conditioning the external IRQ sources to be inactive (if the level sensitive interrupts are enabled via mask option). Doing so before exiting the service routine will mask out additional occurrences of the IRQF.

IRQE1 - PA7 Interrupt Enable

The IRQE1 bit enables/disables the IRQF1 flag bit to initiate an IRQ interrupt sequence.

1 = Enables IRQF1 interrupt, that is, the IRQF1 flag bit can generate an interrupt sequence. Execution of the STOP or WAIT instructions will leave the IRQE1 bit to be UNAFFECTED.

0 = The IRQF1 flag bit cannot generate an interrupt sequence. Reset clears the IRQE1 enable bit, thereby disabling PA7 interrupts.

IRQE - IRQ Interrupt Enable

The IRQE bit enables/disables the IRQF flag bit to initiate an IRQ interrupt sequence.

1 = Enables IRQF interrupt, that is, the IRQF flag bit can generate an interrupt sequence. Reset sets the IRQE enable bit, thereby enabling IRQ interrupts once the I-bit is cleared. Execution of the STOP or WAIT instructions causes the IRQE bit to be set in order to allow the external IRQ to exit these modes.

0 = The IRQF flag bit cannot generate an interrupt sequence.

4.4.3 Optional External Interrupts (PA0-PA3)

The IRQ interrupt can also be triggered by the inputs on the PA0 to PA3 port pins if enabled by a single mask option. If enabled, the lower four bits of Port A can activate the IRQ interrupt function, and the interrupt operation will be the same as for inputs to the $\overline{IRQ}$ pin. This mask option of PA0-3 interrupt allow all of these input pins to be OR'ed with the input present on the $\overline{IRQ}$ pin. All PA0 to PA3 pins must be selected as a group as an additional IRQ interrupt. All the PA0-3 interrupt sources are also controlled by the IRQE enable bit.

NOTE

The BIH and BIL instructions will only apply to the level on the $\overline{IRQ}$ pin itself, and not to the output of the logic OR function with the PA0 to PA3 pins. The state of the individual Port A pins can be checked by reading the appropriate Port A pins as inputs.

NOTE

If enabled, the PA0 to PA3 and PA7 pins will cause an IRQ interrupt regardless of whether these pins are configured as inputs or outputs.

4.4.4 Timer Interrupt (TIMER)

The TIMER interrupt is generated by the multi-function timer when either a timer overflow or a real time interrupt has occurred as described in **Section 8**. The interrupt flags and enable bits for the Timer interrupts are located in the Timer Control/Status Register (TCSR) located at $0008. The I-bit in the CCR must be clear in order for the TIMER interrupt to be enabled. Either of these two interrupts will vector to the same interrupt service routine located at the address specified by the contents of memory locations $07F8 and $07F9.

SECTION 5
RESETS

The MCU can be reset from five sources: one external input and four internal restart conditions.

5.1 EXTERNAL RESET ($\overline{\text{RESET}}$)

The $\overline{\text{RESET}}$ pin is the only external source of a reset. This pin is connected to a Schmitt trigger input gate to provide an upper and lower threshold voltage separated by a minimum amount of hysteresis. This external reset occurs whenever the $\overline{\text{RESET}}$ pin is pulled below the lower threshold and remains in reset until the $\overline{\text{RESET}}$ pin rises above the upper threshold. This active low input will generate the RST signal and reset the CPU and peripherals. This pin is also an output pin whenever the LVR triggers an internal reset. Termination of the external $\overline{\text{RESET}}$ input or the internal COP Watchdog reset or LVR are the only reset sources that can alter the operating mode of the MCU.

5.2 INTERNAL RESETS

The four internally generated resets are the initial power-on reset function, the COP Watchdog Timer reset, the illegal address detector reset and the low voltage reset (LVR). Termination of the external $\overline{\text{RESET}}$ input or the internal COP Watchdog Timer or LVR are the only reset sources that can alter the operating mode of the MCU. The other internal resets will not have any effect on the mode of operation when their reset state ends.

5.2.1 Power-On Reset (POR)

The internal POR is generated on power-up to allow the clock oscillator to stabilize. The POR is strictly for power turn-on conditions and is not able to detect a drop in the power supply voltage (brown-out). There is an oscillator stabilization delay of 4064 internal processor bus clock cycles (PH2) after the oscillator becomes active.

The POR will generate the RST signal which will reset the CPU. If any other reset function is active at the end of this 4064 cycle delay, the RST signal will remain in the reset condition until the other reset condition(s) end.

5.2.2 Computer Operating Properly Reset (COPR)

The internal COPR reset is generated automatically (if the COP is enabled) by a time-out of the COP Watchdog Timer. This time-out occurs if the counter in the COP Watchdog Timer is not reset (cleared) within a specific time by a software reset sequence. The COP Watchdog Timer can be disabled by a mask option. Refer to **Section 8.2** for more information on this time-out feature. COP reset also forces the $\overline{\text{RESET}}$ pin low

The COPR will generate the RST signal which will reset the CPU and other peripherals. Also, the COPR will establish the mode of operation based on the state of the $\overline{\text{IRQ}}$ pin at the time the COPR signal ends. If the voltage on the $\overline{\text{IRQ}}$ pin is at the V_{TST} level, the state of the PB0 pin during the last rising edge of the $\overline{\text{RESET}}$ pin will determine which Test Mode (Internal or Expanded) the MCU will be in. If the voltage at the $\overline{\text{IRQ}}$ pin is in the normal operating range (V_{SS} to V_{DD}), the MCU will enter Single-Chip Mode when the COPR signal ends. If any other reset function is active at the end of the COPR reset signal, the RST signal will remain in the reset condition until the other reset condition(s) end.

5.2.3 Illegal Address Reset (ILADR)

The internal ILADR reset is generated when an instruction opcode fetch occurs from an address which is not implemented in the RAM ($00C0 - $00FF) nor ROM ($0300-$07FF). The ILADR will generate the RST signal which will reset the CPU and other peripherals. If any other reset function is active at the end of the ILADR reset signal, the RST signal will remain in the reset condition until the other reset condition(s) end. Notice that ILADR also forces the $\overline{\text{RESET}}$ pin low

5.2.4 Low Voltage Reset (LVR)

The internal LVR reset is generated when V_{DD} falls below the specified LVR trigger value V_{LVR} for at least one t_{CYC}. In typical applications, the power supply decoupling circuit will eliminate negative-going voltage glitches of less than one t_{CYC}. This reset will hold the MCU in the reset state until V_{DD} rises above V_{LVR}. Whenever V_{DD} is above V_{LVR} and below 4.5V, the MCU is guaranteed to operate although not within specification. The output from the LVR is connected directly to the internal reset circuitry and also forces the $\overline{\text{RESET}}$ pin low. The internal reset will be removed once the power supply voltage rises above V_{LVR}, at which time a normal power-on-reset sequence occurs.

SECTION 6
MODES OF OPERATION

The MC68HC05J5 has the following operating modes: Single-Chip Mode (SCM) and self-check mode.

The Single-Chip Mode allows maximum use of pins for on-chip peripheral functions.

The self-check mode capability of the MC68HC05J5 provides an internal check to determine if the device is functional.

This section also provides a description of the low-power modes.

6.1 MODE ENTRY

The mode entry is done at the rising edge of the $\overline{\text{RESET}}$ pin. Once the device enters one of the operating modes, the mode can be changed only by external reset not software.

At the rising edge of the $\overline{\text{RESET}}$ pin, the device latches the states of the $\overline{\text{IRQ}}$ and PB0 pins and places itself in the specified mode. While the RESET pin low, all pins are configured as Single-Chip Mode. **Table 6-1** shows the states of $\overline{\text{IRQ}}$ and PB0 pins for each mode.

Table 6-1. Mode Select Summary

Mode	$\overline{\text{RESET}}$	$\overline{\text{IRQ}}$	PB0
Single-Chip Mode		L or H	X
Self-Check Mode		V_{TST}	H

$V_{TST}=1.8 \times V_{DD}$
H = V_{DD}
L = GND

6.2 SINGLE-CHIP MODE (SCM)

The Single-Chip Mode allows the MCU to function as a self-contained microcontroller, with maximum use of the pins for on-chip peripheral functions.

In the Single-Chip Mode all address and data activity occurs within the MCU and is not available externally. Single-Chip Mode is entered if the $\overline{\text{IRQ}}$ pin is within the normal operating voltage range when the rising edge of a $\overline{\text{RESET}}$ or a COP Watchdog reset or an internal LVR reset occurs. In Single-Chip Mode, all I/O port pins are available.

6.3 SELF-CHECK MODE

The self-check mode provides an internal check to determine if the device is functional.

6.4 LOW-POWER MODES

In each of its configuration modes the MC68HC05J5 is capable of running in one of several low-power operational modes. The WAIT and STOP instructions provide two modes that reduce the power required for the MCU by stopping various internal clocks and/or the on-chip oscillator. The STOP and WAIT instructions are not normally used if the COP Watchdog Timer is enabled. A mask option is provided to convert the STOP instruction to a HALT, which is a WAIT-like instruction that does not halt the COP Watchdog Timer but has a recovery delay. The flow of the STOP, HALT, and WAIT modes are shown in **Figure 6-1**.

6.4.1 STOP Instruction

The STOP instruction can result in one of two modes of operation depending on the STOP mask option chosen. One option is for the STOP instruction to operate like the STOP in normal MC68HC05 family members and place the device in the STOP Mode. The other option is for the STOP instruction to behave like a WAIT instruction (except that the restart time will involve a delay) and place the device in the HALT Mode.

6.4.1.1 STOP Mode

Execution of the STOP instruction in this mode (as chosen by a mask option) places the MCU in its lowest power consumption mode. In the STOP Mode the internal oscillator is turned off, halting *all* internal processing, including the COP Watchdog Timer.

When the CPU enters STOP Mode the interrupt flags (TOF and RTIF) and the interrupt enable bits (TOFE and RTIE) in the TCSR are cleared by internal hardware to remove any pending timer interrupt requests and to disable any further timer interrupts. Execution of the STOP instruction automatically clears the I-bit in the Condition Code Register and sets the IRQE enable bit in the IRQ Control/Status Register so that the IRQ external interrupt is enabled. All other registers, including the other bits in the TCSR, and memory remain unaltered. All input/output lines remain unchanged.

The MCU can be brought out of the STOP Mode only by an IRQ external interrupt or an externally generated RESET or an LVR reset. When exiting the STOP Mode the internal oscillator will resume after a 4064 internal processor clock cycle oscillator stabilization delay.

NOTE

Execution of the STOP instruction with the STOP Mode Mask Option will cause the oscillator to stop and therefore disable the COP Watchdog Timer. If the COP Watchdog Timer is to be used, the STOP Mode should be changed to the HALT Mode by choosing the appropriate mask option. See **Section 6.6** for more details.

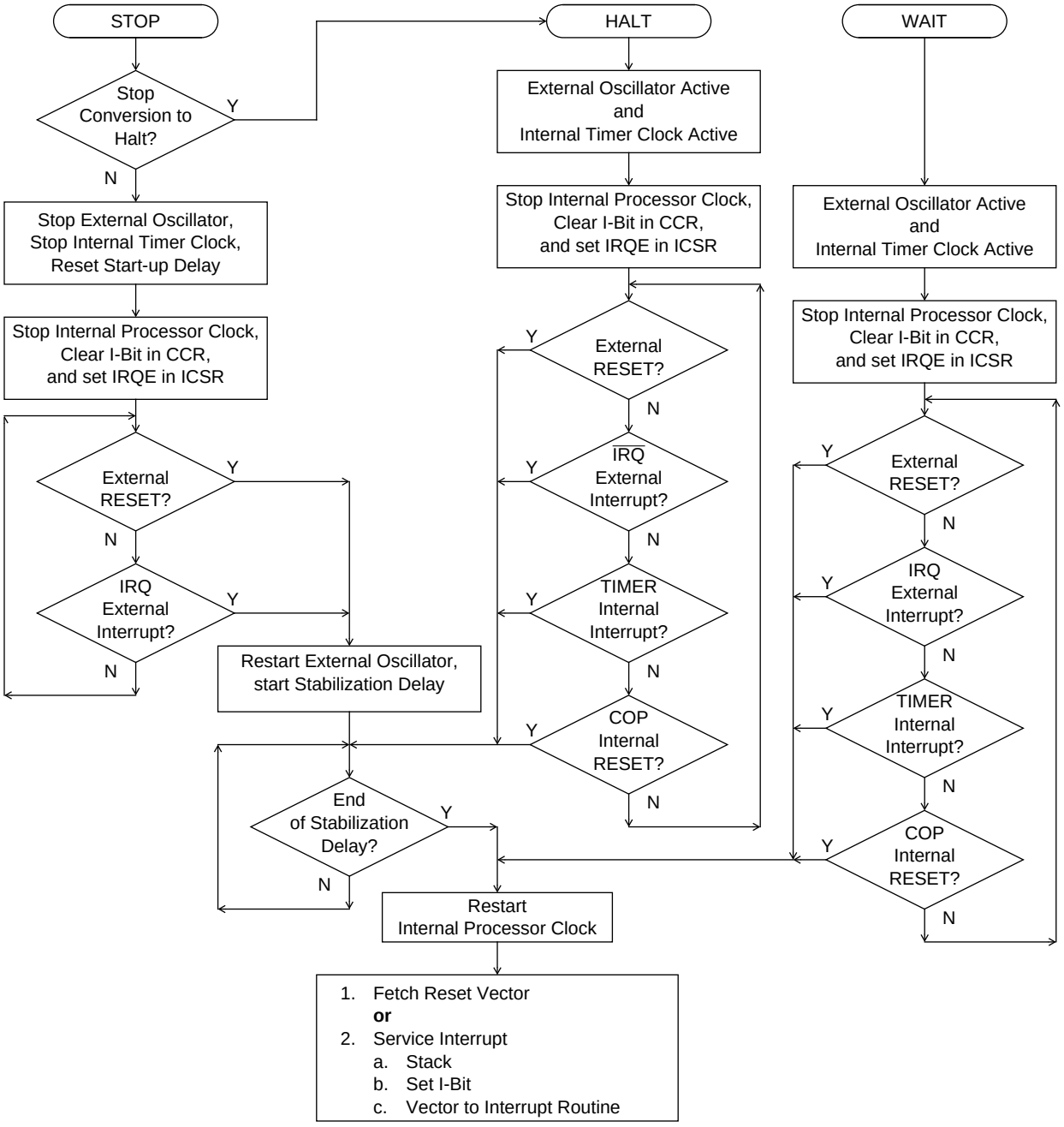


Figure 6-1. STOP/HALT/WAIT Flowcharts

6.4.1.2 HALT Mode

Execution of the STOP instruction in this mode (as chosen by a mask option) places the MCU in a low-power mode, which consumes more power than the STOP Mode. In the HALT Mode the internal processor clock is halted, suspending all processor and internal bus activity. Internal timer clocks remain active, permitting interrupts to be generated from the timer or a reset to be generated from the COP Watchdog Timer. Execution of the STOP instruction automatically clears the I-bit in the Condition Code Register and sets the IRQE enable bit in the IRQ Control/Status Register so that the IRQ external interrupt is enabled. All other registers, memory, and input/output lines remain in their previous states.

The HALT Mode may be exited when a Timer interrupt, an external IRQ, an LVR reset, or external $\overline{\text{RESET}}$ occurs. When exiting the HALT Mode the internal processor clock will resume after a delay of one to 4064 internal processor clock cycles. This varied delay time is due to the HALT Mode testing the oscillator stabilization delay timer (a feature of the STOP Mode) which has been free-running (a feature of the WAIT Mode).

NOTE

The HALT Mode is not intended for normal use, but is provided to keep the COP Watchdog Timer active should the STOP instruction opcode be inadvertently executed.

6.4.2 WAIT Mode

The WAIT instruction places the MCU in a low-power mode, which consumes more power than the STOP Mode. In the WAIT Mode the internal processor clock is halted, suspending all processor and internal bus activity. Internal timer clocks remain active, permitting interrupts to be generated from the timer or a reset to be generated from the COP Watchdog Timer. Execution of the WAIT instruction automatically clears the I-bit in the Condition Code Register and sets the IRQE enable bit in the IRQ Control/Status Register so that the IRQ external interrupt is enabled. All other registers, memory, and input/output lines remain in their previous states.

If timer interrupts are enabled, a TIMER interrupt will cause the processor to exit the WAIT Mode and resume normal operation. The Timer may be used to generate a periodic exit from the WAIT Mode. The WAIT Mode may also be exited when an external IRQ or an LVR reset or an external RESET occurs.

6.5 DATA-RETENTION MODE

If the LVR mask option is selected and since LVR kicks in whenever V_{DD} is below the specified LVR trigger voltage which is higher than that required of the Data Retention mode, the Data Retention mode will not exist. Data Retention Mode is only meaningful if LVR mask option is not selected.

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. The $\overline{\text{RESET}}$ pin must be held low during data-retention mode.

6.6 COP WATCHDOG TIMER CONSIDERATIONS

The COP Watchdog Timer is active in all modes of operation if enabled by a mask option. However, regardless of the mask option chosen, the COP Watchdog Timer will be **disabled** if the voltage on the $\overline{\text{IRQ}}$ pin equals or exceeds the V_{TST} voltage level. Thus, emulation of applications that do not service the COP should only be done with devices that have the COP Mask Option disabled. This prevents the voltage level on the $\overline{\text{IRQ}}$ pin from enabling the COP which would cause a reset and possibly change the operating mode of the device.

If the COP Watchdog Timer is selected by the mask option, any execution of the STOP instruction (either intentional or inadvertent due to the CPU being disturbed) will cause the oscillator to halt and prevent the COP Watchdog Timer from timing out unless the STOP to HALT conversion feature is enabled. Therefore, it is recommended that the STOP instruction should be **converted** to a HALT instruction if the COP Watchdog Timer is enabled.

If the COP Watchdog Timer is selected by the mask option, the COP will reset the MCU when it times out. Therefore, it is recommended that the COP Watchdog should be **disabled** for a system that must have intentional uses of the WAIT Mode for periods longer than the COP time-out period.

The recommended interactions and considerations for the COP Watchdog Timer, STOP instruction, and WAIT instruction are summarized in **Table 6-2**.

Table 6-2. COP Watchdog Timer Recommendations

IF the following conditions exist:			THEN the COP Watchdog Timer should be as follows:
Voltage on $\overline{\text{IRQ}}$ Pin	STOP Instruction	WAIT Time	
less than V_{TST}	converted to HALT by mask option	WAIT Time **less than** COP Time-Out	Enable or disable COP by mask option
less than V_{TST}	converted to HALT by mask option	WAIT Time **more than** COP Time-Out	Disable COP by mask option
less than V_{TST}	Acts as STOP	any length WAIT Time	Disable COP by mask option
more than V_{TST}	Acts as STOP or converted to HALT by mask option	any length WAIT Time	COP is disabled by IRQ input level

$V_{TST} = 1.8 \times V_{DD}$

SECTION 7
INPUT/OUTPUT PORTS

In the Single-Chip Mode there are 14 usable bidirectional I/O lines arranged as one 8-bit I/O port (Port A), and one 6-bit I/O port (Port B). The individual bits in these ports are programmable as either inputs or outputs under software control by the data direction registers (DDR's). Also, if enabled by a single mask option all Port A and Port B I/O pins may have individual software programmable pull-down or pull-up devices. Also, PA4-PA7 and PB1-PB2 pins have properties of sinking higher current; PA0-PA3 may function as additional $\overline{IRQ}$ interrupt input sources. Note that both PA6 and PA7 pins have Schmitt trigger input for better noise immunity. V_{IH} and V_{IL} specified at 2.4V and 0.8V, respectively.

7.1 SLOW OUTPUT FALLING-EDGE TRANSITION

		7	6	5	4	3	2	1	0
DDRB $0005	R	SLOWE	0	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
	W								
reset⇒		0	0	0	0	0	0	0	0

Figure 7-1. Port B Data Direction Register

SLOWE - Slow Transition Enabled

The slow transition feature is controlled by the SLOWE bit of DDRB (Port B Data Direction Register).

1 = Enables the slow falling-edge output transition feature on the four I/O lines: PA6, PA7, PB1, and PB2. If the pin is configured as an output pin.

0 = Disables slow falling-edge output transition feature on the four I/O lines: PA6, PA7, PB1, and PB2. Default value of SLOWE bit is cleared.

7.2 PORT A

Port A is an 8-bit bi-directional port which shares five of its pins with the IRQ interrupt system as shown in **Figure 7-2**. Note that both PA6 and PA7 pins have Schmitt trigger input for better noise immunity. Only PA6 and PA7 are of open-drained type with slow output transition feature. Each Port A pin is controlled by

the corresponding bits in a data direction register, a data register, and a pull-down/up register. The Port A Data Register is located at address $0000. The Port A Data Direction Register (DDRA) is located at address $0004. The Port A Pull-down/up Register (PDURA) is located at address $0010. Reset clears the DDRA and the PDURA. The Port A Data Register is unaffected by reset.

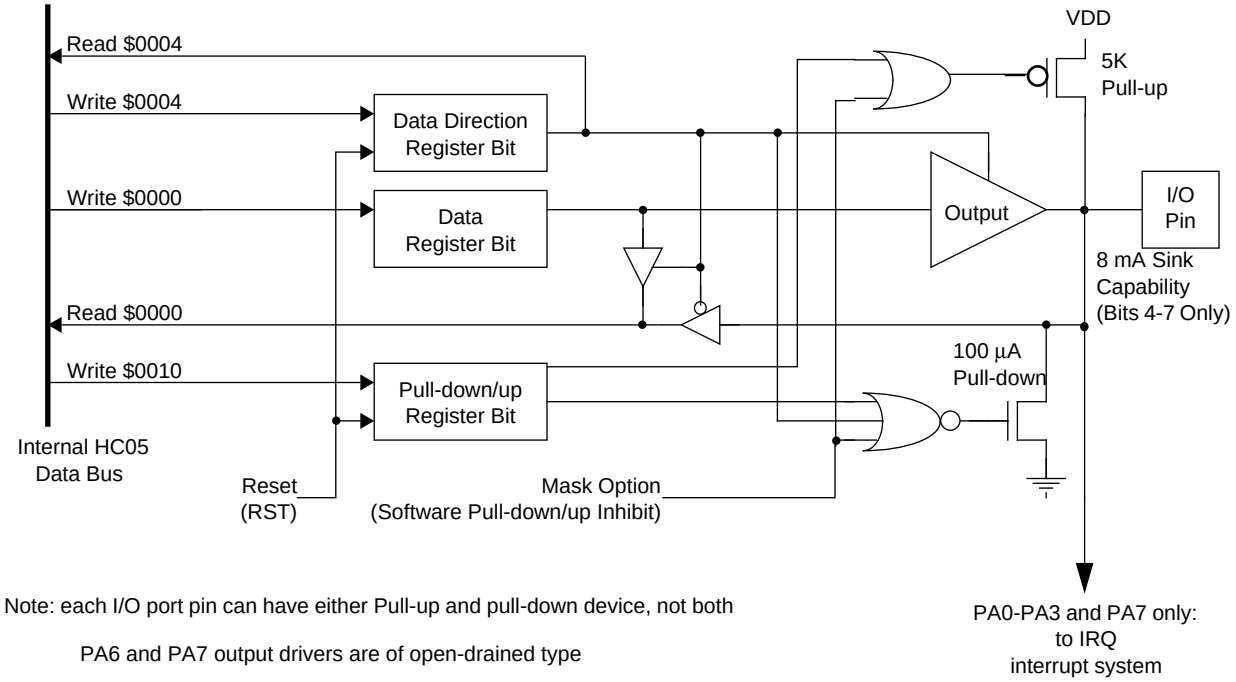


Figure 7-2. Port A I/O Circuitry

7.2.1 Port A Data Register

Each Port A I/O pin has a corresponding bit in the Port A Data Register. When a Port A pin is programmed as an output the state of the corresponding data register bit determines the state of the output pin. When a Port A pin is programmed as an input, any read of the Port A Data Register will return the logic state of the corresponding I/O pin. The Port A data register is unaffected by reset.

7.2.2 Port A Data Direction Register

Each Port A I/O pin may be programmed as an input by clearing the corresponding bit in the DDRA, or programmed as an output by setting the corresponding bit in the DDRA. The DDRA can be accessed at address $0004. The DDRA is cleared by reset.

If configured as output pins, PA6 and PA7 have slow output falling-edge transition feature. The slow transition feature is controlled by the SLOWE bit of DDRB. SLOWE bit, if set and if the pin is configured as an output pin, enables the slow falling-edge output transition feature of all three I/O lines, PA6, PA7, and PB1.

7.2.3 Port A Pull-down/up Register

All Port A I/O pins may have software programmable pull-down/up devices enabled by the applicable mask option. If the pull-down/up mask option is selected, the pull-down/up is activated whenever the corresponding bit in the PDURA is clear. If the corresponding bit in the PDURA bit is set or the mask option for pull-down/up is not chosen, the pull-down/up will be disabled. A pull-down on an I/O pin is activated only if the I/O pin is programmed as an input whereas a Pull-up device on an I/O pin is always activated whenever enabled, regardless of port direction.

The PDURA is a **write-only** register. Any reads of location $0010 will return undefined results. Since reset clears both the DDRA and the PDURA, all pins will initialize as inputs with the pull-down active and pull-up devices active (if enabled by mask option).

Typical value of port A pull-up is 5KΩ.

7.2.4 Port A Drive Capability

The outputs for the upper four bits of Port A (PA4, PA5, PA6 and PA7) are capable of sinking approximately 8 mA of current to V_{SS}.

7.2.5 Port A I/O Pin Interrupts

The inputs to PA0, PA1, PA2, PA3 may be connected to the IRQ input of the CPU if enabled by a mask option. The input to PA7 is also connected to the IRQ input of the CPU, yet it is only enabled or disabled by software, not by mask option. PA7 interrupt capability is controlled by a set of control and status bits (IRQE1, IRQF1, IRQR1), different from the set of control and status bits for that of PA0-PA3 and $\overline{IRQ}$ pin (IRQE, IRQF, IRQR) in the same ICSR (Interrupt Control and Status Register).

When connected as an alternate source of an IRQ interrupt, PA0-3 input pins will behave the same as the $\overline{IRQ}$ pin itself, except that their active state is a logical one or a rising edge. The $\overline{IRQ}$ pin has an active state that is a logical zero or a falling edge. PA7 interrupt occurs, if enabled, only upon the falling edge at the input.

If mask options for both level and edge sensitivity interrupts are chosen, the presence of a logic one or occurrence of a rising edge on any one of the lower four Port A pins will cause an IRQ interrupt request. If the edge-only sensitivity is selected, the occurrence of a rising edge on any one of the lower four Port A pins will cause an IRQ interrupt request. As long as any one of the lower four Port A IRQ inputs remains at a logic one level, the other of the lower four Port A IRQ inputs are effectively ignored.

NOTE

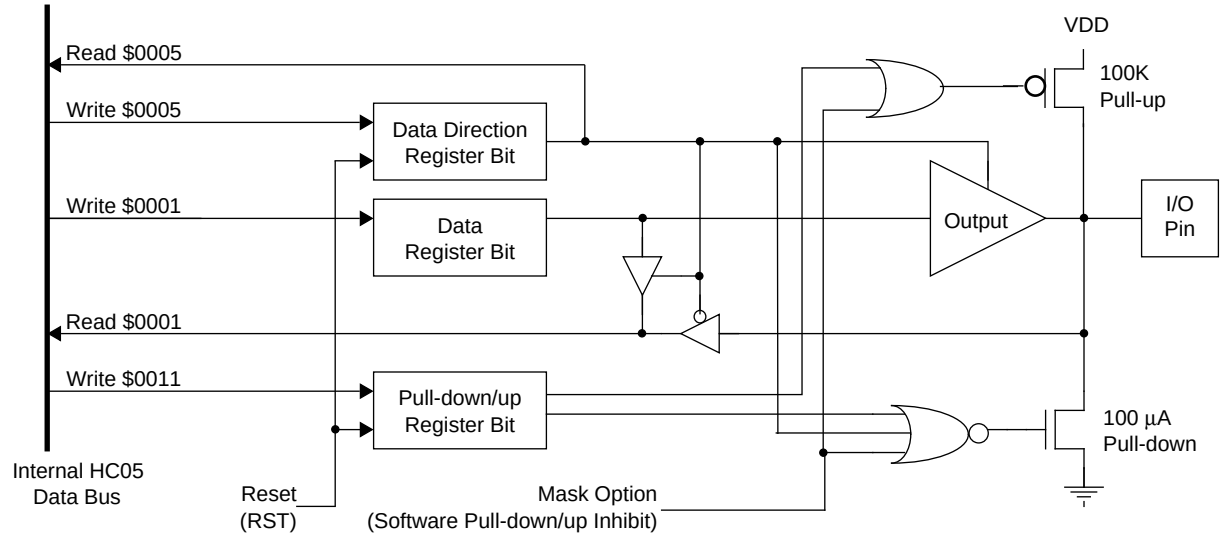
The BIH and BIL instructions will only apply to the level on the $\overline{IRQ}$ pin itself, and not to the internal IRQ input to the CPU. Therefore BIH and BIL cannot be used to test the state of the lower four Port A input pins as a group nor that of PA7.

7.3 PORT B

Port B is a 6-bit bidirectional port which functions as shown in **Figure 7-3**. Each Port B pin is controlled by the corresponding bits in a data direction register, a data register, and a pull-down/up register. The Port B Data Register is located at address $0001. The Port B Data Direction Register (DDRB) is located at address $0005. The Port B Pull-down/up Register (PDURB) is located at address $0011. Reset clears the DDRB and the PDURB. The Port B Data Register is unaffected by reset.

PB1 and PB2 are open-drained type I/Os, capable of typically sinking 25mA current each, at V_{OL} 0.5V max.

For the 16-pin package, PB1 and PB2 are connected together to form the pin labelled PB1 on the package. This PB1 pin will have a maximum sink current of 50mA if both PB1 and PB2 are written with the same value at the same write cycle.



Note: each I/O port pin can have either Pull-up and pull-down device not both

PB1 and PB2 output drivers are of open-drained type

Figure 7-3. Port B I/O Circuitry

7.3.1 Port B Data Register

All Port B I/O pins have a corresponding bit in the Port B Data Register. When a Port B pin is programmed as an output the state of the corresponding data register bit determines the state of the output pin. When a Port B pin is programmed as an input, any read of the Port B Data Register will return the logic state of the corresponding I/O pin. The Port B data register is unaffected by reset. Unused bits 6 and 7 will always read as logic zeros, and any write to these bits will be ignored. The Port B data register is unaffected by reset.

7.3.2 Port B Data Direction Register

Port B I/O pins may be programmed as an input by clearing the corresponding bit in the DDRB, or programmed as an output by setting the corresponding bit in the DDRB. The DDRB can be accessed at address $0005. Unused bits 6 and 7 will always read as logic zeros, and any write to these bits will be ignored.The DDRB is cleared by reset.

If configured as output pins, PB1 and PB2 have slow output falling-edge transition feature. The slow transition feature is controlled by the SLOWE bit of DDRB. SLOWE bit, if set and if the pin is configured as an output pin, enables the slow falling-edge output transition feature of all four I/O lines, PA6, PA7, PB1 and PB2.

For the 16-pin package type, care should be taken in using PB1 pin, which is bonded to two internal port B I/O lines PB1 and PB2, to constitute a 50 mA current sinking driver. Both PB1 and PB2 I/O lines are capable of sinking 25 mA. If they are written with the same logic 0 value in the same write cycle, PB1 pin will sink 50 mA. If they are written with different values in the same write cycle, PB1 pin will sink only 25 mA.

For the 20-pin package type, I/O lines PB1 and PB2 are not bonded to the same pin. Hence, to constitute a 50mA current sinking driver, PB1 and PB2 pins have to be tied together externally and controlled in the same way as in the16-pin package type case.

Also, if the slow transition feature of pin PB1 is enabled, a combination of I/O lines PB1 and PB2, is also a combination of slow transition features of I/O lines PB1 and PB2. PB2 line falling-edge output transition occurs $t_{CYC}/2$ after the write cycle, with a standard I/O edge transition time. Whereas for PB1 line, the falling-edge transition occurring immediately after the write cycle, but with an edge transition time slower than standard I/Os, similar to PA6 and PA7 pins.

The net result is, for the 16-pin package type, since both PB1 and PB2 I/O lines are bonded to the same PB1 pin, the combination of delayed PB1 line sharp-edge output and the non-delayed slow transition output yields the desired slow output falling-edge transition.

For the 20-pin package, PB1 and PB2 pins should be tied externally to create a driver with the desired slow output falling-edge transition feature. If SLOWE is set and PB2 pin is not tied to PB1 pin, be advised that the output at PB2 changes state $t_{CYC}/2$ after the write cycle.

7.3.3 Port B Pull-down/up Register

All Port B I/O pins may have software programmable pull-down/up devices enabled by a mask option. If the pull-down/up mask option is selected, the pull-down/up is activated whenever the corresponding bit in the PDURB is clear. A pull-down on an I/O pin is activated only if the I/O pin is programmed as an input whereas a Pull-up device on an I/O pin is always activated whenever enabled, regardless of port direction.

The PDURB is a write-only register. Any reads of location $0011 will return undefined results. Since reset clears both the DDRB and the PDURB, all pins will initialize as inputs with the pull-down devices active and pull-up devices active (if chosen via mask option).

Typical value of port B pull-up is 100KΩ.

7.4 I/O PORT PROGRAMMING

All I/O pins can be programmed as inputs or outputs, with or without pull-down/up devices.

7.4.1 Pin Data Direction

The direction of a pin is determined by the state of its corresponding bit in the associated port Data Direction Register (DDR). A pin is configured as an output if its corresponding DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared to a logic zero.

The data direction bits DDRB0 to DDRB2 and DDRA0 to DDRA7 are read/write bits which can be manipulated with read-modify-write instructions. At power-on or reset, all DDRs are cleared which configures all port pins as inputs. If the pull-down/up mask option is chosen, all pins will initially power-up with their software programmable pull-downs/ups enabled.

7.4.2 Output Pin

When an I/O pin is programmed as an output pin, the state of the corresponding data register bit will determine the state of the pin. The state of the data register bits can be altered by writing to address $0000 for Port A and address $0001 for Port B. Reads of the corresponding data register bit at address $0000 or $0001 will return the state of the data register bit (not the state of the I/O pin itself). Therefore bit manipulation is possible on all pins programmed as outputs.

If the corresponding bit in the pull-down/up register is clear (and the pull-down/up mask option is chosen), only output pins with pull-ups have an activated pull-up device connected to the pin. For those pins with pull-downs and configured as output pins, the pull-downs will be inactivated regardless of the state of the corresponding pull-down/up register bit. Since the pull-down/up register bits are write-only, bit manipulation should not be used on these register bits.

7.4.3 Input Pin

When an I/O pin is programmed as an input pin, the state of the pin can be determined by reading the corresponding data register bit. Any writes to the corresponding data register bit for an input pin will be ignored in the sense that the written value will not be reflected on the pin, rather it is only reflected in the port data register. Please refer to **Table 7-1** and **Table 7-2** for details.

If the corresponding bit in the pull-down/up register is clear (and the pull-down/up mask option is chosen) the input pin will also have an activated pull-down/up device. Since the pull-down/up register bits are write-only, bit manipulation should not be used on these register bits.

7.4.4 I/O Pin Transitions

A "glitch" can be generated on an I/O pin when changing it from an input to an output unless the data register is first preconditioned to the desired state before changing the corresponding DDR bit from a zero to a one.

If pull-downs are enabled by mask option, a floating input can be avoided by clearing the pull-down/up register bit before changing the corresponding DDR from a one to a zero. This will insure that the pull-down device will be activated before the I/O pin changes from a driven output to a pulled low/high input.

7.4.5 I/O Pin Truth Tables

Every pin on Port A and Port B may be programmed as an input or an output under software control as shown in **Table 7-1** and **Table 7-2**. All port I/O pins may also have software programmable pull-down/up devices if selected by the appropriate mask option.

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

DDRA	I/O Pin Mode	Accesses to PDURA at $0010		Accesses to DDRA @ $0004	Accesses to Data Register @ $0000	
		Read	Write	Read/Write	Read	Write
0	IN, Hi-Z	U	PDURA0-7	DDRA0-7	I/O Pin	*
1	OUT	U	PDURA0-7	DDRA0-7	PA0-7	PA0-7

U is undefined

* Does not affect input, but stored to data register

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

DDRA	I/O Pin Mode	Accesses to PDURB at $0011		Accesses to DDRB @ $0005	Accesses to Data Register @ $0001	
		Read	Write	Read/Write	Read	Write
0	IN, Hi-Z	U	PDURB0-2	DDRB0-2	I/O Pin	*
1	OUT	U	PDURB0-2	DDRB0-2	PB0-2	PB0-2

U is undefined

* Does not affect input, but stored to data register

SECTION 8
MULTI-FUNCTION TIMER

The MC68HC05J5 timer is a 15-stage multi-function ripple counter. The features include Timer Over Flow (TOF), Power-On Reset (POR), Real Time Interrupt (RTI), and COP Watchdog Timer.

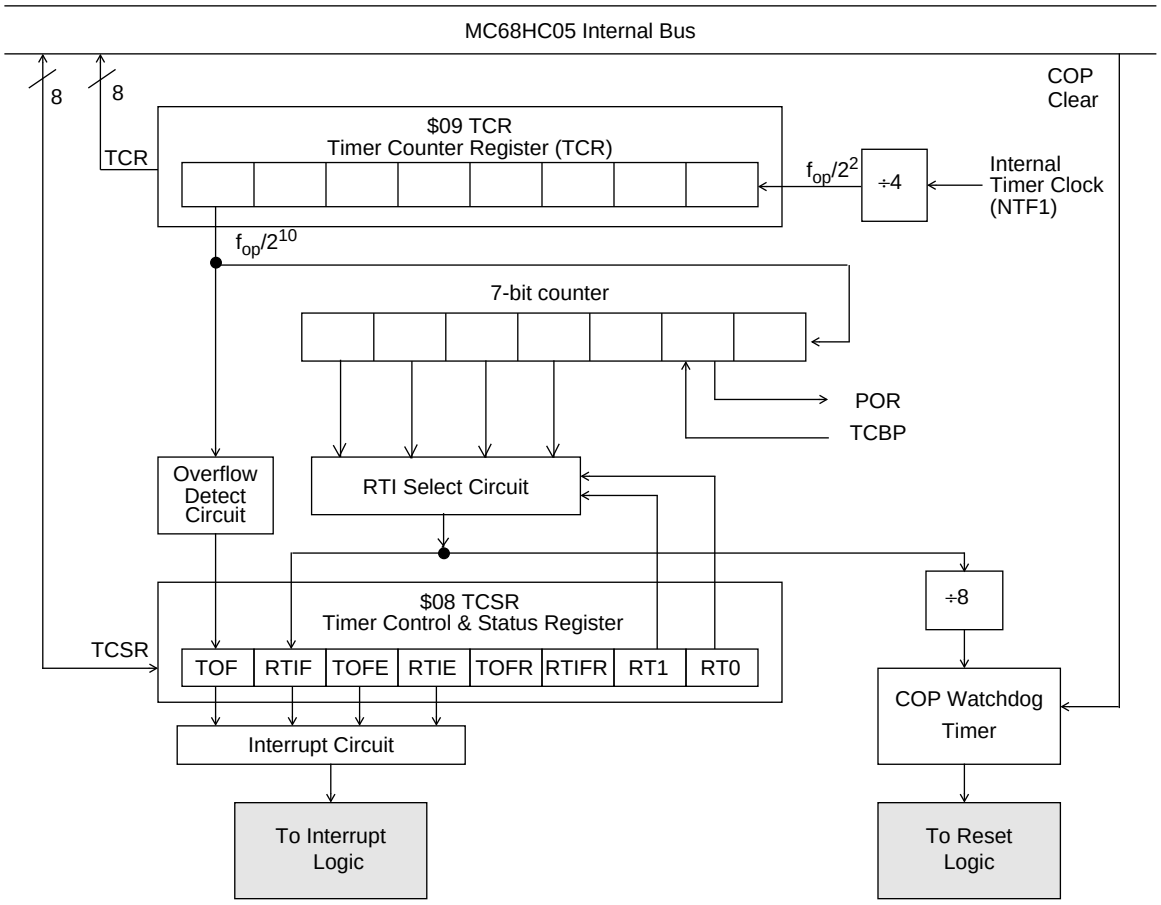


Figure 8-1. Multi-Function Timer Block Diagram

As shown in **Figure 8-1**, the Timer is driven by the timer clock, NTF1, divided by four (4). NTF1 has the same phase and frequency as the processor bus clock, PH2, but is not stopped by the WAIT or HALT Modes. This signal drives an 8-bit ripple counter. The value of this 8-bit ripple counter can be read by the CPU at any time by accessing the Timer Counter Register (TCR) at address $09. A timer overflow function is implemented on the last stage of this counter, giving a possible

interrupt at the rate of $f_{op}/1024$. Two additional stages produce the POR function at $f_{op}/4064$. The Timer Counter Bypass circuitry (available only in Expanded Test Mode) is at this point in the timer chain. This circuit is followed by two more stages, with the resulting clock ($f_{op}/16384$) driving the Real Time Interrupt circuit. The RTI circuit consists of three divider stages with a 1 of 4 selector. The output of the RTI circuit is further divided by eight to drive the optional COP Watchdog Timer circuit, which can be enabled by a mask option. The RTI rate selector bits, and the RTI and TOF enable bits and flags are located in the Timer Control and Status Register at location $08.

The Real Time Interrupt circuit consists of a three stage divider and a 1 of 4 selector. The clock frequency that drives the RTI circuit is $f_{op}/2^{14}$ (or $f_{op}/16384$) with three additional divider stages giving a maximum interrupt period of $f_{op}/2^{17}$ (or $f_{op}/131072$).

The power-on cycle 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, the timer will start counting up from zero and normal device operation will begin. If $\overline{\text{RESET}}$ is asserted at any time during operation the counter chain will be cleared.

8.1 TIMER REGISTERS

The 15-stage Multi-function Timer contains two registers: a Timer Counter Register and a Timer Control/Status Register.

8.1.1 Timer Counter Register (TCR), $09

The Timer Counter Register is a read-only register which contains the current value of the 8-bit ripple counter at the beginning of the timer chain. This counter is clocked at f_{op} divided by 4 and can be used for various functions including a software input capture. Extended time periods can be attained using the TOF function to increment a temporary RAM storage location thereby simulating a 16-bit (or more) counter. The value of each bit of the TCR is shown in **Figure 8-2**. This register is cleared by reset.

		7	6	5	4	3	2	1	0
TCR $09	R	TMR7	TMR6	TMR5	TMR4	TMR3	TMR2	TMR1	TMR0
	W								
Reset ⇒		0	0	0	0	0	0	0	0

Figure 8-2. Timer Counter Register

8.1.2 Timer Control/status Register (TCSR), $08

The TCSR contains the timer interrupt flag bits, the timer interrupt enable bits, and the real time interrupt rate select bits. Bit 2 and bit 3 are write-only bits which will read as logical zeros. **Figure 8-3** shows the value of each bit in the TCSR following reset.

		7	6	5	4	3	2	1	0
TCSR $08	R	TOF	RTIF	TOFE	RTIE	0	0	RT1	RT0
	W					TOFR	RTIFR		
Reset ⇒		0	0	0	0	0	0	1	1

Figure 8-3. Timer Control/Status Register (TCSR)

TOF - Timer Overflow Flag

The TOF is a read-only flag bit.

1 = Set when the 8-bit ripple counter rolls over from $FF to $00. A TIMER Interrupt request will be generated if TOFE is also set.

0 = Reset by writing a logical one to the TOF acknowledge bit, TOFR. Writing to the TOF flag bit has no effect on its value. This bit is cleared by reset.

RTIF - Real Time Interrupt Flag

The RTIF is a read-only flag bit.

1 = Set when the output of the chosen (1 of 4 selections) Real Time Interrupt stage goes active. A TIMER Interrupt request will be generated if RTIE is also set.

0 = Reset by writing a logical one to the RTIF acknowledge bit, RTIFR. Writing to the RTIF flag bit has no effect on its value. This bit is cleared by reset.

TOFE - Timer Overflow Enable

The TOFE is an enable bit that allows generation of a TIMER Interrupt upon overflow of the Timer Counter Register.

1 = When set, the TIMER Interrupt is generated when the TOF flag bit is set.

0 = When cleared, no TIMER interrupt caused by TOF bit set will be generated. This bit is cleared by reset.

RTIE - Real Time Interrupt Enable

The RTIE is an enable bit that allows generation of a TIMER Interrupt by the RTIF bit.

1 = When set, the TIMER Interrupt is generated when the RTIF flag bit is set.

0 = When cleared, no TIMER interrupt caused by RTIF bit set will be generated. This bit is cleared by reset.

TOFR - Timer Overflow Acknowledge

The TOFR is an acknowledge bit that resets the TOF flag bit. This bit is unaffected by reset. Reading the TOFR will always return a logical zero.

1 = Clears the TOF flag bit.
0 = Does not clear the TOF flag bit.

RTIFR - Real Time Interrupt Acknowledge

The RTIFR is an acknowledge bit that resets the RTIF flag bit. This bit is unaffected by reset. Reading the RTIFR will always return a logical zero.

1 = Clears the RTIF flag bit.
0 = Does not clear the RTIF flag bit.

RT1:RT0 - Real Time Interrupt Rate Select

The RT0 and RT1 control bits select one of four taps for the Real Time Interrupt circuit. **Table 8-1** shows the available interrupt rates for two f_{OP} values. Both the RT0 and RT1 control bits are set by reset, selecting the lowest periodic rate and therefore the maximum time in which to alter these bits if necessary. Care should be taken when altering RT0 and RT1 if the time-out 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 additional one could be generated. To avoid problems, the COP should be cleared just prior to changing RTI taps.

Table 8-1. RTI Rates and COP Reset Times

RT1:RT0	RTI RATES AT f_{OP} FREQ. SPECIFIED:			MIN. COP RESET AT f_{OP} FREQ. SPECIFIED:		
	Divider	1.000 MHz	2.000 MHz	Divider	1.000 MHz	2.000 MHz
00	16384	16.384 ms	8.192 ms	131072	131 ms	66 ms
01	32768	32.768 ms	16.384 ms	262144	262 ms	131 ms
10	65536	65.536 ms	32.768 ms	524288	524 ms	262 ms
11	131072	131.072 ms	65.536 ms	1048576	1.059 s	524 ms

8.2 COP WATCHDOG TIMER

The COP (Computer Operating Properly) Watchdog Timer function is implemented on this device by using the output of the RTI circuit and further dividing it by eight. The minimum COP reset times are listed in **Table 8-1**. If the COP circuit times out, an internal reset is generated and the reset vector is fetched. Preventing a COP time-out is done by writing a logical zero to bit 0 of address $07F0 as shown in **Figure 8-4**. The COP register is shared with a Test ROM byte. This address location is not affected by any reset signals. Reading this location will return the Test ROM byte. When the COP is cleared, only the final divide by eight stage (output of the RTI) is cleared. The COP Watchdog Timer can be enabled/disabled by a mask option.

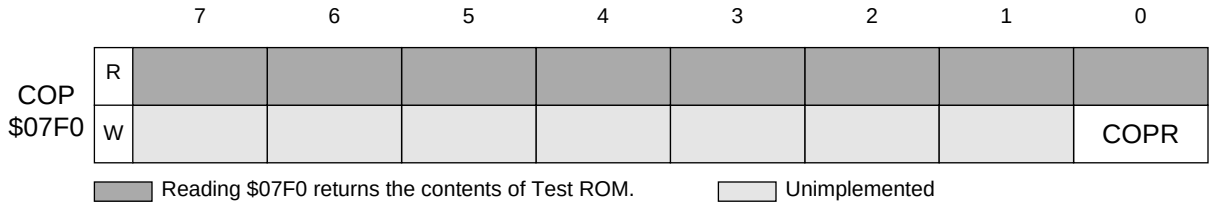


Figure 8-4. COP Watchdog Timer Location

8.3 OPERATION DURING STOP MODE

The timer system is cleared when going into STOP mode. When STOP is exited by an external interrupt or an LVR reset or an external $\overline{\text{RESET}}$, the internal oscillator will resume, followed by a 4064 internal processor oscillator stabilization delay. The timer system counter is then cleared and operation resumes. If chosen by a mask option, the STOP instruction will initiate HALT mode and the effects on the timer are as described in **Section 8.4**.

8.4 OPERATION DURING WAIT/HALT MODE

The CPU clock halts during the WAIT/HALT mode, but the timer remains active. If interrupts are enabled, a timer interrupt or custom periodic interrupt will cause the processor to exit the WAIT/HALT mode.

SECTION 9
INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. For more information on the instruction set, refer to the M6805 Family User's Manual (M6805UM/AD2) or the associated MC68HC05 Data Sheet.

9.1 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

9.2 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. Do not use these read-modify-write instructions on write-only locations. Refer to the following list of instructions.

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

9.3 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

9.4 BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any read/write bit which resides in the first 256 bytes of the memory space where all port registers, module registers, and part or all of 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 and branch functions are each 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. The bit set and bit clear instructions are also read-modify-write instructions and should not be used to manipulate write-only locations. Refer to the following list for bit manipulation instructions.

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

9.5 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

9.6 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.

9.6.1 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).

9.6.2 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 single two-byte instructions.

9.6.3 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.

9.6.4 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 +127 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.

9.6.5 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.

9.6.6 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 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 highest location which can be accessed in this way.

9.6.7 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 address 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.

9.6.8 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 register bit in the first 256 locations of memory, including I/O, can by selectively set or cleared with a single two-byte instruction.

9.6.9 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.

9.6.10 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 10
ELECTRICAL SPECIFICATIONS

10.1 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
Expanded Test Mode ($\overline{IRQ}$ Pin Only)	V_{IN}	V_{SS}–0.3 to $2V_{DD}$ +0.3	V
Current Drain Per Pin Excluding PB1, PB2, V_{DD} and V_{SS}	I	–25	mA
Operating Temperature Range (Standard) (Extended)	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 $V_{SS} \le (V_{IN}$ or $V_{OUT}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

10.2 THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic SOIC	θ_{JA}	 60 60	 °C/W °C/W

10.3 DC ELECTRICAL CHARACTERISTICS

(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 VDC, T_A = –40°C to +85°C, 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 V
Output High Voltage (I_{Load} = -0.8 mA) PA0-5, PB0, PB3-5	V_{OH}	V_{DD}–0.8	—	—	V
Output Low Voltage (I_{Load} = 1.6 mA) PA0-3, PB0, PB3-5 (I_{Load} = 8.0 mA) PA4-PA7 (I_{Load} = 25.0 mA) PB1, PB2 (note 8)	V_{OL}	— — —	— — —	0.4 0.4 0.5	V
Input High Voltage PA0-5, PB0-5, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IH}	0.7×V_{DD}	—	V_{DD}	V
Input Low Voltage PA0-5, PB0-5, $\overline{IRQ}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2×V_{DD}	V
Positive-Going Input Threshold Voltage PA6, PA7 (note 9)	V_{T+}	—	1.7	—	V
Negative-Going Input Threshold Voltage PA6, PA7 (note 9)	V_{T-}	—	1.15	—	V
Supply Current (see Notes) Run Wait Stop (LVR on) 25°C –40°C to +85°C Stop (LVR off) 25°C –40°C to +85°C	I_{DD}	 *TBD* *TBD* *TBD* *TBD* *TBD* *TBD*	 *TBD* *TBD* *TBD* *TBD* *TBD* *TBD*	 *TBD* *TBD* *TBD* *TBD* *TBD* *TBD*	 mA mA µA µA µA µA
I/O Ports Hi-Z Leakage Current PA0-7, PB0-5 (without individual pull-down/up activated)	I_{IL}	—	—	±10	µA
Input Pull-down Current PA0-5, PB0, PB3-5 (with individual pull-down activated)	I_{IL}	50	100	200	µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, OSC1	I_{IN}	—	—	±1	µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$, OSC1, OSC2/R	C_{OUT} C_{IN}	— —	— —	12 8	pF pF
Crystal/Ceramic Resonator Oscillator Mode Internal Resistor OSC1 to OSC2/R	R_{OSC}	1.0	2.0	3.0	MΩ
Pull-up Resistor PA6, PA7 (note 10) PB1, PB2	R_{PULLUP}	2 25	5 100	10 200	KΩ KΩ
LVR Trigger Voltage	V_{LVR}	*TBD*	2.8	*TBD*	V

NOTES:
1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system (MFT) active.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source to OSC1 (f_{OSC} = 2.0 MHz), all inputs 0.2 Vdc from rail; no dc loads, less than 50pF on all outputs, C_L = 20 pF on OSC2/R.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 Vdc, V_{IH} = V_{DD}–0.2 Vdc.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2/R capacitance.
8. T_A = 0°C to +40°C.
9. Minimum and maximum values of both V_{T+} and V_{T-} will be determined after enough characterization has been done. Input voltage level on PA6 or PA7 higher than 2.4V is guaranteed to be recognized as a logical one and as a logic zero if lower than 0.8V.
10. PA6 and PA7 pull-up resistor values are specified under the condition that pin voltage ranges from 0V to 2.4V.

10.4 CONTROL TIMING

(V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 VDC, T_A = –40°C to +85°C, unless otherwise noted)

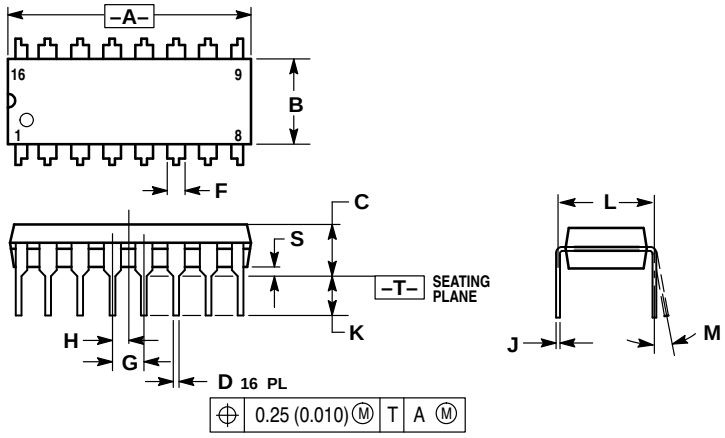
Characteristic	Symbol	Min	Max	Units
Frequency of Operation RC Oscillator Option (note 3) Crystal Oscillator Option External Clock Source	f_{OSC}	 3.8 — dc	 4.2 4.2 4.2	 MHz MHz MHz
Internal Operating Frequency Crystal Oscillator ($f_{OSC} \div 2$) RC Oscillator ($f_{OSC} \div 2$) (note 3) External Clock ($f_{OSC} \div 2$)	f_{OP}	 — 1.9 dc	 2.1 2.1 2.1	 MHz MHz MHz
Cycle Time ($1/f_{OP}$)	t_{CYC}	415	—	ns
$\overline{\text{RESET}}$ Pulse Width Low	t_{RL}	1.5	—	t_{CYC}
$\overline{\text{IRQ}}$ Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	0.5	—	t_{CYC}
$\overline{\text{IRQ}}$ Interrupt Pulse Period	t_{ILIL}	note 1	—	t_{CYC}
PA0 to PA3 Interrupt Pulse Width High (Edge-Triggered)	t_{IHIL}	0.5	—	t_{CYC}
PA0 to PA3 Interrupt Pulse Period	t_{IHIH}	note 1	—	t_{CYC}
PA7 Interrupt Pulse Width Low	t_{ILIH}	0.5	—	t_{CYC}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns
Output High to Low Transition Period on PA6, PA7, PB1	t_{SLOW}	*TBD*	*TBD*	ns

NOTES:
1. The minimum period t_{ILIL} or t_{IHIH} should not be less than the number of cycles it takes to execute the interrupt service routine plus 19 t_{CYC}.
2. Effects of processing, temperature, and supply voltage (excluding tolerances of external R and C).
3. RC oscillator: Typical center frequency is 4.0 MHz. For the specified range of the operating center frequency from 3.8MHz (min) to 4.2 MHz (max), the frequency tolerance is guaranteed to be no more than ±15% under the conditions that V_{DD} = 5.0 VDC ±10%, T_A = 0°C to +40°C and the tolerance of the external R is at most ±1%.

SECTION 11
MECHANICAL SPECIFICATION

This section provides the mechnical dimensions for the four available packages for MC68HC05J5.

11.1 16-PIN PLASTIC DUAL-IN-LINE PACKAGE (PDIP)



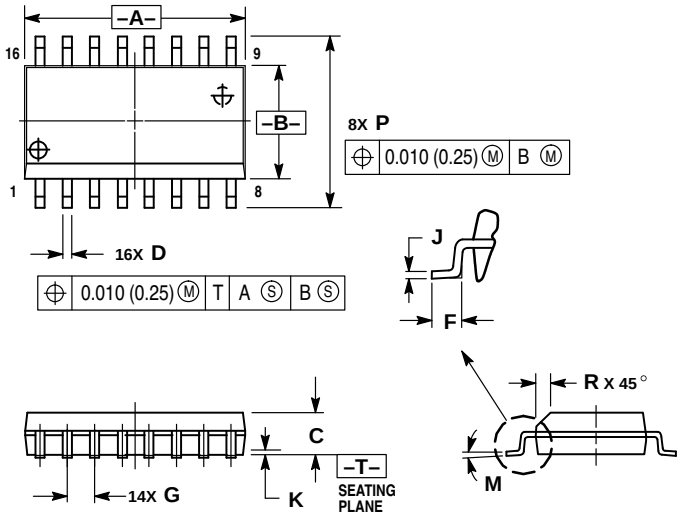
NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
4. DIMENSION B DOES NOT INCLUDE MOLD FLASH.
5. ROUNDED CORNERS OPTIONAL.

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.740	0.770	18.80	19.55
B	0.250	0.270	6.35	6.85
C	0.145	0.175	3.69	4.44
D	0.015	0.021	0.39	0.53
F	0.040	0.70	1.02	1.77
G	0.100 BSC		2.54 BSC	
H	0.050 BSC		1.27 BSC	
J	0.008	0.015	0.21	0.38
K	0.110	0.130	2.80	3.30
L	0.295	0.305	7.50	7.74
M	0°	10°	0°	10°
S	0.020	0.040	0.51	1.01

STYLE 1:
PIN 1. CATHODE
2. CATHODE
3. CATHODE
4. CATHODE
5. CATHODE
6. CATHODE
7. CATHODE
8. CATHODE
9. ANODE
10. ANODE
11. ANODE
12. ANODE
13. ANODE
14. ANODE
15. ANODE
16. ANODE

STYLE 2:
PIN 1. COMMON DRAIN
2. COMMON DRAIN
3. COMMON DRAIN
4. COMMON DRAIN
5. COMMON DRAIN
6. COMMON DRAIN
7. COMMON DRAIN
8. COMMON DRAIN
9. GATE
10. SOURCE
11. GATE
12. SOURCE
13. GATE
14. SOURCE
15. GATE
16. SOURCE

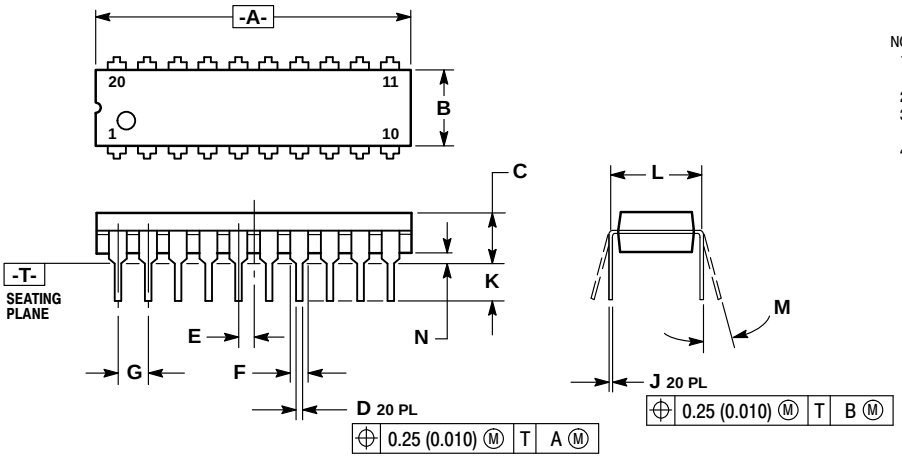
11.2 16-PIN SMALL OUTLINE INTERGRATED CIRCUIT (SOIC)



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

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	10.15	10.45	0.400	0.411
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

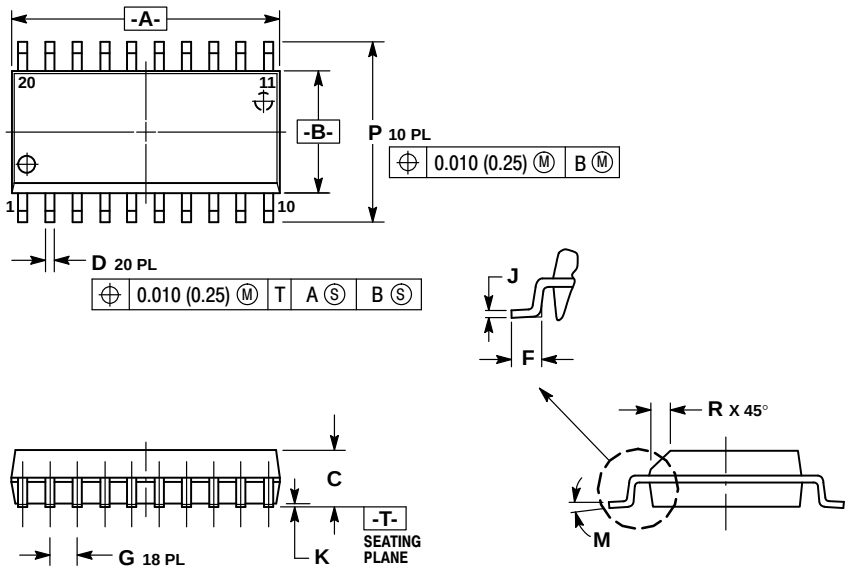
11.3 20-PIN PLASTIC DUAL-IN-LINE PACKAGE (PDIP) (CASE 738-03)



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.

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

11.4 20-PIN SMALL OUTLINE INTERGRATED CIRCUIT (SOIC) (CASE 751D-04)



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.150 (0.006) PER SIDE.
5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.13 (0.005) TOTAL IN EXCESS OF D DIMENSION AT MAXIMUM MATERIAL CONDITION.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	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

How to reach us:

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

INTERNET: http://Design-NET.com

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036. 1-800-441-2447 or 602-303-5454

JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC, 6F Seibu-Butsuryu-Center, 3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 03-81-3521-8315

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