MC68HC705J1A/D
Rev. 1

HC05

MC68HC705J1A

TECHNICAL DATA

MOTOROLA

MC68HC705J1A

HCMOS MICROCONTROLLER UNIT

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

TABLE OF CONTENTS

Paragraph **Title** **Page**

SECTION 1
GENERAL DESCRIPTION

SECTION 2
MEMORY

SECTION 3
CENTRAL PROCESSING UNIT (CPU)

TABLE OF CONTENTS

Paragraph **Title** **Page**

SECTION 4
INTERRUPTS

SECTION 5
RESETS

SECTION 6
LOW POWER MOES

SECTION 7
INPUT/OUTPUT PORTS

TABLE OF CONTENTS

Paragraph Title Page

SECTION 8
MULTIFUNCTION TIMER

SECTION 9
USER EPROM

SECTION 10
INSTRUCTION SET

TABLE OF CONTENTS

Paragraph **Title** **Page**

SECTION 11
ELECTRICAL SPECIFICATIONS

SECTION 12
MECHANICAL SPECIFICATIONS

LIST OF FIGURES

Figure **Title** **Page**

LIST OF FIGURES

Figure **Title** **Page**

LIST OF TABLES

SECTION 1
GENERAL DESCRIPTION

1.1 Introduction

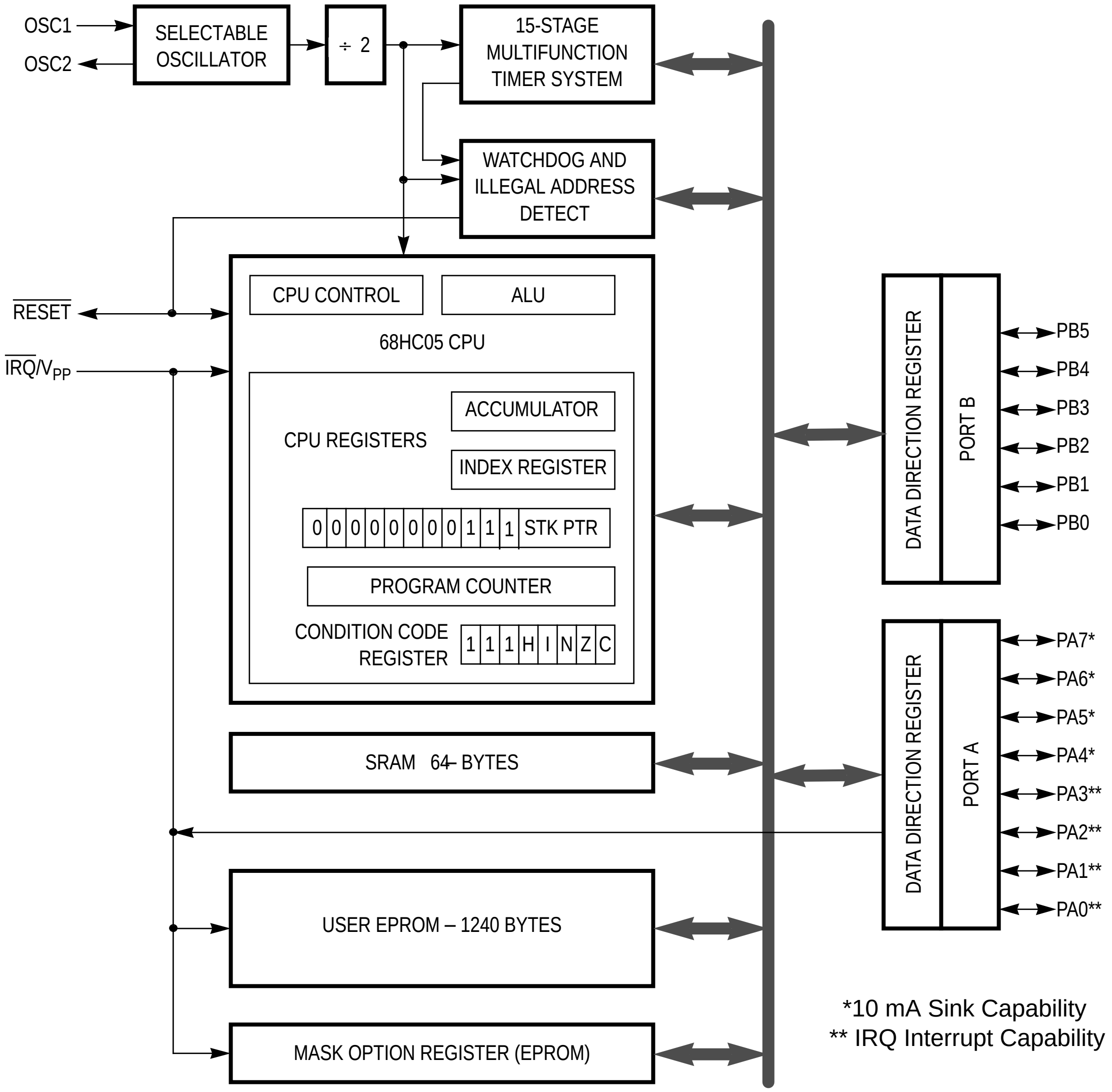
The Motorola MC68HC705J1A is an EPROM version of the MC68HC05J1A microcontroller in the M68HC05 Family of microcontrollers. The HC05 CPU core has been enhanced with a 15-stage multifunctional timer and is available in a 20-pin package with 64 bytes of RAM, 1240 bytes of user EPROM, and input/output (I/O).

1.2 Features

- Low Cost, HC05 Core
- 20-Pin Package
- 1240 Bytes of User EPROM, Including Eight Bytes of User Vectors
- 64 Bytes of User RAM
- 15-Stage Multifunction Timer Including:
  - 8-Bit Free-Running Counter
  - Four-Stage Selectable Real-Time Interrupt Generator
- 14 Bidirectional I/O Lines Including:
  - 10 mA Sink Capability on Four I/O Pins
  - MOR (Mask Option Register) and Software Programmable Pulldowns on All I/O pins
  - MOR Selectable Interrupt on Four I/O Pins (Keyboard Scan Feature)
- MOR Selectable Sensitivity on IRQ Interrupt (Edge- and Level-Sensitive or Edge-Sensitive Only)
- Software Mask and Request Bit for IRQ Interrupt
- On-Chip Oscillator (Crystal/Ceramic Resonator or RC Oscillator)
- MOR Selectable Parallel Oscillator Bias Resistor (Approximately 2 MΩ)
- MOR Selectable Computer Operating Properly (COP) Watchdog System
- Power Saving Stop and Wait Mode Instructions (MOR Selectable STOP Instruction Conversion to HALT)
- Illegal Address Reset
- Internal Steering Diode and Pullup Resistor on $\overline{\text{RESET}}$ Pin to V_{DD}

1.3 MCU Structure

Figure 1-1 shows the structure of the MC68HC705J1A.



*10 mA Sink Capability
** IRQ Interrupt Capability

Figure 1-1. Block Diagram

NOTE

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

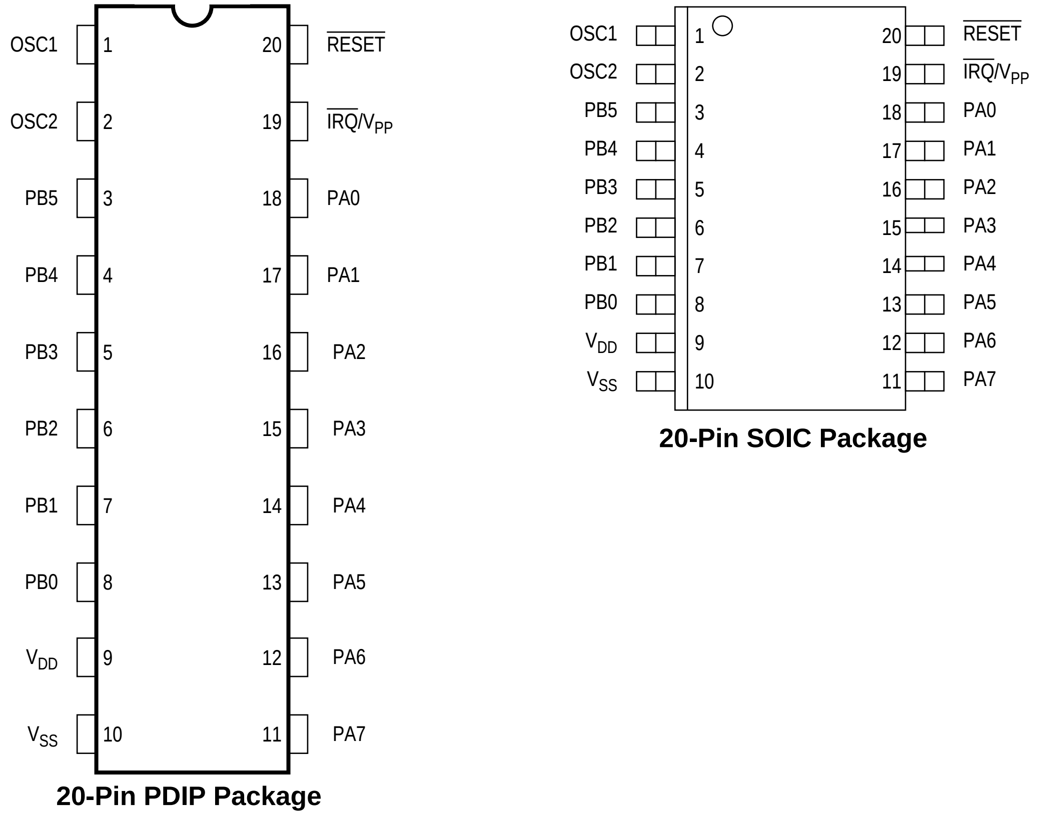
NOTE

Any reference to voltage, current, or frequency specified in the following sections will refer to the nominal values. The exact values and their tolerances or limits are specified in **SECTION 11 ELECTRICAL SPECIFICATIONS.**

1.4 Functional Pin Description

Figure 1-2 shows the MC68HC705J1A pin assignments for both the plastic dual-in-line package (PDIP) and the small outline integrated circuit (SOIC) package.

A description of the general function of each pin is given in the following paragraphs. Where applicable, reference is made to other sections for more detailed information.



20-Pin SOIC Package

20-Pin PDIP Package

Figure 1-2. Pin Assignments

1.4.1 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. As shown in Figure 1-3, this can be accomplished 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.

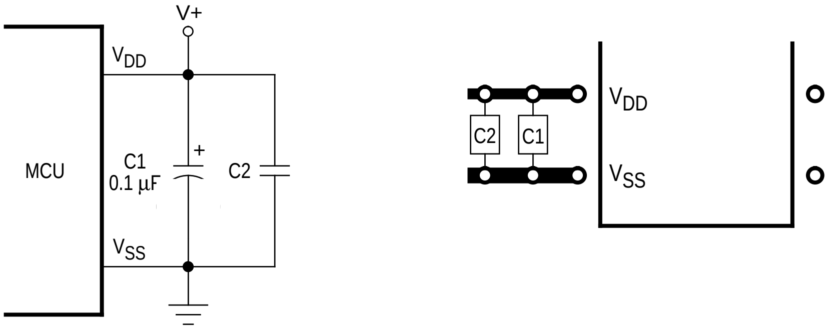


Figure 1-3. Bypassing Layout Recommendation

1.4.2 OSC1 and OSC2

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

1. A crystal as shown in Figure 1-4 and Figure 1-5
2. A ceramic resonator as shown in Figure 1-6 and Figure 1-7
3. An RC oscillator
4. An external clock signal as shown in Figure 1-8

The frequency, f_{osc}, of the oscillator or external clock source is divided by two to produce the internal operating frequency, f_{op}. An internal 2 MΩ resistor may be selected between OSC1 and OSC2 by a programmable mask option.

Crystal Oscillator

Figure 1-4 and Figure 1-5 show a typical crystal oscillator circuit for an AT-cut, parallel resonant crystal. Follow the crystal supplier’s recommendations, as the crystal parameters determine the external component values required to provide reliable start-up and maximum stability. The load capacitance values used in the oscillator circuit design should include all stray layout capacitances.

To minimize output distortion, mount the crystal and capacitors as close as possible to the pins. An internal start-up resistor of approximately 2 MΩ is provided between OSC1 and OSC2 for the crystal type oscillator as a programmable mask option.

NOTE

Use an AT-cut crystal and not an AT-strip crystal. The MCU may overdrive an AT-strip crystal.

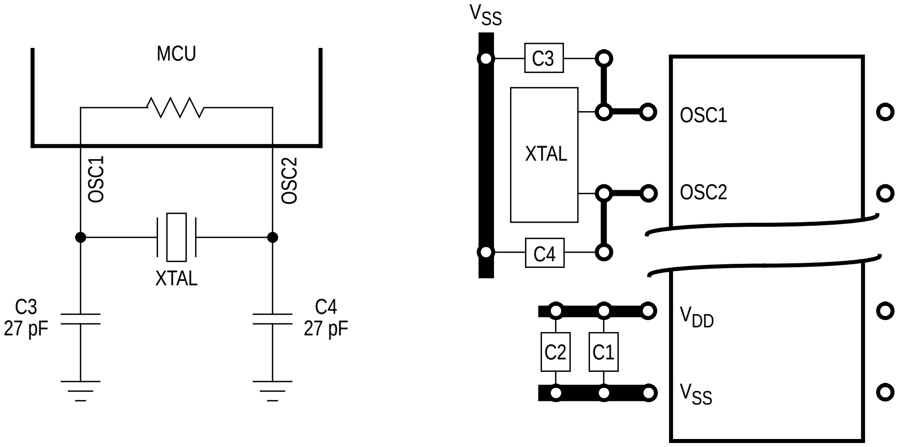


Figure 1-4. Crystal Connections with Feedback Resistor Mask Option

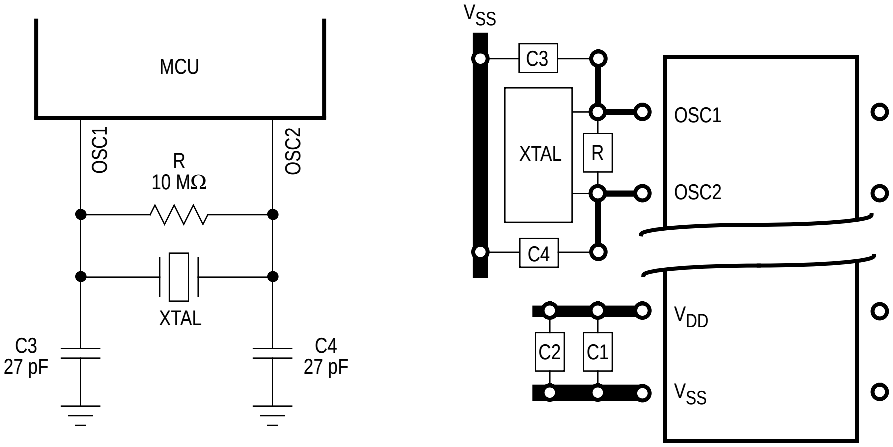


Figure 1-5. Crystal Connections without Feedback Resistor Mask Option

Ceramic Resonator Oscillator

In cost-sensitive applications, a ceramic resonator can be used in place of the crystal. The circuit shown in Figure 1-6 or Figure 1-7 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 for the ceramic resonator-type oscillator as a MOR programmable mask option.

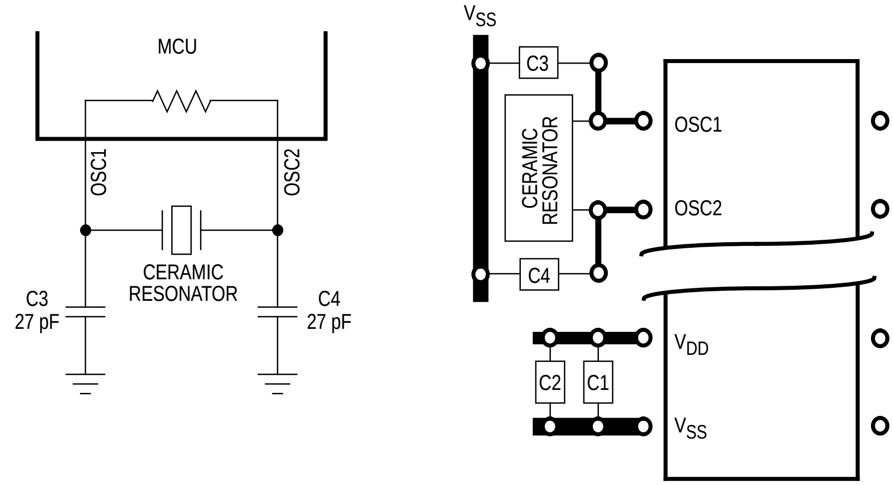


Figure 1-6. Ceramic Resonator Connections with Feedback Resistor Mask Option

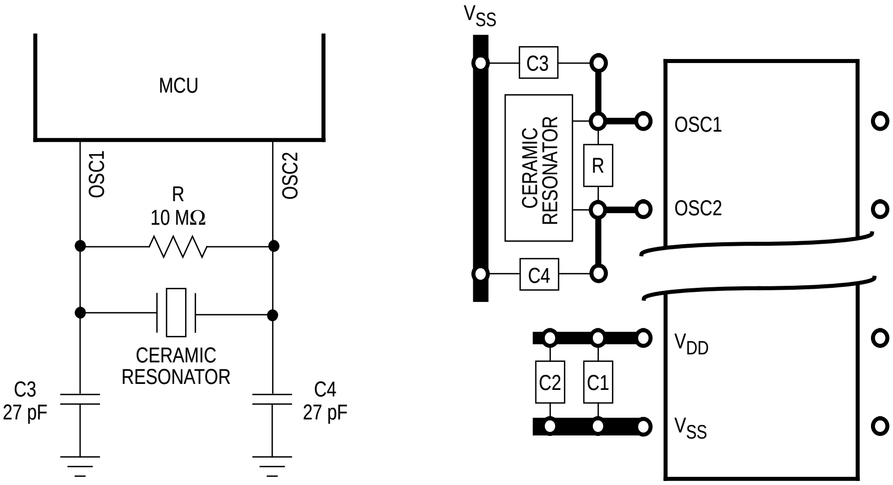


Figure 1-7. Ceramic Resonator Connections without Feedback Resistor Mask Option

RC Oscillator

The RC oscillator option is not yet available but will be in the near future. Please consult the factory for availability schedule.

External Clock

An external clock from another CMOS-compatible device can be connected to the OSC1 input, with the OSC2 input not connected, as shown in Figure 1-8. This configuration is possible regardless of whether the crystal/ceramic resonator or the RC oscillator is enabled.

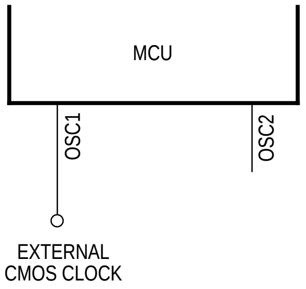


Figure 1-8. External Clock Connections

1.4.3 $\overline{\text{RESET}}$

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. It functions as an output to indicate that an internal COP watchdog or illegal address reset has occurred. An internal resistor to V_{DD} pulls the RESET pin high. When power is removed, the RESET pin has a steering diode to discharge any voltage on the pin to V_{DD}. The RESET pin contains an internal Schmitt trigger to improve its noise immunity as an input. Refer to **SECTION 5 RESETS**.

NOTE

The resistor and steering diode are not available on the MC68HC05J1A (ROM version of this device).

1.4.4 $\overline{\text{IRQ}}/V_{PP}$ (Maskable Interrupt Request/Programming Voltage)

This input pin drives the asynchronous IRQ interrupt function of the CPU. The IRQ interrupt function has a MOR bit 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 IRQ/V_{PP} input requires an external resistor to V_{DD} for “wired-OR” operation, if desired. If the IRQ/V_{PP} pin is not used, it must be tied to the V_{DD} supply. The IRQ/V_{PP} pin contains an internal Schmitt trigger as part of its input to improve noise immunity. The voltage on this pin may affect the mode of operation and should not exceed V_{DD}**.** See **SECTION 4 INTERRUPTS** for more details on the interrupts.

NOTE

Each of the PA0 through PA3 I/O pins may be connected as an OR function with the IRQ interrupt function by a MOR bit. This capability allows keyboard scan applications where the transitions or levels on the I/O pins will behave the same as the $\overline{\text{IRQ}}/V_{PP}$ pin, except for the inverted phase. The edge or level sensitivity selected by a separate MOR bit for the $\overline{\text{IRQ}}/V_{PP}$ pin also applies to the I/O pins OR'ed to create the $\overline{\text{IRQ}}$ signal.

The $\overline{\text{IRQ}}/V_{PP}$ pin is also used for programming voltage when programming the user EPROM and MOR.

1.4.5 PA0 Through PA7

These eight I/O lines comprise port A, a general-purpose bidirectional I/O port. See **7.2 Port A** for more details on port A.

1.4.6 PB0 Through PB5

These six I/O lines comprise port B, a general-purpose bidirectional I/O port. See **7.3 Port B** for more details on port B.

SECTION 2
MEMORY

2.1 Introduction

This section describes the organization of the on-chip memory.

2.2 Memory Map

The CPU can address 2 Kbytes of memory space. The EPROM portion of memory holds the program instructions, fixed data, user-defined vectors, and interrupt service routines. The RAM portion of memory holds variable data. I/O registers are memory-mapped so that the CPU can access their locations in the same way that it accesses all other memory locations. See Figure 2-1.

2.3 Input/Output Section

The first 32 addresses of the memory space, $0000–$001F, comprise the I/O section. These are the addresses of the I/O control registers, status registers, and data registers.

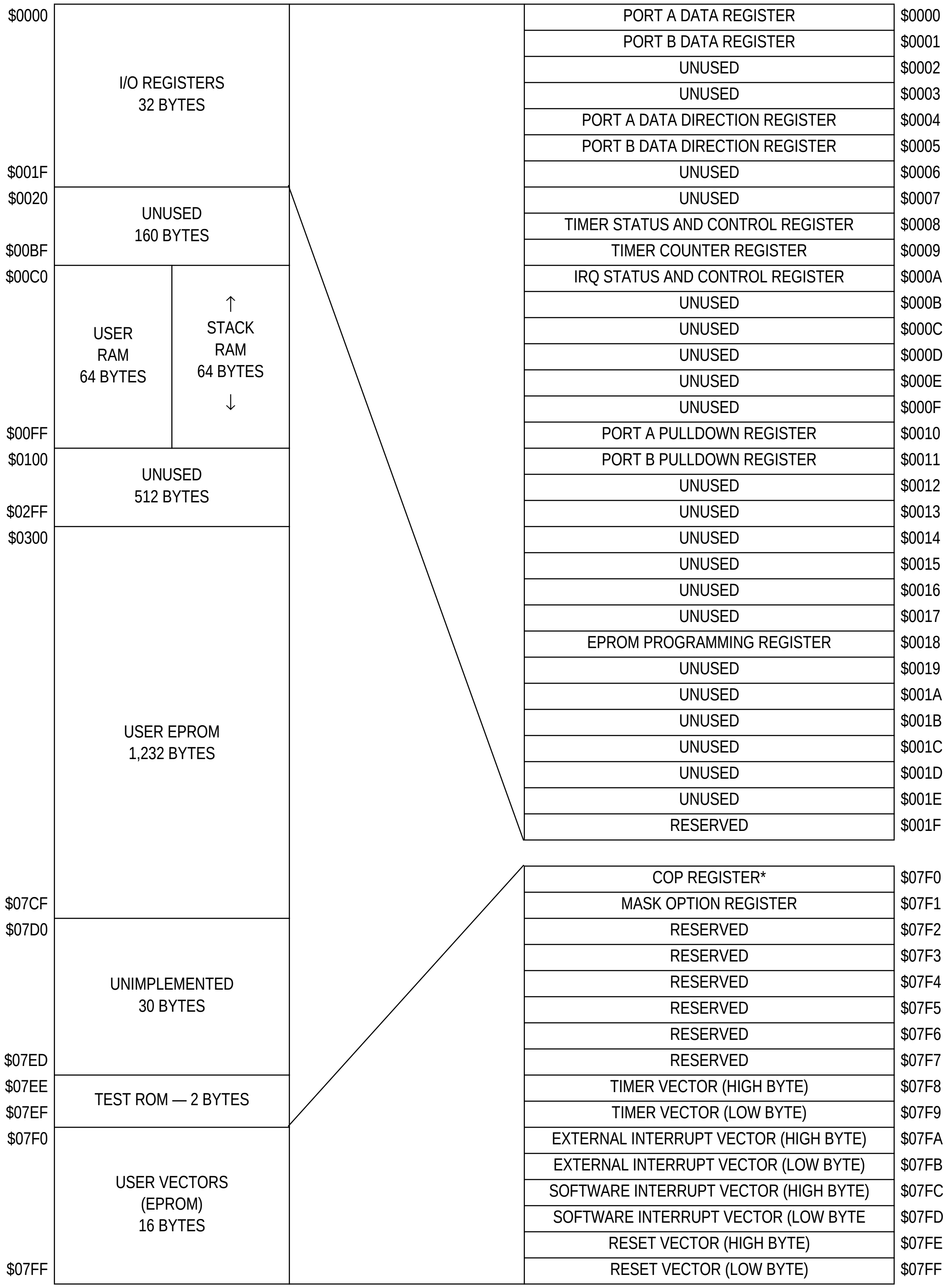
One I/O register is located outside the 32-byte I/O section: the computer operating properly (COP) register is mapped at $07F0.

2.4 RAM

The 64 addresses from $00C0 to $00FF serve as both the user RAM and the stack RAM. The CPU uses five RAM bytes to save all CPU register contents before processing an interrupt. During a subroutine call, the CPU uses two bytes to store the return address. The stack pointer decrements during pushes and increments during pulls.

NOTE

Be careful when using nested subroutines or multiple interrupt levels. The CPU may overwrite data in the RAM during a subroutine or during the interrupt stacking operation.



PORT A DATA REGISTER	$0000
PORT B DATA REGISTER	$0001
UNUSED	$0002
UNUSED	$0003
PORT A DATA DIRECTION REGISTER	$0004
PORT B DATA DIRECTION REGISTER	$0005
UNUSED	$0006
UNUSED	$0007
TIMER STATUS AND CONTROL REGISTER	$0008
TIMER COUNTER REGISTER	$0009
IRQ STATUS AND CONTROL REGISTER	$000A
UNUSED	$000B
UNUSED	$000C
UNUSED	$000D
UNUSED	$000E
UNUSED	$000F
PORT A PULLDOWN REGISTER	$0010
PORT B PULLDOWN REGISTER	$0011
UNUSED	$0012
UNUSED	$0013
UNUSED	$0014
UNUSED	$0015
UNUSED	$0016
UNUSED	$0017
EPROM PROGRAMMING REGISTER	$0018
UNUSED	$0019
UNUSED	$001A
UNUSED	$001B
UNUSED	$001C
UNUSED	$001D
UNUSED	$001E
RESERVED	$001F

COP REGISTER*	$07F0
MASK OPTION REGISTER	$07F1
RESERVED	$07F2
RESERVED	$07F3
RESERVED	$07F4
RESERVED	$07F5
RESERVED	$07F6
RESERVED	$07F7
TIMER VECTOR (HIGH BYTE)	$07F8
TIMER VECTOR (LOW BYTE)	$07F9
EXTERNAL INTERRUPT VECTOR (HIGH BYTE)	$07FA
EXTERNAL INTERRUPT VECTOR (LOW BYTE)	$07FB
SOFTWARE INTERRUPT VECTOR (HIGH BYTE)	$07FC
SOFTWARE INTERRUPT VECTOR (LOW BYTE	$07FD
RESET VECTOR (HIGH BYTE)	$07FE
RESET VECTOR (LOW BYTE)	$07FF

*Writing to bit 0 of $07F0 clears the COP watchdog.

Figure 2-1. Memory Map

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

▨ = Unimplemented R = Reserved for factory test

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

ADDR	REGISTER	READ WRITE	Bit 7	6	5	4	3	2	1	Bit 0
$0010	PORT A PULLDOWN REGISTER — PDRA	READ								
		WRITE	PDIA7	PDIA6	PDIA5	PDIA4	PDIA3	PDIA2	PDIA1	PDIA0
$0011	PORT B PULLDOWN REGISTER — PDRB	READ								
		WRITE			PDIB5	PDIB4	PDIB3	PDIB2	PDIB1	PDIB0
$0012	UNIMPLEMENTED	READ								
		WRITE								
$0013	UNIMPLEMENTED	READ								
		WRITE								
$0014	UNIMPLEMENTED	READ								
		WRITE								
$0015	UNIMPLEMENTED	READ								
		WRITE								
$0016	UNIMPLEMENTED	READ								
		WRITE								
$0017	UNIMPLEMENTED	READ								
		WRITE								
$0018	EPROM PROGRAMMING EPROG	READ	0	0	0	0	0	ELAT	MPGM	EPGM
		WRITE		R	R	R	R			
$0010	UNIMPLEMENTED	READ								
		WRITE								
$001A	UNIMPLEMENTED	READ								
		WRITE								
$001B	UNIMPLEMENTED	READ								
		WRITE								
$001C	UNIMPLEMENTED	READ								
		WRITE								
$001D	UNIMPLEMENTED	READ								
		WRITE								
$001E	UNIMPLEMENTED	READ								
		WRITE								
$001F	RESERVED FOR TEST	READ	R	R	R	R	R	R	R	R
		WRITE								

□ = Unimplemented R = Reserved for factory test

Figure 2-3. I/O Registers $0010-$001F

2.5 EPROM

The EPROM is located in three areas of the memory map:

- Addresses $0300–$07CF contain 1232 bytes of user EPROM
- Addresses $07F8–07FF contain 8 bytes of EPROM reserved for user vectors
- Address $07F1 contains one byte of EPROM for mask option register

2.6 COP Register

A write-only register location is provided at $07F0 to reset the COP watchdog timer. See **8.4 COP Watchdog**.

SECTION 3
CENTRAL PROCESSING UNIT (CPU)

3.1 Introduction

This section describes the CPU registers.

3.1.1 CPU Registers

Figure 3-1 shows the five CPU registers. CPU registers are not part of the memory map.

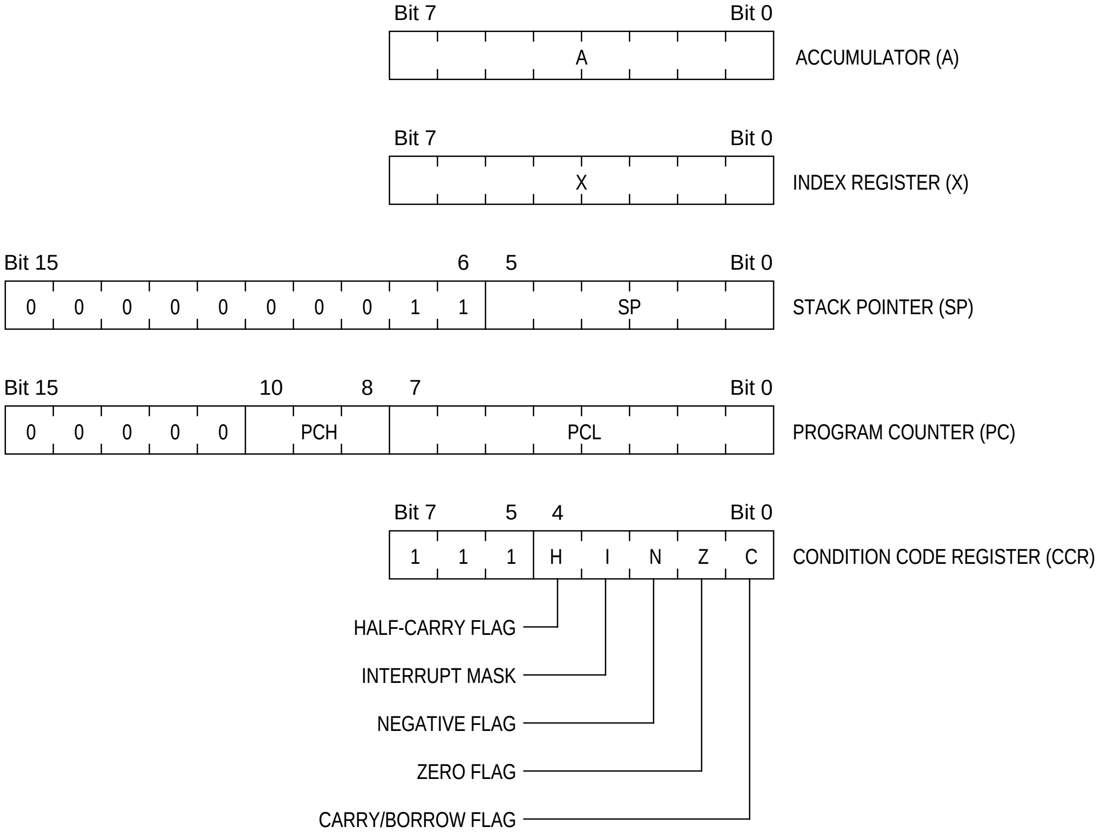


Figure 3-1. Programming Model

3.1.2 Accumulator

The accumulator is a general-purpose 8-bit register. The CPU uses the accumulator to hold operands and results of arithmetic and non-arithmetic operations.

	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	Unaffected by Reset							

Figure 3-2. Accumulators

3.1.3 Index Register

In the indexed addressing modes, the CPU uses the byte in the index register to determine the conditional address of the operand.

	Bit 7	6	5	4	3	2	1	Bit 0
Reset:	Unaffected by Reset							

Figure 3-3. Index Register

The 8-bit index register can also serve as a temporary data storage location.

3.1.4 Stack Pointer

The stack pointer is a 16-bit register that contains the address of the next location on the stack. During a reset or after the reset stack pointer (RSP) instruction, the stack pointer is preset to $00FF. The address in the stack pointer decrements as data is pushed onto the stack and increments as data is pulled from the stack.

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
Reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Figure 3-4. Stack Pointer

The ten most significant bits of the stack pointer are permanently fixed at 000000011, so the stack pointer produces addresses from $00C0 to $00FF. If subroutines and interrupts use more than 64 stack locations, the stack pointer wraps around to address $00FF and begins writing over the previously stored data. A subroutine uses two stack locations; an interrupt uses five locations.

3.1.5 Program Counter

The program counter is a 16-bit register that contains the address of the next instruction or operand to be fetched. The five most significant bits of the program counter are ignored internally and appear as 00000.

Normally, the address in the program counter automatically 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.

	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
	0	0	0	0	0											
Reset:	0	0	0	0	0	Loaded with Vector from $07FE and $07FF										

Figure 3-5. Program Counter

3.1.6 Condition Code Register

The condition code register is an 8-bit register whose three most significant bits are permanently fixed at 111. The condition code register contains the interrupt mask and four flags that indicate the results of the instruction just executed. The following paragraphs describe the functions of the condition code register.

	Bit 7	6	5	4	3	2	1	Bit 0
	1	1	1	H	I	N	C	Z
Reset:	1	1	1	U	1	U	U	U

U = Unaffected

Figure 3-6. Condition Code Register

Half-Carry Flag (H)

The CPU sets the half-carry flag when a carry occurs between bits 3 and 4 of the accumulator during an ADD or ADC operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations.

Interrupt Mask (I)

Setting the interrupt mask disables interrupts. If an interrupt request occurs while the interrupt mask is logic zero, the CPU saves the CPU registers on the stack, sets the interrupt mask, and then fetches the interrupt vector. If an interrupt request occurs while the interrupt mask is set, the interrupt request is latched. Normally, the CPU processes the latched interrupt as soon as the interrupt mask is cleared again.

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

Negative Flag (N)

The CPU sets the negative flag when an arithmetic operation, logical operation, or data manipulation produces a negative result.

Carry/Borrow Flag (C)

The CPU sets the carry/borrow flag when an addition operation produces a carry out of bit 7 of the accumulator or when a subtraction operation requires a borrow. Some logical operations and data manipulation instructions also clear or set the carry/borrow flag.

Zero Flag (Z)

The CPU sets the zero flag when an arithmetic operation, logical operation, or data manipulation produces a result of $00.

3.2 Arithmetic/Logic Unit (ALU)

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

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

SECTION 4
INTERRUPTS

4.1 Introduction

The MCU can be interrupted four different ways:

1. Non-maskable Software Interrupt Instruction (SWI)
2. External Asynchronous Interrupt (IRQ)
3. Optional External Interrupt via IRQ on PA0 through PA3 (Enabled via MOR PIRQ Bit)
4. Internal Timer Interrupt

4.2 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 in the same way 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 through $07FF. Refer to Table 4-1.

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.

Table 4-1. Vector Address for Interrupts and Reset

Register	Flag Name	Interrupts	CPU Interrupt	Vector Address
N/A	N/A	Reset	RESET	$07FE–$07FF
N/A	N/A	Software	SWI	$07FC–$07FD
ISCR	IRQF	External Interrupt	IRQ	$07FA–$07FB
TSCR	TOF	Timer Overflow	TIMER	$07F8–$07F9
TSCR	RTIF	Real Time Interrupt	TIMER	$07F8–$07F9

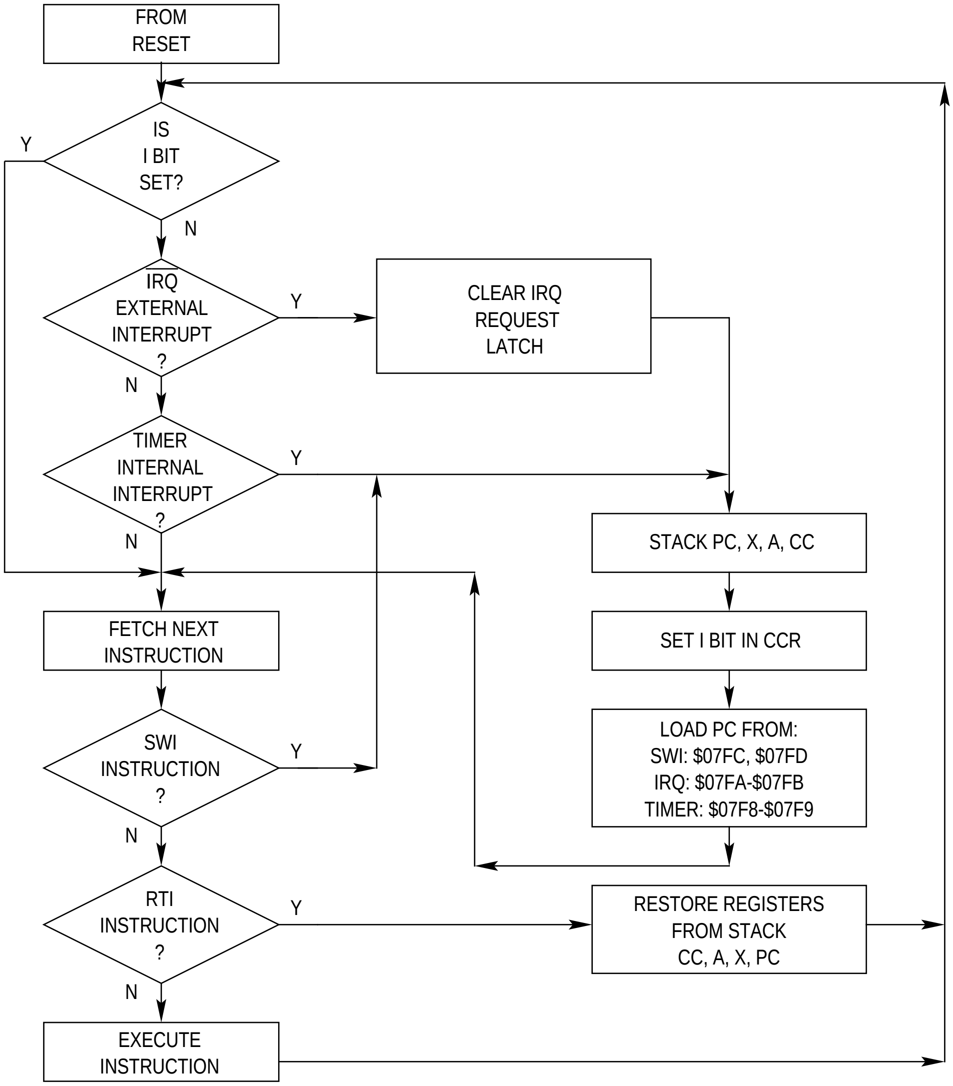


Figure 4-1. Interrupt Processing Flowchart

4.3 Reset Interrupt Sequence

The reset function is not in the strictest sense an interrupt; however, it is acted upon in a similar manner as shown in Figure 4-1. A low level input on the $\overline{\mathrm{RESET}}$ 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. Also, the I bit in the condition code register is set and the MCU is configured to a known state as described in **SECTION 5 RESETS**.

4.4 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.

4.5 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. The two types of hardware interrupts are explained in the following paragraphs.

4.6 External Interrupt (IRQ)

The $\overline{\mathrm{IRQ}}/V_{PP}$ pin provides an asynchronous interrupt to the CPU. A block diagram of the IRQ function is shown in Figure 4-2.

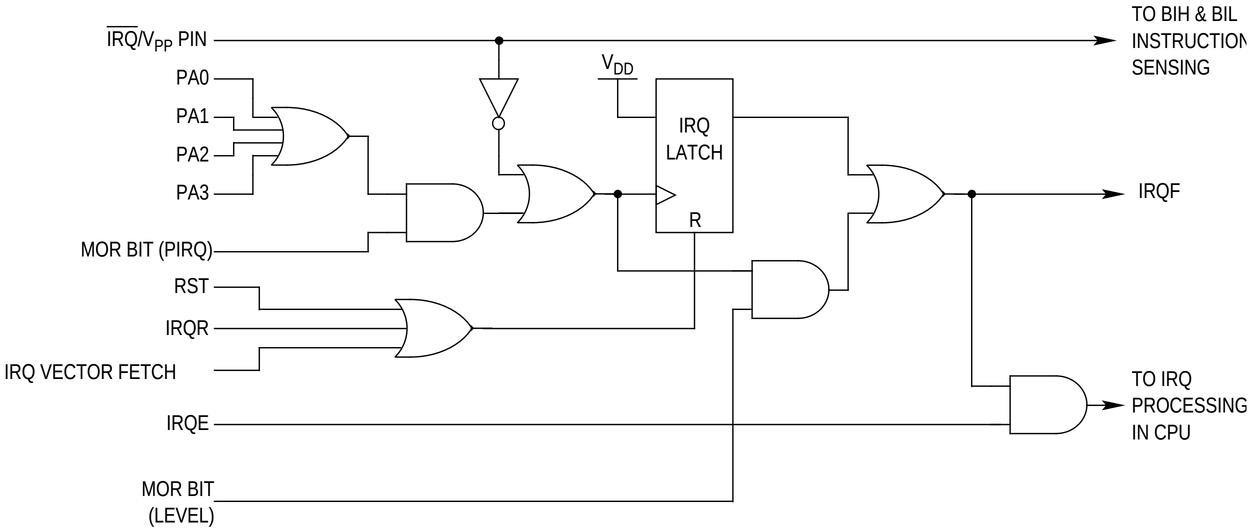


Figure 4-2. IRQ Function Block Diagram

The $\overline{\text{IRQ}}/V_{PP}$ pin is one source of an IRQ interrupt. And, the PIRQ MOR bit can also enable the four lower port A pins (PA0 through PA3) to act as other IRQ interrupt sources. These sources are all combined into a single OR'ing function to be latched by the IRQ latch.

Any enabled IRQ interrupt source sets the IRQ latch on the falling edge of the $\overline{\text{IRQ}}/V_{PP}$ pin or on the rising edge of a port A pin. If "edge-only" sensitivity is chosen by the level MOR bit, only the IRQ latch output can activate an IRQF flag which creates a request to the CPU to generate the IRQ interrupt sequence. This makes the IRQ interrupt sensitive to the following cases:

1. Falling edge on the $\overline{\text{IRQ}}/V_{PP}$ pin
2. Rising edge on any PA0-PA3 pin with port IRQ enabled (via PIRQ).

As long as any one port A IRQ input or the IRQ input remains at the active level, the port A IRQ inputs and the IRQ input are effectively ignored. If level sensitivity is chosen, the active high state of the IRQ input can also activate an IRQF flag which creates an IRQ request to the CPU to generate the IRQ interrupt sequence. This makes the IRQ interrupt sensitive to the following cases:

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

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. The user may also clear the IRQ latch by writing a logic one to the IRQR reset bit in the ISCR. Also, 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.

4.6.1 IRQ Status/Control Register (ISCR)

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

		Bit 7	6	5	4	3	2	1	Bit 0
ISCR $000A	Read:	IRQE	0	0	0	IRQF	0	0	0
	Write:		▨	▨	R	▨	▨	IRQR	▨
	Reset:	1	0	0	0	0	0	0	0

▨ = Unimplemented R = Reserved for factory test

Figure 4-3. IRQ Status and Control Register

IRQR — IRQ Interrupt Reset
 1 = Clears the IRQ interrupt and IRQF bit
 0 = No effect on IRQ interrupt and IRQF bit

IRQF — IRQ Interrupt Request Flag
 1 = External interrupt request pending
 0 = No external interrupt request pending

IRQE — IRQ Interrupt Enable
 1 = IRQ interrupts enabled — IRQF initiates the IRQ interrupt sequence
 0 = IRQ interrupts disabled — IRQF does not initiate the IRQ interrupt sequence

Execution of the STOP or WAIT instructions causes the IRQE bit to be set to allow the external IRQ to exit these modes. In addition, reset also sets the I bit which masks all interrupt sources.

4.6.2 Optional External Interrupts (PA0 through PA3)

The IRQ interrupt can also be triggered by the inputs on the PA0 through PA3 port pins if enabled by a single MOR bit, PIRQ. 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 $\overline{\text{IRQ}}/V_{PP}$ pin. This MOR bit allows all of these input pins to be OR'ed with the input present on the $\overline{\text{IRQ}}/V_{PP}$ pin. All PA0 through PA3 pins must be selected as a group as an additional IRQ interrupt. All the port A interrupt sources are also controlled by the IRQE enable bit.

NOTE

The BIH and BIL instructions will apply only to the level on the $\overline{\text{IRQ}}/V_{PP}$ pin itself, and not to the output of the logic OR function with the PA0 through 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 through PA3 pins will cause an IRQ interrupt regardless of whether these pins are configured as inputs or outputs.

NOTE

The $\overline{\text{IRQ}}$ pin has an internal Schmitt trigger. The optional external interrupts (PA0–PA3) do not have internal Schmitt triggers.

4.6.3 Timer Interrupts (TIMER)

The TIMER interrupt is generated by the multifunction timer when either a timer overflow or a real time interrupt has occurred as described in **SECTION 8 MULTIFUNCTION TIMER**. The interrupt flags and enable bits for the timer interrupts are located in the timer status and control register (TSCR) located at $0008. The I bit in the CCR must be clear 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

5.1 Introduction

The MCU can be reset from four sources: one external input and three internal restart conditions. The $\overline{\text{RESET}}$ pin is an input with a Schmitt trigger as shown in Figure 5-1. It is also an output pin which indicates the occurrence of an internal reset condition. All the peripheral modules which drive external pins will be reset by the synchronous reset signal (RST) coming from a latch, which is synchronized to the PH2 bus clock and set by any of the four reset sources.

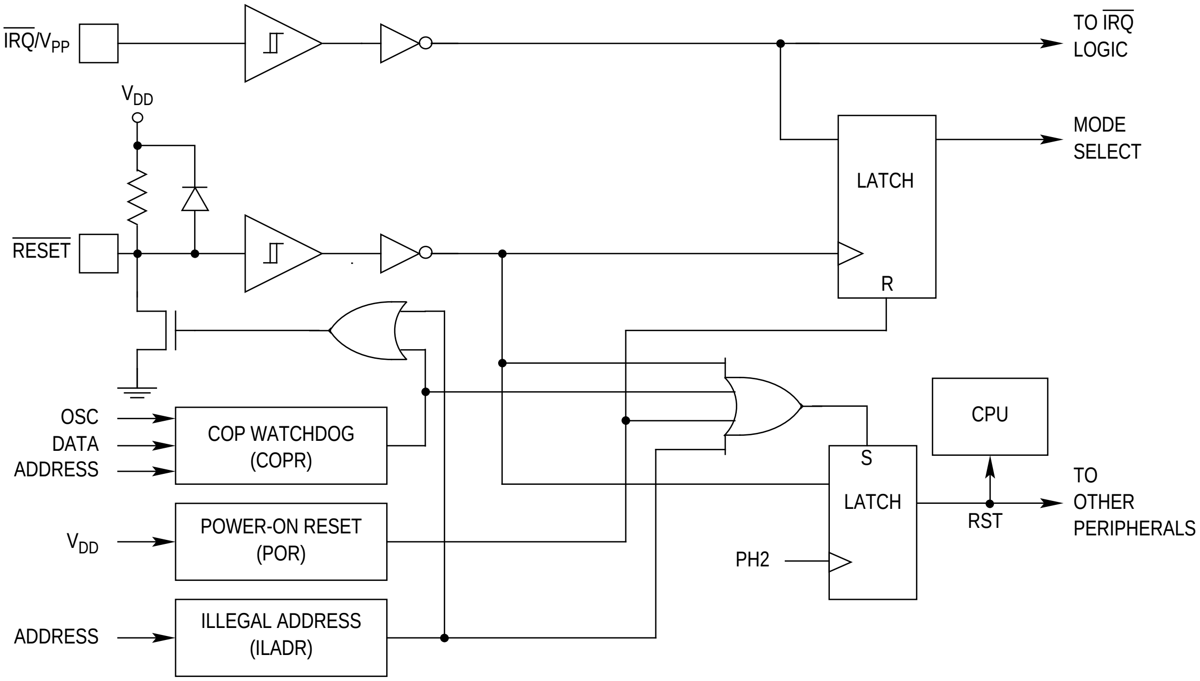


Figure 5-1. Reset Block Diagram

5.2 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 I/O pin will generate the RST signal and reset the CPU and peripherals.

As shown in Figure 5-1, an internal resistor to V_{DD} pulls the $\overline{\text{RESET}}$ pin high. When power is removed, the $\overline{\text{RESET}}$ pin has a steering diode to discharge any voltage on the pin to V_{DD}.

NOTE

The resistor and steering diode are not available on the MC68HC05J1A (ROM version of this device).

NOTE

Activation of the RST signal is generally referred to as reset of the device, unless otherwise specified.

5.3 Internal Resets

The three internally generated resets are the initial power-on reset function, the COP watchdog timer reset, and the illegal address detector reset.

5.3.1 Power-On Reset (POR)

The internal POR is generated on power-up to allow the clock oscillator to stabilize. The POR will generate the RST signal which will reset the CPU. If any other reset function is active at the end of this delay, the RST signal will remain in the reset condition until the other reset condition(s) end.

A power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset function is strictly for power turn-on conditions and should not be used to detect drops in the power supply voltage (brown out). The power-on circuitry provides an oscillator stabilization delay of 4,064 (or 16, if the MOR[SOSCD] bit is set) internal processor bus clock cycles after the oscillator becomes active. If the external $\overline{\text{RESET}}$ pin is low at the end of this delay, the processor remains in the reset state until $\overline{\text{RESET}}$ goes high. The user must ensure that the voltage on V_{DD} has risen to a point where the MCU can operate properly by the time the 4,064 bus cycles have elapsed. If there is doubt, the external $\overline{\text{RESET}}$ pin should remain low until the voltage on V_{DD} has reached the specified minimum operating voltage.

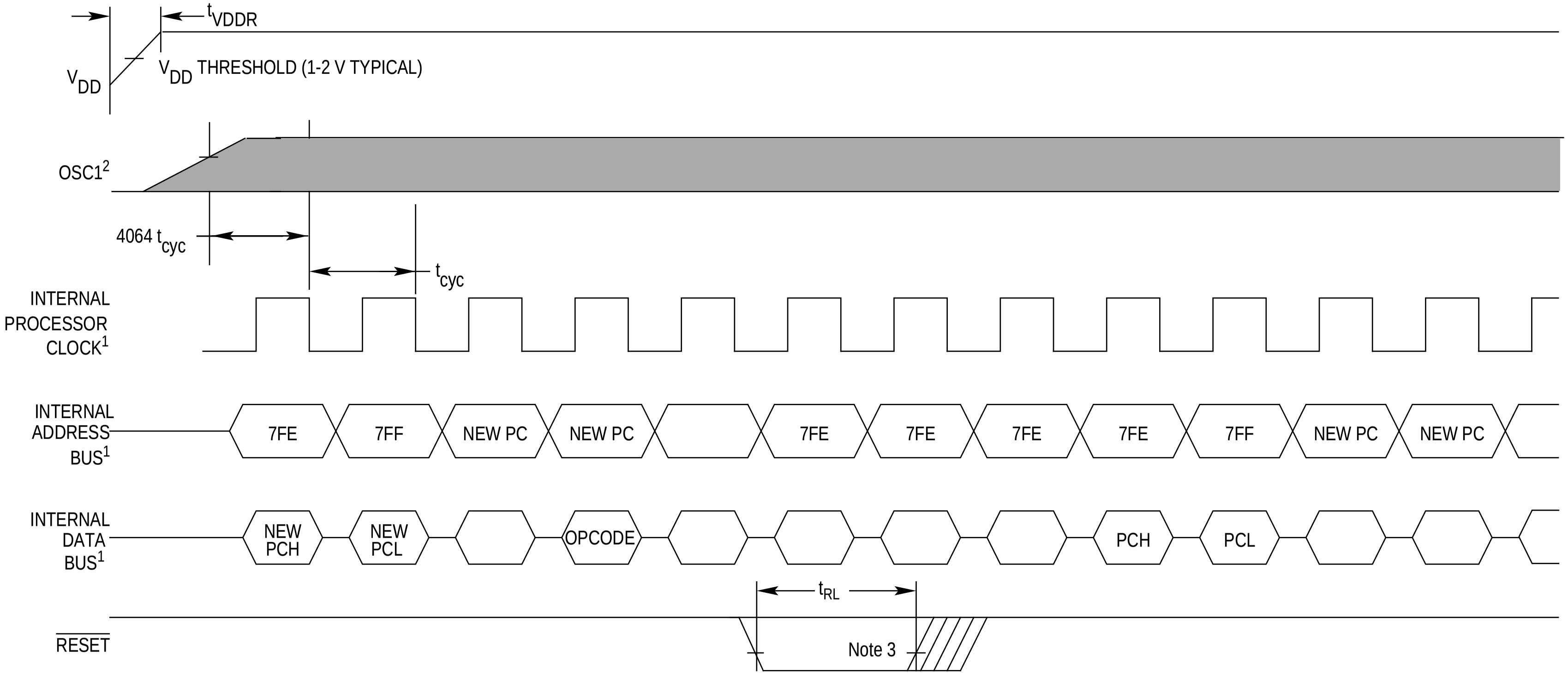
5.3.2 Computer Operating Properly Reset (COPR)

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 enabled by the COPEN MOR bit. Refer to **8.4 COP Watchdog** for more information on this time-out feature.

The COPR will generate the RST signal which will reset the CPU and other peripherals. If the voltage at the $\overline{\text{IRQ}}$/V$_{PP}$ 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.

5.3.3 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), internal test ROM ($07EE–$07EF), or EPROM ($0300–$07CF, $07F8–$07FF). The ILADR will generate the RST signal which will reset the CPU and other peripherals. If the voltage at the $\overline{\text{IRQ}}$/V$_{PP}$ pin is in the normal operating range (V$_{SS}$ to V$_{DD}$), the MCU will enter user mode when the illegal address reset signal ends. 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.



NOTES:

1. Internal timing signal and bus information not available externally.
2. OSC1 line is not meant to represent frequency. It is only used to represent time.
3. The next rising edge of the PH2 clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence.

Figure 5-2. Power-On Reset and External Timing Diagram

SECTION 6
LOW POWER MODES

6.1 Introduction

In user mode the MC68HC705J1A 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 MOR bit 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.

6.2 STOP Instruction

The STOP instruction can result in one of two modes of operation depending on the SWAIT bit in the MOR. 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.

6.2.1 Stop Mode

Execution of the STOP instruction while enabled (as chosen by the MOR) 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 TSCR 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 status/control register so that the IRQ external interrupt is enabled. All other registers, including the other bits in the TSCR, and memory remain unaltered. All input/output lines remain unchanged.

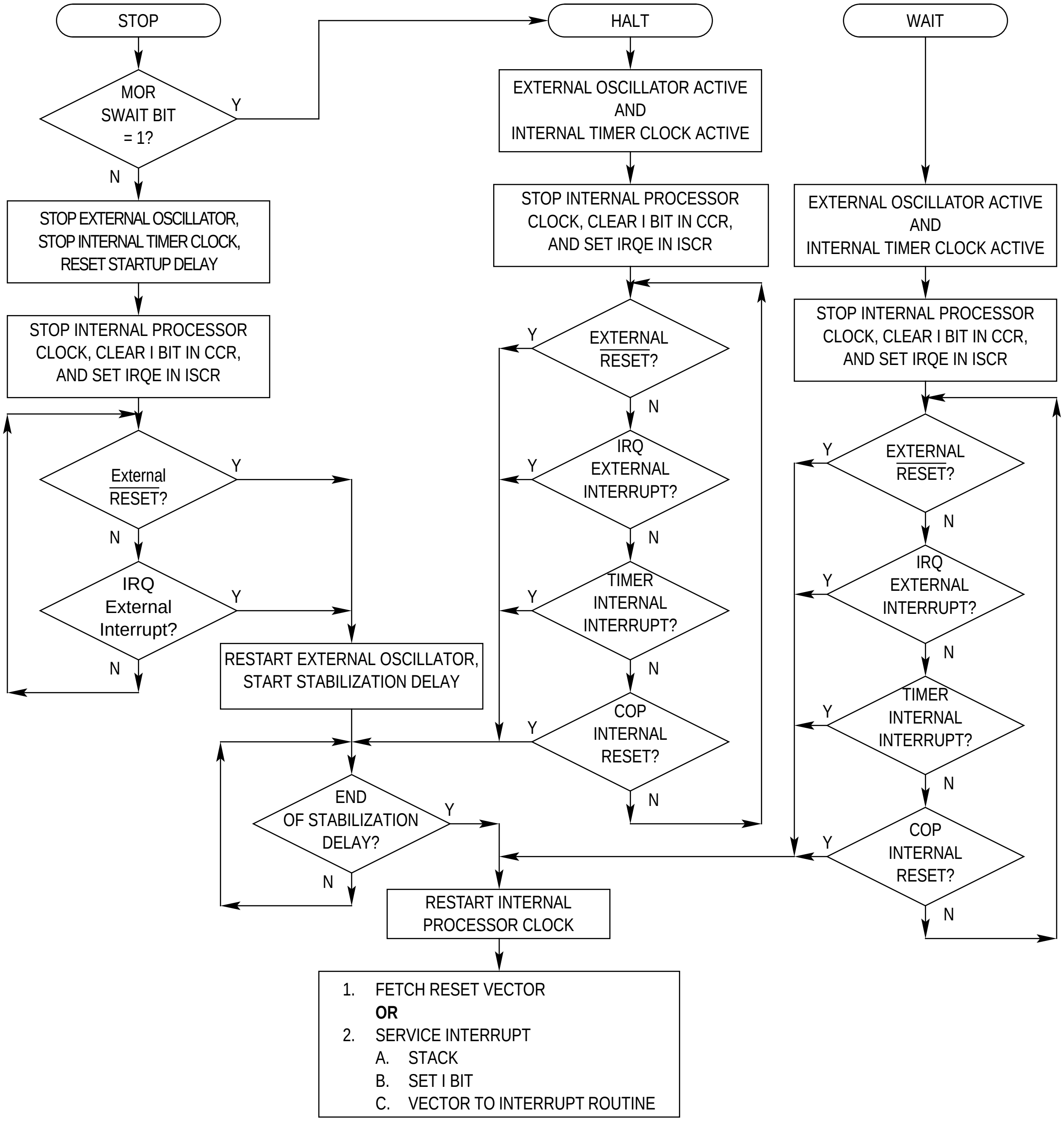

STOP
HALT
WAIT
MOR SWAIT BIT = 1?
Y
N
EXTERNAL OSCILLATOR ACTIVE AND INTERNAL TIMER CLOCK ACTIVE
STOP EXTERNAL OSCILLATOR, STOP INTERNAL TIMER CLOCK, RESET STARTUP DELAY
STOP INTERNAL PROCESSOR CLOCK, CLEAR I BIT IN CCR, AND SET IRQE IN ISCR
External RESET?
IRQ External Interrupt?
EXTERNAL RESET?
IRQ EXTERNAL INTERRUPT?
TIMER INTERNAL INTERRUPT?
COP INTERNAL RESET?
RESTART EXTERNAL OSCILLATOR, START STABILIZATION DELAY
END OF STABILIZATION DELAY?
RESTART INTERNAL PROCESSOR CLOCK
1. FETCH RESET VECTOR
OR
2. SERVICE INTERRUPT
A. STACK
B. SET I BIT
C. VECTOR TO INTERRUPT ROUTINE

Figure 6-1. Stop/Halt/Wait Flowcharts

The MCU can be brought out of the stop mode only by an IRQ external interrupt or an externally generated reset. When exiting the stop mode the internal oscillator will resume after a 4,064 (or 16, depending upon the MOR[SOSCD] bit) internal processor clock cycle oscillator stabilization delay.

NOTE

Execution of the STOP instruction with the SWAIT MOR bit cleared 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 setting the appropriate MOR bit. See **6.5 COP Watchdog Timer Considerations** for more details.

6.2.2 Halt Mode

Execution of the STOP instruction while SWAIT in the MOR is enabled 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 status/control 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, or external reset occurs. When exiting the halt mode the internal processor clock will resume after a delay of one to 4,064 (or 16, depending upon the MOR[SOSCD] bit) 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.

6.3 WAIT Instruction

The WAIT instruction places the MCU in a low-power mode, which consumes less power than the run 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 status/control 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 reset occurs.

6.4 Data-Retention Mode

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

6.5 COP Watchdog Timer Considerations

If the COP watchdog timer is selected by the COP MOR bit, 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 operation if the COP watchdog timer is enabled (for additional information see **9.4.2 Mask Option Register (MOR)**).

If the COP watchdog timer is selected by the COP MOR bit, 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 6-1.

Table 6-1. COP Watchdog Timer Recommendations

IF the following conditions exist:		**THEN the COP Watchdog Timer should be as follows:**
STOP Instruction	**WAIT Time**	
Converted to HALT by MOR bit	WAIT Time **Less Than** COP Time-Out	Enable or Disable COP by MOR Bit
Converted to HALT by MOR bit	WAIT Time **More Than** COP Time-Out	Disable COP by MOR Bit
Acts as STOP	Any Length WAIT Time	Disable COP by MOR Bit

SECTION 7
INPUT/OUTPUT PORTS

7.1 Introduction

In the user mode, there are 14 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 (DDRs). Also, if enabled by a single MOR bit (PDI) all port A and port B I/O pins may have individual software programmable pulldown devices. Some port A (PA4-PA7) pins also have the additional properties of sinking higher current and the others (PA0-PA3) may function as additional IRQ interrupt input sources.

7.2 Port A

Port A is an 8-bit bidirectional port which shares four of its pins with the IRQ interrupt system as shown in Figure 7-1. Each port A pin is controlled by the corresponding bits in a data direction register, a data register, and a pulldown 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 pulldown register (PDRA) is located at address $0010. Reset clears the DDRA and the PDRA. The port A data register is unaffected by reset.

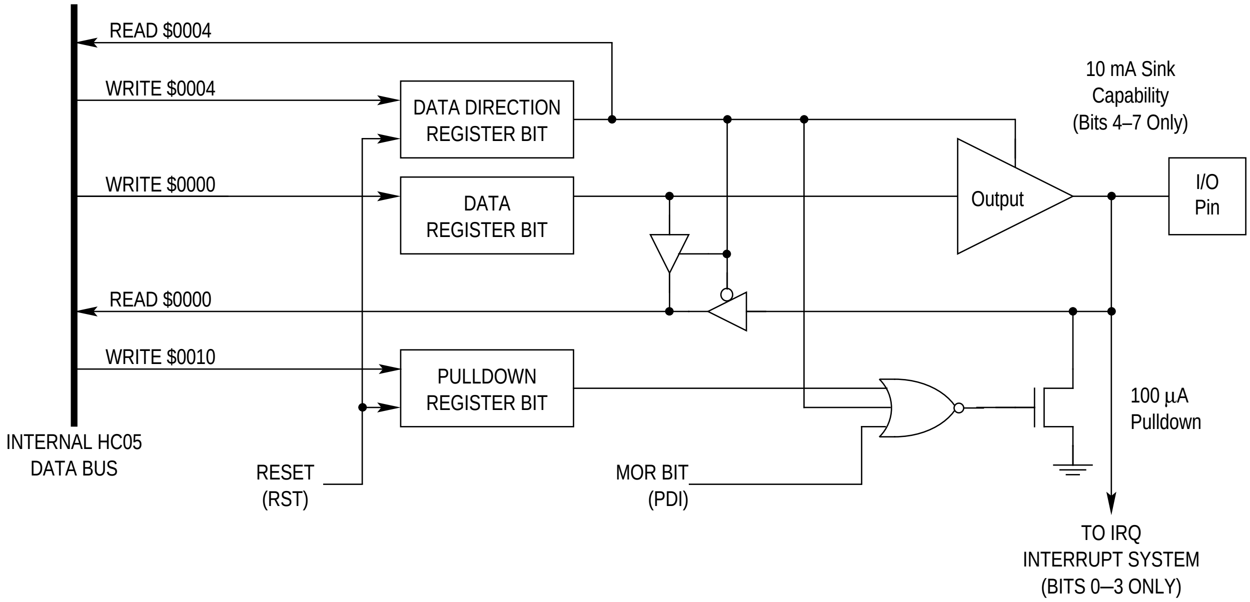


Figure 7-1. Port A I/O Circuitry

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.

7.2.3 Port A Pulldown Register

All port A I/O pins may have software programmable pulldown devices enabled or disabled by the applicable MOR bit (PDI). If PDI is cleared, the pulldown is activated whenever the corresponding bit in the PDRA is clear. If the PDRA bit is set or the MOR PDI bit is set, the pulldown will be disabled. A pulldown on an I/O pin is activated only if the I/O pin is programmed as an input.

The PDRA is a write-only register. Any reads of location $0010 will return undefined results. Since reset clears both the DDRA and the PDRA, all pins will initialize as inputs with the pulldown devices active (if enabled by PDI MOR bit being cleared).

NOTE

Read-modify-write instructions should NOT be executed on registers with write-only bits.

7.2.4 Port A LED Drive Capability

The outputs for the upper four bits of port A (PA4, PA5, PA6, and PA7) are capable of sinking approximately 10 mA of current to V_{SS}.

7.2.5 Port A I/O Pin Interrupts

The inputs for the lower four bits of port A (PA0, PA1, PA2, and PA3) may be connected to the IRQ input of the CPU if enabled by the MOR bit, PIRQ. When connected as an alternate source of an IRQ interrupt, the port A input pins will behave the same as the $\overline{\text{IRQ}}/V_{PP}$ pin itself, except that their active state is a logic one or a rising edge. The $\overline{IRQ}/V_{PP}$ pin has an active state that is a logic zero or a falling edge.

If MOR bits for both level sensitivity and port A interrupts are programmed to a logic one, 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 port A IRQ input or the IRQ input remains at the active level, the port A IRQ inputs and the IRQ input are effectively ignored.

NOTE

The BIH and BIL instructions will apply only to the level on the $\overline{IRQ}/V_{PP}$ 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.

7.3 Port B

Port B is a 6-bit bidirectional port which functions as shown in Figure 7-2. Each port B pin is controlled by the corresponding bits in a data direction register, a data register, and a pulldown 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 pulldown register (PDRB) is located at address $0011. Reset clears the DDRB and the PDRB. The port B data register is unaffected by reset.

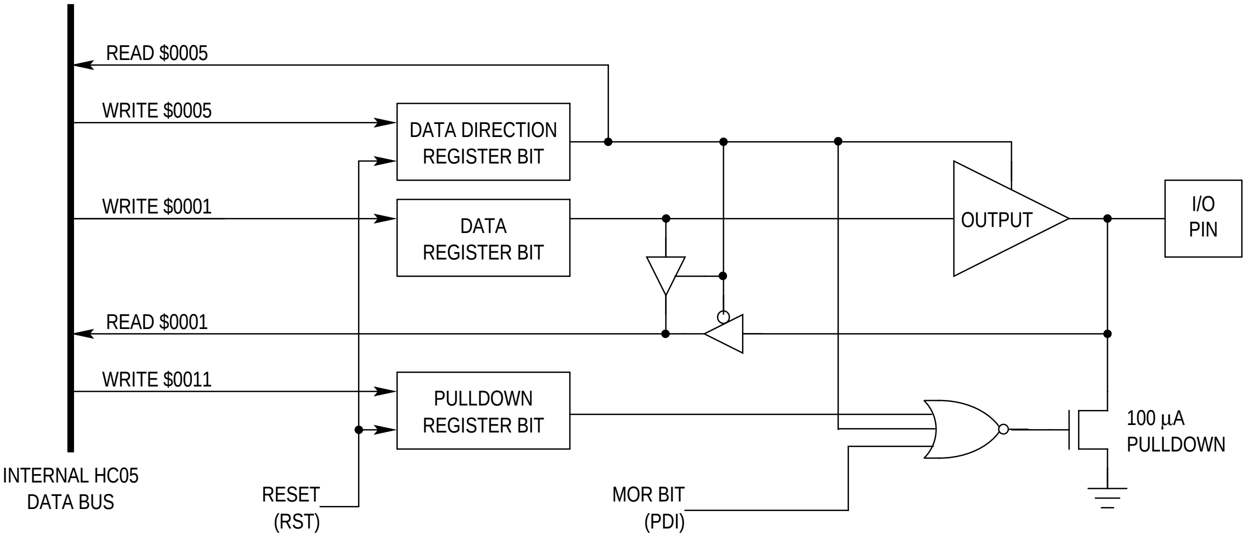


Figure 7-2. Port B I/O Circuitry

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.

7.3.3 Port B Pulldown Register

All port B I/O pins may have software programmable pulldown devices enabled by PDI in the MOR. If the pulldown inhibit MOR bit is cleared, the pulldown is activated whenever the corresponding bit in the PDRB is clear. A pulldown on an I/O pin is activated only if the I/O pin is programmed as an input.

The PDRB is a write-only register. Any reads of location $0011 will return undefined results. Since reset clears both the DDRB and the PDRB, all pins will initialize as inputs with the pulldown devices active (if MOR bit, PDI is cleared).

NOTE

Read-modify-write instructions should NOT be executed on registers with write-only bits.

7.4 I/O Port Programming

All I/O pins can be programmed as inputs or outputs, with or without pulldown 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 through DDRB5 and DDRA0 through 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 pulldown inhibit MOR bit is cleared, all pins will initially power up with their software programmable pulldowns 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.

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.

If the corresponding bit in the pulldown register is clear (and the PDI MOR bit is cleared) the input pin will also have an activated pulldown device. Since the pulldown 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 pre-conditioned to the desired state before changing the corresponding DDR bit from a zero to a one.

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 pulldown devices if selected by the appropriate MOR bit.

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

DDRA	I/O Pin Mode	Accesses to PDRA at $0010		Accesses to DDRA @ $0004	Accesses to Data Register @ $0000	
		Read	Write	Read/Write	Read	Write
0	In, Hi-Z	U	PDRA0–PDRA7	DDRA0–DDRA7	I/O Pin	*
1	Out	U	PDRA0–PDRA7	DDRA0–DDRA7	PA0–PA7	PA0–PA7

Note: U is undefined
*Does not affect input, but stored to data register

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

DDRA	I/O Pin Mode	Accesses to PDRB at $0011		Accesses to DDRB @ $0005	Accesses to Data Register @ $0001	
		Read	Write	Read/Write	Read	Write
0	In, Hi-Z	U	PDRB0–PDRB5	DDRB0–DDRB5	I/O Pin	*
1	Out	U	PDRB0–PDRB5	DDRB0–DDRB5	PA0–PA5	PA0–PA5

Note: U is undefined
*Does not affect input, but stored to data register

SECTION 8
MULTIFUNCTION TIMER

8.1 Introduction

This section describes the operation of the multifunction timer and the COP watchdog. Figure 8-0? shows the organization of the timer subsystem.

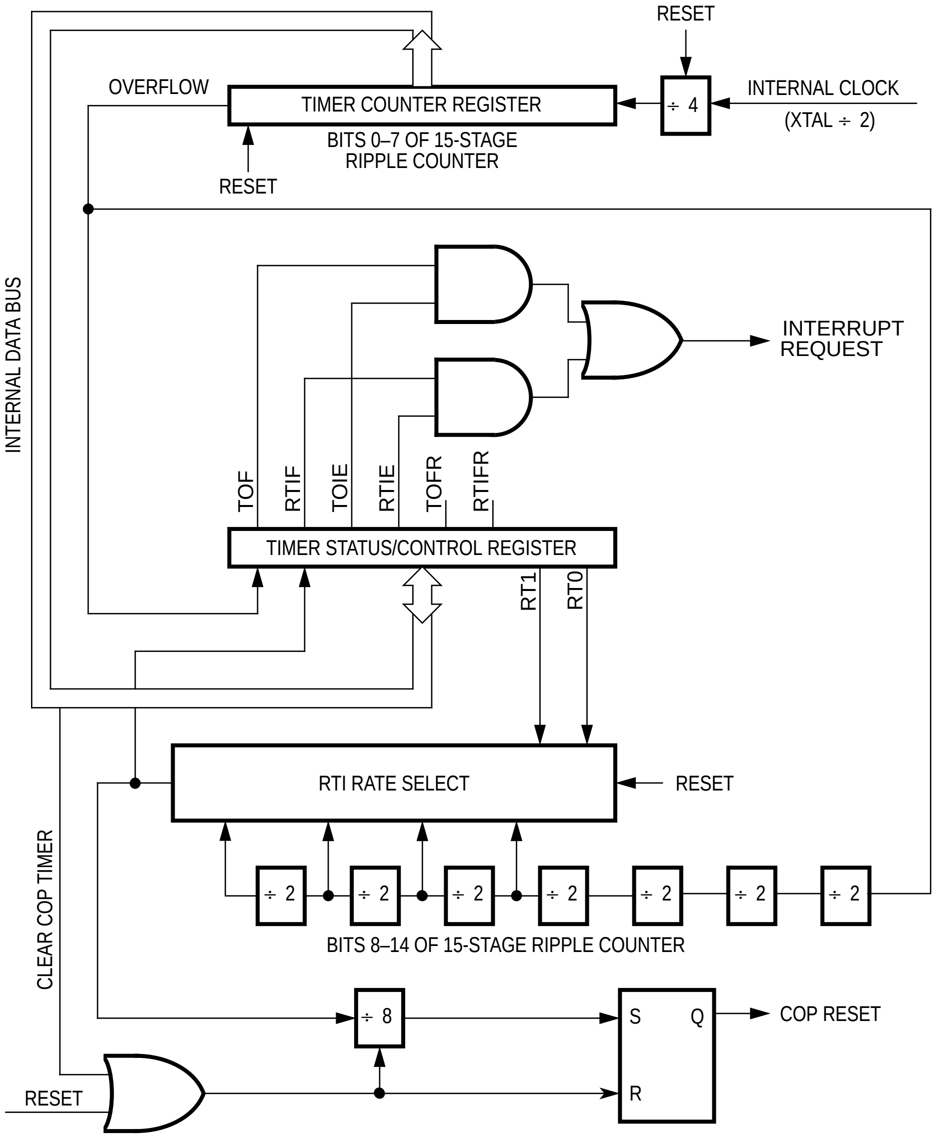


Figure 8-1. Multifunction Timer Block Diagram

8.2 Timer Status and Control Register (TSCR)

The read/write timer status and control register contains the following bits:

- Timer Interrupt Enable Bits
- Timer Interrupt Flags
- Timer Interrupt Flag Reset Bits
- Timer Interrupt Rate Select Bits

		Bit 7	6	5	4	3	2	1	Bit 0
TSCR $08	Read:	TOR	RTIF	TOIE	RTIE	0	0	RT1	RT0
	Write:	░	░			TOFR	RTIFR		
Reset:		0	0	0	0	0	0	1	1

░ = Unimplemented

Figure 8-2. Timer Status/Control Register

TOF — Timer Overflow Flag

This read-only flag becomes set when the first eight stages of the counter roll over from $FF to $00. TOF generates a timer overflow interrupt request if TOIE is also set. Clear TOF by writing a logic one to the TOFR bit. Writing to TOF has no effect. Reset clears TOF.

RTIF — Real-Time Interrupt Flag

This read-only flag becomes set when the selected RTI output becomes active. RTIF generates a real-time interrupt request if RTIE is also set. Clear RTIF by writing a logic one to the RTIFR bit. Writing to RTIF has no effect. Reset clears RTIF.

8

TOIE — Timer Overflow Interrupt Enable

This read/write bit enables timer overflow interrupts.
1 = Timer overflow interrupts enabled
0 = Timer overflow interrupts disabled

RTIE — Real-Time Interrupt Enable

This read/write bit enables real-time interrupts
1 = Real-time interrupts enabled
0 = Real-time interrupts disabled

TOFR — Timer Overflow Flag Reset

Writing a logic one to this write-only bit clears the TOF bit. TOFR always reads as logic zero. Reset clears TOFR.

RTIFR — Real-Time Interrupt Flag Reset

Writing a logic one to this write-only bit clears the RTIF bit. RTIFR always reads as logic zero. Reset clears RTIFR.

RT1 and RT0 — Real-Time Interrupt Select Bits 1 and 0

These read/write bits select one of four real-time interrupt rates, as shown in Table 8-1. Because the selected RTI output drives the COP watchdog, changing the real-time interrupt rate also changes the counting rate of the COP watchdog. Reset sets RT1 and RT0.

NOTE

Changing RT1 and RT0 when a COP timeout is imminent or uncertain may cause a real-time interrupt request to be missed or an additional real-time interrupt request to be generated. Clear the COP timer just before changing RT1 and RT0.

Table 8-1. Real-Time Interrupt Rate Selection

RT1:RT0	RTI Rate	RTI Period (f_{OP} = 2 MHz)	COP Timeout Period (–0/+1 RTI Period)	Minimum COP Timeout Period (f_{OP} = 2 MHz)
0 0	$f_{OP} \div 2^{14}$	8.2 ms	7 × RTI Period	57.3 ms
0 1	$f_{OP} \div 2^{15}$	16.4 ms	7 × RTI Period	114.6 ms
1 0	$f_{OP} \div 2^{16}$	32.8 ms	7 × RTI Period	229.3 ms
1 1	$f_{OP} \div 2^{17}$	65.5 ms	7 × RTI Period	458.7 ms

8.3 Timer Counter Register (TCNTR)

A 15-stage ripple counter is the core of the timer. The value of the first eight stages is readable at any time from the read-only timer counter register shown in Table 8-1?.

		Bit 7	6	5	4	3	2	1	Bit 0
TCR $09	Read:	TMR7	TMR6	TMR5	TMR4	TMR3	TMR2	TMR1	TMR0
	Write:	▨	▨	▨	▨	▨	▨	▨	▨
Reset:		0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 8-3. Timer Counter Register

Power on clears the entire counter chain and begins clocking the counter. After 4,064 (or 16, if the MOR[SOSCD] bit is set) cycles, the power-on reset circuit is released, clearing the counter again and allowing the MCU to come out of reset.

A timer overflow function at the eighth counter stage allows a timer interrupt every 1,024 internal clock cycles.

8.4 COP Watchdog

Four counter stages at the end of the timer make up the MOR selectable computer operating properly (COP) watchdog. See Figure 8-1?. The COP watchdog is a software error detection system that automatically times out and resets the MCU if not cleared periodically by a program sequence. Writing a logic zero to bit 0 of the COP register clears the COP watchdog and prevents a COP reset.

		Bit 7	**6**	5	4	3	2	1	Bit 0
COPR $07F0	Read:	▨	▨	▨	▨	▨	▨	▨	▨
	Write:	▨	▨	▨	▨	▨	▨	▨	COPC
	Reset:	U	U	U	U	U	U	U	0

▨ = Unimplemented U = Undetermined

Figure 8-4. COP Watchdog Timer Location

COPC — COP Clear

This write-only bit resets the COP watchdog. Reading address $07F0 returns undefined results.

NOTE

The STOP instruction turns off the COP watchdog. In applications that depend on the COP watchdog, the STOP instruction can be disabled via the SWAIT bit in the MOR.

8

SECTION 9
USER EPROM

9.1 Introduction

The user EPROM consists of 1232 bytes of user EPROM from $0300 to $07CF and 8 bytes of user vectors EPROM from $07F8 to $07FF.

The mask option register (MOR) is an additional EPROM byte provided to control various functions as described in **9.4.2 Mask Option Register (MOR)**.

9.2 EPROM Erasing

The MC68HC705J1A can be erased by the exposure to a high-intensity ultraviolet (UV) light with a wavelength of 2537 angstroms. The recommended dose (UV intensity multiplied by exposure time) is 15 Ws/cm^2. UV lamps without shortwave filters should be used, and the EPROM device should be positioned about one inch from the UV lamp.

NOTE

Unlike many commercial EPROMs, an erased EPROM byte will read as $00. All unused locations should be programmed as logic zeros.

9.3 EPROM Programming

The M68HC705JICS board is available from Motorola for programming the on-chip EPROM and the MOR.

9.4 EPROM Registers

Two registers are associated with the EPROM: the EPROM programming register (EPROG) and the mask option register (MOR). The EPROG register controls the actual programming of the EPROM bytes and the MOR. The MOR register controls six mask options found on the ROM version of this MCU (MC68HC05J1A) with an additional option for EPROM array security and for short oscillator recovery delay.

9.4.1 EPROM Programming Register (EPROG)

This register is used to program the EPROM array and the MOR. Only the ELAT, MPGM, and EPGM bits are available in user mode. Figure 9-1 shows the location of each bit in the EPROG register and the state of these bits coming out of reset. All the bits in the EPROG register are cleared by reset.

		Bit 7	6	5	4	3	2	1	Bit 0
EPROG $18	Read:	0	0	0	0	0	ELAT	MPGM	EPGM
	Write:	▨	R	R	R	R			
	Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented   R = Reserved for factory test

Figure 9-1. EPROM Programming Register

EPGM — EPROM Program Control

The EPGM bit cannot be set unless the ELAT bit is already set. Whenever the ELAT bit is cleared the EPGM bit is also cleared. Both the EPGM and the ELAT bit cannot be set using the same write instruction. Any attempt to set both the EPGM and ELAT bit on the same write instruction cycle will result in the ELAT bit being set and the EPGM bit being cleared. The EPGM bit is a read-write bit and can be read at any time. The EPGM bit is cleared by reset.

1 = Programming voltage ($\overline{IRQ}/V_{PP}$ pin) is applied to the EPROM array
0 = Programming voltage ($\overline{IRQ}/V_{PP}$ pin) is not applied to the EPROM array

MPGM — MOR Program Control

The MPGM bit is cleared by reset. See **Section 9.5** for MOR programming information.

1 = Programming voltage ($\overline{IRQ}/V_{PP}$ pin) is applied to the MOR
0 = Programming voltage ($\overline{IRQ}/V_{PP}$ pin) is not applied to the MOR

ELAT — EPROM Latch Control

Whenever the ELAT bit is cleared the EPGM bit is also cleared. Both the EPGM and the ELAT bits cannot be set using the same write instruction. Any attempt to set both the EPGM and ELAT bit on the same write instruction cycle will result in the ELAT bit being set and the EPGM bit being cleared. The ELAT bit is a read-write bit and can be read at any time. The ELAT bit is cleared by reset.

1 = EPROM address and data bus configured for programming to the array. The address and data bus are latched in the EPROM array when a subsequent write to the array is made. Data in the EPROM array cannot be read
0 = EPROM address and data bus configured for normal reading of data from the array

To program a byte of EPROM, manipulate the EPROG register as follows:

1. Set the ELAT bit in the EPROG register
2. Write the desired data to the desired EPROM address
3. Set the EPGM bit in the EPROG register for the specified programming time (t_{EPGM})
4. Clear the ELAT and EPGM bits in the EPROG register

9.4.2 Mask Option Register (MOR)

The mask option register (MOR) is a byte of EPROM used to enable or disable six of the features controlled by mask options on the MC68HC05J1A (a ROM version of the MC68HC705J1A) in addition to EPROM security and short oscillator delay options. This structure does not contain any latches which may become corrupted by erratic supply voltages or CPU operation. The programmable options on the MC68HC705J1A are:

1. COP Watchdog Timer (Enable or Disable)
2. IRQ Triggering (Edge- or Edge- and Level-Sensitive)
3. Port A IRQ Interrupts (Enable or Disable)
4. Port Pulldown Resistors (Enable or Disable)
5. STOP Instruction Mode (Stop Mode or Halt Mode)
6. Crystal Oscillator Parallel Resistor (Enable or Disable)
7. EPROM Security (Enable or Disable)
8. Short Oscillator Delay Counter (Enable or Disable)

		Bit 7	6	5	4	3	2	1	Bit 0
MOR $07F1	Read: Write:	SOSCD	EPMSEC	OSCRES	SWAIT	SWPDI	PIRQ	LEVEL	COPEN
	Reset:	0	0	0	0	0	0	0	0

Figure 9-2. **Mask Option Register (EPROM)**

COPEN — COP Watchdog Enable

1 = COP watchdog enabled. The COP will reset the MCU if the time-out period is reached before the COP watchdog timer is cleared by the application software *and* the voltage applied to the $\overline{IRQ}/V_{PP}$ pin is between V_{SS} and V_{DD}. Clearing the COP bit will disable the COP watchdog timer regardless of the voltage applied to the $\overline{IRQ}/V_{PP}$ pin

0 = COP Watchdog disabled

LEVEL — IRQ Edge Sensitivity

1 = $\overline{\text{IRQ}}/V_{PP}$ pin sensitive to both the falling edge and the logic low level of the input signal on the $\overline{\text{IRQ}}/V_{PP}$ pin

0 = $\overline{\text{IRQ}}/V_{PP}$ pin will only be sensitive to the falling edge of the signal applied to the $\overline{\text{IRQ}}/V_{PP}$ pin

PIRQ — Port A IRQ Interrupt Function

1 = Port A pins PA0-PA3 can function as IRQ interrupts

0 = Port A IRQ interrupts disabled

SWPDI — Port Pulldown Inhibit

1 = Pulldowns inhibited regardless of the state of the software selectable pulldown registers

0 = Port pins can have software selectable input pulldowns

SWAIT — STOP Instruction Mode

1 = The STOP instruction will initiate halt mode

0 = The STOP instruction will intiate stop mode

OSCRES — Oscillator Parallel Resistor

1 = Optional 2 MΩ parallel resistor in the oscillator circuit is enabled

0 = Optional oscillator resistor is disabled

NOTE

The optional oscillator resistor is NOT recommended for devices that use the RC oscillator. For such devices, this bit should be programmed to a logic zero.

EPMSEC — EPROM Security Enable

1 = Access to the EPROM array in non-user modes is denied

0 = Access to the EPROM array in non-user modes is possible

SOSCD — Short Oscillator Delay Enable

This bit is provided to control the oscillator stabilization counter.

1 = POR/STOP recovery delay is 16 CPU cycles

0 = POR/STOP recovery delay is 4,064 CPU cycles

CAUTION

The 16-cycle delay option will work properly in devices with the RC oscillator. Check crystal/ceramic resonator specifications carefully before using this option with a crystal or ceramic resonator.

NOTE

This option is NOT available as a mask option on the MC68HC05J1A (ROM version of this device).

9.5 MOR Programming

To program any bits in the MOR, the desired bit states must be written to the MOR and then the MPGM bit in the EPROG must be used. The following sequence will program the MOR:

1. Write the desired data to the MOR.
2. Set the MPGM bit in the EPROG.
3. Wait for the programming time (t_{MPGM}).
4. Clear the MPGM bit in the EPROG.

Once the MOR bits have been programmed, some of the options may experience glitches in operation following removal of the programming voltage. It is recommended that the part be reset before trying to verify the contents of the user EPROM or the MOR itself.

SECTION 10
INSTRUCTION SET

10.1 Introduction

This section describes the M68HC705J1A addressing modes and instruction types.

10.2 Addressing Modes

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

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

10.2.1 Inherent

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

10

10.2.2 Immediate

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

10.2.3 Direct

Direct instructions can access any of the first 256 memory addresses with two bytes. The first byte is the opcode, and the second is the low byte of the operand address. In direct addressing, the CPU automatically uses $00 as the high byte of the operand address. BRSET and BRCLR are three-byte instructions that use direct addressing to access the operand and relative addressing to specify a branch destination.

10.2.4 Extended

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

When using the Motorola assembler, the programmer does not need to specify whether an instruction is direct or extended. The assembler automatically selects the shortest form of the instruction.

10.2.5 Indexed, No Offset

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

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

10.2.6 Indexed, 8-Bit Offset

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

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

10.2.7 Indexed, 16-Bit Offset

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

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

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

10.2.8 Relative

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

When using the Motorola assembler, the programmer does not need to calculate the offset, because the assembler determines the proper offset and verifies that it is within the span of the branch.

10.3 Instruction Types

The MCU instructions fall into the following five categories:

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

10.3.1 Register/Memory Instructions

Most of these instructions use two operands. One operand is in either the accumulator or the index register. The CPU finds the other operand in memory. Table 10-1 lists the register/memory instructions.

Table 10-1. Register/Memory Instructions

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

10.3.2 Read-Modify-Write Instructions

These instructions read a memory location or a register, modify its contents, and write the modified value back to the memory location or to the register. The test for negative or zero instruction (TST) is an exception to the read-modify-write sequence because it does not write a replacement value. Table 10-2 lists the read-modify-write instructions.

Table 10-2. Read-Modify-Write Instructions

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

10.3.3 Jump/Branch Instructions

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

Bit test and branch instructions cause a branch based on the state of any readable bit in the first 256 memory locations. These three-byte instructions use a combination of direct addressing and relative addressing. The direct address of the byte to be tested is in the byte following the opcode. The third byte is the signed offset byte. The CPU finds the conditional branch destination by adding the third byte to the program counter if the specified bit tests true. The bit to be tested and its condition (set or clear) is part of the opcode. The span of branching is from –128 to +127 from the address of the next location after the branch instruction. The CPU also transfers the tested bit to the carry/borrow bit of the condition code register. Table 10-3 lists the jump and branch instructions.

Table 10-3. Jump and Branch Instructions

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

10.3.4 Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory. Port registers, port data direction registers, timer registers, and on-chip RAM locations are in the first 256 bytes of memory. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations. Bit manipulation instructions use direct addressing. Table 10-4 lists these instructions.

Table 10-4. Bit Manipulation Instructions

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

10.3.5 Control Instructions

These register reference instructions control CPU operation during program execution. Control instructions, listed in Table 10-5, use inherent addressing.

Table 10-5. Control Instructions

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

10.4 Instruction Set Summary

Table 10-6 is an alphabetical list of all M68HC05 instructions and shows the effect of each instruction on the condition code register.

Table 10-6. Instruction Set Summary

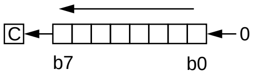
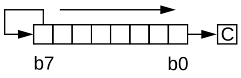
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	×	—	×	×	×	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr* ADD *opr* ADD *opr*,X ADD *opr*,X ADD ,X	Add without Carry	A ← (A) + (M)	×	—	×	×	×	IMM DIR EXT IX2 IX1 IX	AB BB CB DB EB FB	ii dd hh ll ee ff ff	2 3 4 5 4 3
AND #*opr* AND *opr* AN*D opr* AND *opr*,X AND *opr*,X AND ,X	Logical AND	A ← (A) ∧ (M)	—	—	×	×	—	IMM DIR EXT IX2 IX1 IX	A4 B4 C4 D4 E4 F4	ii dd hh ll ee ff ff	2 3 4 5 4 3
ASL *opr* ASLA ASLX ASL *opr*,X ASL ,X	Arithmetic Shift Left (Same as LSL)		—	—	×	×	×	DIR INH INH IX1 IX	38 48 58 68 78	dd ff	5 3 3 6 5
ASR *opr* ASRA ASRX ASR *opr*,X ASR ,X	Arithmetic Shift Right		—	—	×	×	×	DIR INH INH IX1 IX	37 47 57 67 77	dd ff	5 3 3 6 5
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BCLR *n opr*	Clear Bit n	Mn ← 0	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + 2 + *rel* ? Z = 1	—	—	—	—	—	REL	27	rr	3
BHCC *rel*	Branch if Half-Carry Bit Clear	PC ← (PC) + 2 + *rel* ? H = 0	—	—	—	—	—	REL	28	rr	3

Table 10-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BHCS *rel*	Branch if Half-Carry Bit Set	PC ← (PC) + 2 + *rel* ? H = 1	—	—	—	—	—	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? C ∨ Z = 0	—	—	—	—	—	REL	22	rr	3
BHS *rel*	Branch if Higher or Same	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BIH *rel*	Branch if $\overline{\text{IRQ}}$ Pin High	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 1	—	—	—	—	—	REL	2F	rr	3
BIL *rel*	Branch if $\overline{\text{IRQ}}$ Pin Low	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 0	—	—	—	—	—	REL	2E	rr	3
BIT #*opr* BIT *opr* BIT *opr* BIT *opr*,X BIT *opr*,X BIT ,X	Bit Test Accumulator with Memory Byte	(A) ∧ (M)	—	—	×	×	—	IMM DIR EXT IX2 IX1 IX	A5 B5 C5 D5 E5 F5	ii dd hh ll ee ff ff p	2 3 4 5 4 3
BLO *rel*	Branch if Lower (Same as BCS)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	PC ← (PC) + 2 + *rel* ? C ∨ Z = 1	—	—	—	—	—	REL	23	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	PC ← (PC) + 2 + *rel* ? I = 0	—	—	—	—	—	REL	2C	rr	3
BMI *rel*	Branch if Minus	PC ← (PC) + 2 + *rel* ? N = 1	—	—	—	—	—	REL	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	PC ← (PC) + 2 + *rel* ? I = 1	—	—	—	—	—	REL	2D	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + 2 + *rel* ? Z = 0	—	—	—	—	—	REL	26	rr	3
BPL *rel*	Branch if Plus	PC ← (PC) + 2 + *rel* ? N = 0	—	—	—	—	—	REL	2A	rr	3
BRA *rel*	Branch Always	PC ← (PC) + 2 + *rel* ? 1 = 1	—	—	—	—	—	REL	20	rr	3
BRCLR *n opr rel*	Branch if bit n clear	PC ← (PC) + 2 + *rel* ? Mn = 0	—	—	—	—	×	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRSET *n opr rel*	Branch if Bit n Set	PC ← (PC) + 2 + *rel* ? Mn = 1	—	—	—	—	×	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN *rel*	Branch Never	PC ← (PC) + 2 + *rel* ? 1 = 0	—	—	—	—	—	REL	21	rr	3

Table 10-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BSET *n opr*	Set Bit n	Mn ← 1	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BSR *rel*	Branch to Subroutine	PC ← (PC) + 2; push (PCL) SP ← (SP) – 1; push (PCH) SP ← (SP) – 1 PC ← (PC) + *rel*	—	—	—	—	—	REL	AD	rr	6
CLC	Clear Carry Bit	C ← 0	—	—	—	—	0	INH	98		2
CLI	Clear Interrupt Mask	I ← 0	—	0	—	—	—	INH	9A		2
CLR *opr* CLRA CLRX CLR *opr*,X CLR ,X	Clear Byte	M ← $00 A ← $00 X ← $00 M ← $00 M ← $00	—	—	0	1	—	DIR INH INH IX1 IX	3F 4F 5F 6F 7F	dd ff 	5 3 3 6 5
CMP #*opr* CMP *opr* CMP *opr* CMP *opr*,X CMP *opr*,X CMP ,X	Compare Accumulator with Memory Byte	(A) – (M)	—	—	×	×	×	IMM DIR EXT IX2 IX1 IX	A1 B1 C1 D1 E1 F1	ii dd hh ll ee ff ff 	2 3 4 5 4 3
COM *opr* COMA COMX COM *opr*,X COM ,X	Complement Byte (One's Complement)	$M \leftarrow (\overline{M}) = \$FF - (M)$ $A \leftarrow (\overline{A}) = \$FF - (M)$ $X \leftarrow (\overline{X}) = \$FF - (M)$ $M \leftarrow (\overline{M}) = \$FF - (M)$ $M \leftarrow (\overline{M}) = \$FF - (M)$	—	—	×	×	1	DIR INH INH IX1 IX	33 43 53 63 73	dd ff 	5 3 3 6 5
CPX #*opr* CPX *opr* CPX *opr* CPX *opr*,X CPX *opr*,X CPX ,X	Compare Index Register with Memory Byte	(X) – (M)	—	—	×	×	1	IMM DIR EXT IX2 IX1 IX	A3 B3 C3 D3 E3 F3	ii dd hh ll ee ff ff 	2 3 4 5 4 3
DEC *opr* DECA DECX DEC *opr*,X DEC ,X	Decrement Byte	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1	—	—	×	×	—	DIR INH INH IX1 IX	3A 4A 5A 6A 7A	dd ff 	5 3 3 6 5
EOR #*opr* EOR *opr* EOR *opr* EOR *opr*,X EOR *opr*,X EOR ,X	EXCLUSIVE OR Accumulator with Memory Byte	A ← (A) ⊕ (M)	—	—	×	×	—	IMM DIR EXT IX2 IX1 IX	A8 B8 C8 D8 E8 F8	ii dd hh ll ee ff ff 	2 3 4 5 4 3

Table 10-6. Instruction Set Summary (Continued)

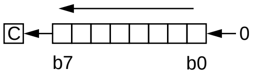
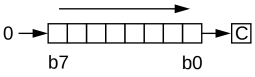
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
INC *opr* INCA INCX INC *opr*,X INC ,X	Increment Byte	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1	—	—	×	×	—	DIR INH INH IX1 IX	3C 4C 5C 6C 7C	dd ff	5 3 3 6 5
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Unconditional Jump	PC ← Jump Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BC C C D C EC FC	dd hh ll ee ff ff	2 3 4 3 2
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Conditional Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BD C D D D ED FD	dd hh ll ee ff ff	5 6 7 6 5
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X	Load Accumulator with Memory Byte	A ← (M)	—	—	×	×	—	IMM DIR EXT IX2 IX1 IX	A6 B6 C6 D6 E6 F6	ii dd hh ll ee ff ff	2 3 4 5 4 3
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X	Load Index Register with Memory Byte	X ← (M)	—	—	×	×	—	IMM DIR EXT IX2 IX1 IX	AE BE CE DE EE FE	ii dd hh ll ee ff ff	2 3 4 5 4 3
LSL *opr* LSLA LSLX LSL *opr*,X LSL ,X	Logical Shift Left (Same as ASL)		—	—	×	×	×	DIR INH INH IX1 IX	38 48 58 68 78	dd ff	5 3 3 6 5
LSR *opr* LSRA LSRX LSR *opr*,X LSR ,X	Logical Shift Right		—	—	0	×	×	DIR INH INH IX1 IX	34 44 54 64 74	dd ff	5 3 3 6 5
MUL	Unsigned Multiply	X : A ← (X) × (A)	0	—	—	—	0	INH	42		11
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X	Negate Byte (Two's Complement)	M ← –(M) = $00 – (M) A ← –(A) = $00 – (A) X ← –(X) = $00 – (X) M ← –(M) = $00 – (M) M ← –(M) = $00 – (M)	—	—	×	×	×	DIR INH INH IX1 IX	30 40 50 60 70	ii ff	5 3 3 6 5

Table 10-6. Instruction Set Summary (Continued)

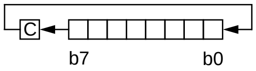
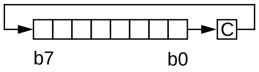
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
NOP	No Operation		—	—	—	—	—	INH	9D		2
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X	Logical OR Accumulator with Memory	A ← (A) ∨ (M)	—	—	×	×	—	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff	2 3 4 5 4 3
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X	Rotate Byte Left through Carry Bit		—	—	×	×	×	DIR INH INH IX1 IX	39 49 59 69 79	dd ff	5 3 3 6 5
ROR *opr* RORA RORX ROR *opr*,X ROR ,X	Rotate Byte Right through Carry Bit		—	—	×	×	×	DIR INH INH IX1 IX	36 46 56 66 76	dd ff	5 3 3 6 5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	×	×	×	×	×	INH	80		6
RTS	Return from Subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)						INH			
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X	Subtract Memory Byte and Carry Bit from Accumulator	A ← (A) – (M) – (C)	—	—	×	×	×	IMM DIR EXT IX2 IX1 IX	A2 B2 C2 D2 E2 F2	ii dd hh ll ee ff ff	2 3 4 5 4 3
SEC	Set Carry Bit	C ← 1	—	—	—	—	1	INH	99		2
SEI	Set Interrupt Mask	I ← 1	—	1	—	—	—	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X	Store Accumulator in Memory	M ← (A)	—	—	×	×	—	DIR EXT IX2 IX1 IX	B7 C7 D7 E7 F7	dd hh ll ee ff ff	4 5 6 5 4
STOP	Stop Oscillator and Enable $\overline{\text{IRQ}}$ Pin		—	0	—	—	—	INH	8E		2
STX *opr* STX *opr* STX *opr*,X STX *opr*,X STX ,X	Store Index Register In Memory	M ← (X)	—	—	×	×	—	DIR EXT IX2 IX1 IX	BF CF DF EF FF	dd hh ll ee ff ff	4 5 6 5 4

Table 10-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
SUB #*opr* SUB *opr* SUB *opr* SUB *opr*,X SUB *opr*,X SUB ,X	Subtract Memory Byte from Accumulator	A ← (A) – (M)	—	—	×	×	×	IMM DIR EXT IX2 IX1 IX	A0 B0 C0 D0 E0 F0	ii dd hh ll ee ff ff	2 3 4 5 4 3
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	—	1	—	—	—	INH	83		10
TAX	Transfer Accumulator to Index Register	X ← (A)	—	—	—	—	—	INH	97		2
TST *opr* TSTA TSTX TST *opr*,X TST ,X	Test Memory Byte for Negative or Zero	(M) – $00	—	—	—	—	—	DIR INH INH IX1 IX	3D 4D 5D 6D 7D	dd ff	4 3 3 5 4
TXA	Transfer Index Register to Accumulator	A ← (X)	—	—	—	—	—	INH	9F		2
WAIT	Stop CPU Clock and Enable Interrupts		—	×	—	—	—	INH	8F		2

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

Table 10-7. Opcode Map

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

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

	LSB \ MSB	0	MSB of Opcode in Hexadecimal
LSB of Opcode in Hexadecimal	0	5 BRSET0 3 DIR	Number of Cycles Opcode Mnemonic Number of Bytes/Addressing Mode

SECTION 11
ELECTRICAL SPECIFICATIONS

11.1 Introduction

This section contains electrical and timing specifications.

11.2 Maximum Ratings

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 (for example, either V_{SS} or V_{DD}). Internal I/O pulldowns may be used.

Table 11-1. Maximum Ratings*

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	–0.3 to +7.0	V
Input Voltage	V_{IN}	V_{SS} –0.3 to V_{DD} +0.3	V
$\overline{IRQ}/V_{PP}$ Pin	V_{IN}	V_{SS} –0.3 to 2V_{DD} +0.3	V
Current Drain Per Pin Excluding V_{DD}, V_{SS}, and PA4–PA7	I	25	mA
Storage Temperature Range	T_{STG}	–65 to +150	°C

*Voltages are referenced to V_{SS}

11.3 Thermal Characteristics

Table 11-2. Thermal Characteristics

Rating	Symbol	Value	Unit
Thermal Resistance MC68HC705J1AP[1] MC68HC705J1ADW[2]	 θ_{JA} θ_{JA}	 60 60	 °C/W °C/W
Operating Temperature Range MC68HC705J1A (Standard) MC68HC705J1A (Extended)	T_A	T_L to T_H 0 to +70 –40 to +85	°C

NOTES:

1. P = Plastic dual-in-line package (PDIP)
2. DW = Small outline integrated circuit (SOIC)

11.4 Power Considerations

The average chip-junction temperature, $T_{J,}$ can be obtained in °C from:

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

where:

T_A = Ambient temperature, °C

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

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

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

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

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

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

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

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

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

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

11.5 DC Electrical Characteristics

Table 11-3. DC Electrical Characteristics (V_{DD} = 5 V)[1]

Characteristic	Symbol	Min	Typ[2]	Max	Unit
Output Voltage I_{LOAD} = 10.0 µA 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-PA7, PB0-PB5	 V_{OH}	 V_{DD} –0.8	 —	 —	 V
Output Low Voltage (I_{LOAD} = 1.6 mA) PA0-PA3, PB0-PB5 (I_{LOAD} = 10.0 mA) PA4-PA7	 V_{OL} V_{OL}	 — —	 — —	 0.4 0.4	 V V
Input High Voltage PA0-PA7, PB0-PB5, $\overline{\text{IRQ}}/V_{PP}$, $\overline{\text{RESET}}$, OSC1	 V_{IH}	 0.7 × V_{DD}	 —	 V_{DD}	 V
Input Low Voltage PA0-PA7, PB0-PB5, $\overline{\text{IRQ}}/V_{PP}$, $\overline{\text{RESET}}$, OSC1	 V_{IL}	 V_{SS}	 —	 0.2 × V_{DD}	 V
Supply Current Run[4] Wait[3, 4, 5, 7] Stop[5, 6] 25°C –40°C to +85°C	 I_{DD} I_{DD} I_{DD} I_{DD}	 — — — —	 3.5 0.45 0.2 2.0	 6.0 2.75 10 20	 mA mA µA µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB5 (without individual pulldown activated)	 I_{IL}	 —	 —	 ±10	 µA
Input Pulldown Current PA0-PA7, PB0-PB5 (with individual pulldown activated)	 I_{IL}	 40	 80	 200	 µA
Input Pullup Current $\overline{\text{RESET}}$	 I_{IL}	 –15	 –35	 –80	 µA
Input Current[8] $\overline{\text{RESET}}$, $\overline{\text{IRQ}}/V_{PP}$, OSC1	 I_{IN}	 —	 —	 ±1	 µA
Capacitance Ports (as Input or Output) $\overline{\text{RESET}}$, $\overline{\text{IRQ}}/V_{PP}$, OSC1, OSC2	 C_{OUT} C_{IN}	 — —	 — —	 12 8	 pF pF
Crystal/Ceramic Resonator Oscillator Mode Internal Resistor OSC1 to OSC2	 R_{OSC}	 1.0	 2.0	 3.0	 MΩ

NOTES:

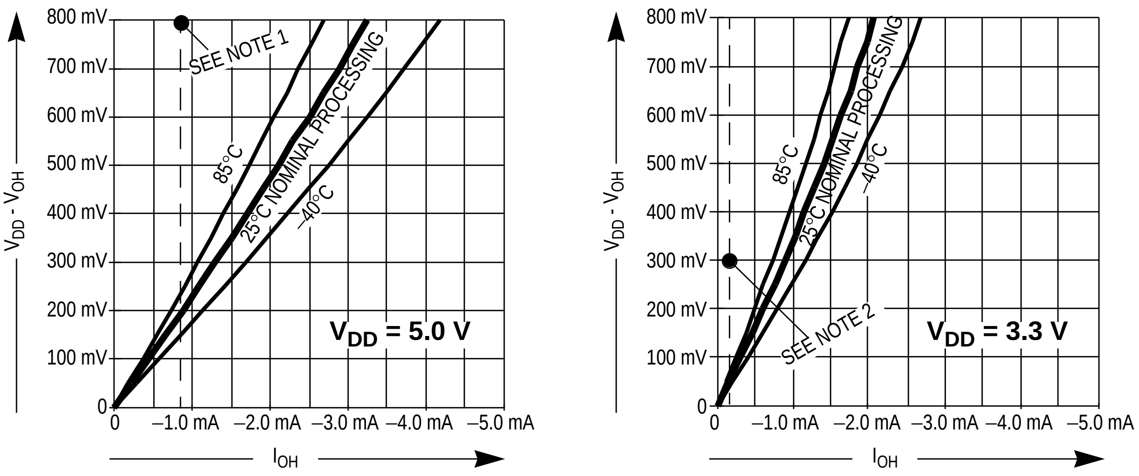
1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = –40°C to +85°C, unless otherwise noted
2. Typical values reflect average measurements at midpoint of voltage range, 25°C.
3. Wait I_{DD}: Only timer system active.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{OP} = 2.1 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} -0.2 V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.
8. Only input high current rated to +1 µA on $\overline{\text{RESET}}$.

Table 11-4. DC Electrical Characteristics (V_{DD} = 3.3 V)[1]

Characteristic	Symbol	Min	Typ[2]	Max	Unit
Output Voltage I_{LOAD} = 10.0 µA I_{LOAD} = 10.0 µA	 V_{OL} V_{OH}	 — V_{DD} –0.1	 — —	 0.1 —	 V V
Output High Voltage (I_{LOAD} = –0.2 mA) PA0-PA7, PB0-PB5	 V_{OH}	 V_{DD} –0.3	 —	 —	 V
Output Low Voltage (I_{LOAD} = 0.4mA) PA0-PA3, PB0-PB5 (I_{LOAD} = 5.0mA) PA4-PA7	 V_{OL} V_{OL}	 — —	 — —	 0.3 0.3	 V V
Input High Voltage PA0-PA7, PB0-PB5, $\overline{\text{IRQ}}/V_{PP}$, $\overline{\text{RESET}}$, OSC1	 V_{IH}	 0.7 × V_{DD}	 —	 V_{DD}	 V
Input Low Voltage PA0-PA7, PB0-PB5, $\overline{\text{IRQ}}/V_{PP}$, $\overline{\text{RESET}}$, OSC1	 V_{IL}	 V_{SS}	 —	 0.2 × V_{DD}	 V
Supply Current Run[4] Wait[3, 4, 5, 7] Stop[5, 6] 25°C –40°C to +85°C	 I_{DD} I_{DD} I_{DD} I_{DD}	 — — — —	 1.2 0.25 0.1 1	 4.0 1.5 5 10	 mA mA µA µA
I/O Ports Hi-Z Leakage Current PA0-PA7, PB0-PB5 (without individual pulldown activated)	 I_{IL}	 —	 —	 ±10	 µA
Input Pulldown Current PA0-PA7, PB0-PB5 (with individual pulldown activated)	 I_{IL}	 15	 30	 100	 µA
Input Pullup Current $\overline{\text{RESET}}$	 I_{IL}	 –10	 –20	 –40	 µA
Input Current[8] $\overline{\text{RESET}}$, $\overline{\text{IRQ}}/V_{PP}$, OSC1	 I_{IN}	 —	 —	 ±1	 µA
Capacitance Ports (as Input or Output) $\overline{\text{RESET}}$, $\overline{\text{IRQ}}/V_{PP}$, OSC1, OSC2	 C_{OUT} C_{IN}	 — —	 — —	 12 8	 pF pF
Crystal/Ceramic Resonator Oscillator Mode Internal Resistor OSC1 to OSC2	 R_{OSC}	 1.0	 2.0	 3.0	 MΩ

NOTES:

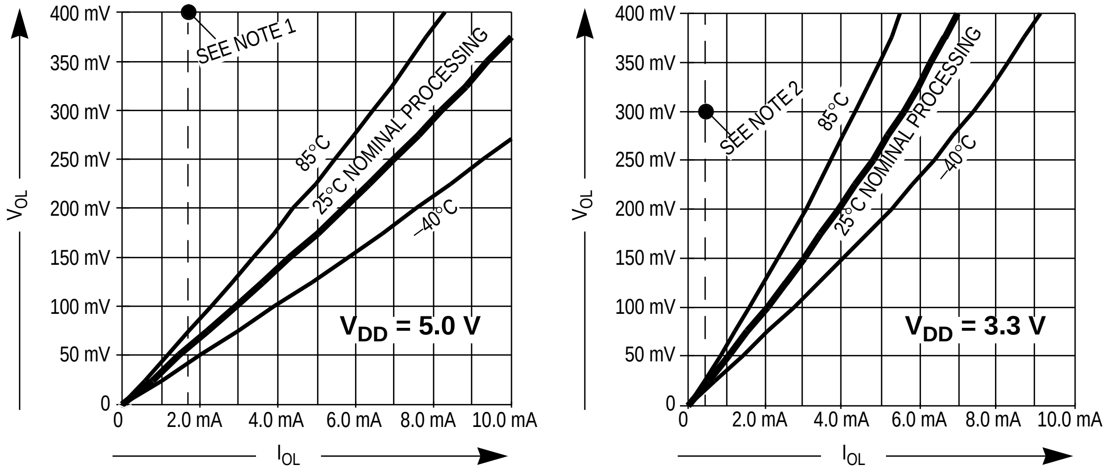
1. V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = –40°C to +85°C, unless otherwise noted
2. Typical values reflect average measurements at midpoint of voltage range, 25°C.
3. Wait I_{DD}: Only timer system active.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{OP} = 1.0 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} -0.2 V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.
8. Only input high current rated to +1 µA on $\overline{\text{RESET}}$.



NOTES:

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

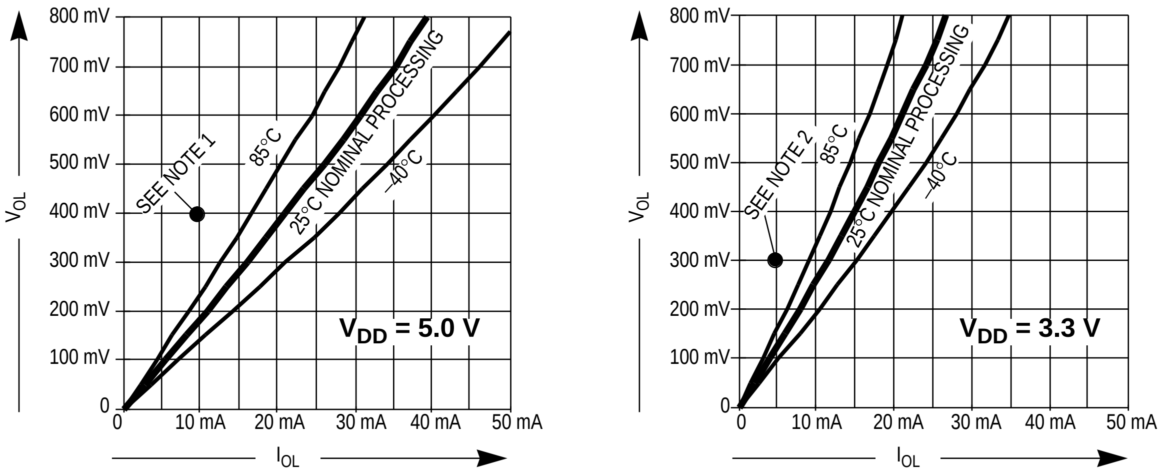
Figure 11-1. PA0–PA7, PB0–PB5 Typical High-Side Driver Characteristics



NOTES:

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

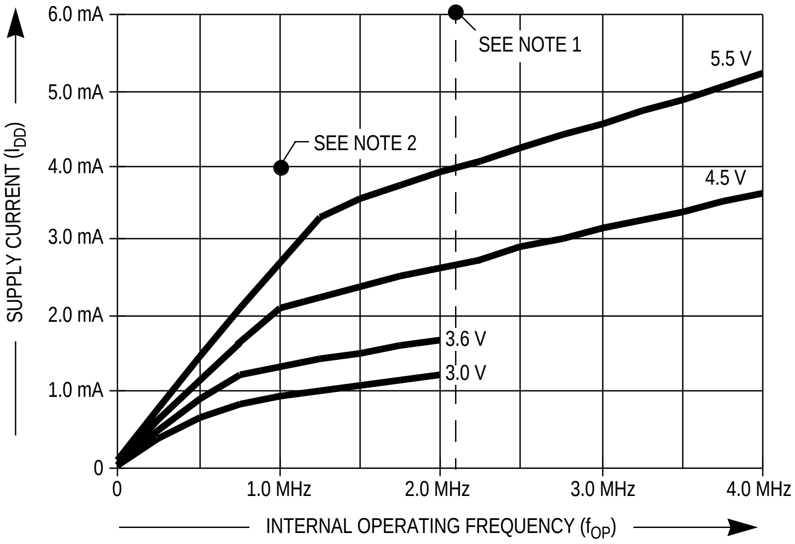
Figure 11-2. PA0–PA3, PB0–PB5 Typical Low-Side Driver Characteristics



NOTES:

1. At V_{DD} = 5.0 V, devices are specified and tested for $V_{OL} \leq 400$ mV @ I_{OL} = 10.0 mA.
2. At V_{DD} = 3.3 V, devices are specified and tested for $V_{OL} \leq 300$ mV @ I_{OL} = 5.0 mA.

Figure 11-3. PA4–PA7 Typical Low-Side Driver Characteristics

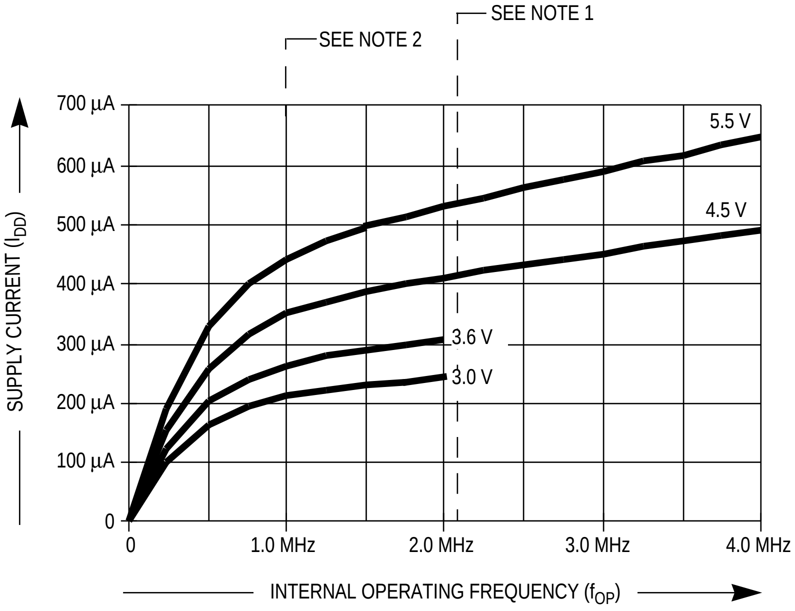


NOTES:

1. At V_{DD} = 5.0 V, devices are specified and tested for $I_{DD} \leq 6.0$ mA @ f_{OP} = 2.1 MHz.
2. At V_{DD} = 3.3 V, devices are specified and tested for $I_{DD} \leq 4.0$ mA @ f_{OP} = 1.0 MHz.

11

Figure 11-4. Typical Operating I_{DD} (25°C)



NOTES:

1. At V_{DD} = 5.0 V, devices are specified and tested for $I_{DD} \leq 2.75$ mA @ f_{OP} = 2.1 MHz.
2. At V_{DD} = 3.3 V, devices are specified and tested for $I_{DD} \leq 1.5$ mA @ f_{OP} = 1.0 MHz.

Figure 11-5. Typical Wait Mode I_{DD} (25°C)

11.6 EPROM Programming Characteristics

Table 11-5. EPROM Programming Characteristics[1]

Characteristic	Symbol	Min	Typ	Max	Unit
Programming Voltage $\overline{IRQ}/V_{PP}$	V_{PP}	16.0	16.5	17.0	V
Programming Current $\overline{IRQ}/V_{PP}$	I_{PP}	—	3.0	10.0	mA
Programming Time Per Array Byte MOR	 t_{EPGM} t_{MPGM}	 4 4	 — —	 — —	 ms ms

NOTE: 1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = –40°C to +85°C, unless otherwise noted.

11.7 Control Timing

Table 11-6. Control Timing (5 V)[1]

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Oscillator Option External Clock Source	 f_{OSC} f_{OSC}	 — DC	 4.2 4.2	 MHz MHz
Internal Operating Frequency Crystal Oscillator ($f_{OSC} \div 2$) External Clock ($f_{OSC} \div 2$)	 f_{OP} f_{OP}	 — DC	 2.1 2.1	 MHz MHz
Cycle Time ($1/f_{OP}$)	t_{CYC}	476	—	ns
$\overline{\text{RESET}}$ Pulse Width Low	t_{RL}	1.5	—	t_{CYC}
$\overline{\text{IRQ}}$ Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	1.5	—	t_{CYC}
$\overline{\text{IRQ}}$ Interrupt Pulse Width (Edge and Level Triggered)	t_{ILIL}	1.5	Note 2	t_{CYC}
PA0 through PA3 Interrupt Pulse Width High (Edge-Triggered)	t_{IHIL}	1.5	—	t_{CYC}
PA0 through PA3 Interrupt Pulse Width (Edge and Level Triggered)	t_{IHIH}	1.5	Note 2	t_{CYC}
OSC1 Pulse Width	t	200	—	ns

NOTES:

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = –40°C to +85°C, unless otherwise noted
2. The maximum width t_{ILIL} or t_{IHIH} should not be more than the number of cycles it takes to execute the interrupt service routine plus 19 t_{CYC} or the interrupt service routine will be re-entered.

Table 11-7. Control Timing (3.3 V)[1]

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Oscillator Option External Clock Source	 f_{OSC} f_{OSC}	 — DC	 2.0 2.0	 MHz MHz
Internal Operating Frequency Crystal Oscillator ($f_{OSC} \div 2$) External Clock ($f_{OSC} \div 2$)	 f_{OP} f_{OP}	 — DC	 1.0 1.0	 MHz MHz
Cycle Time ($1/f_{OP}$)	t_{CYC}	1000	—	ns
$\overline{\text{RESET}}$ Pulse Width Low	t_{RL}	1.5	—	t_{CYC}
$\overline{\text{IRQ}}$ Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	1.5	—	t_{CYC}
$\overline{\text{IRQ}}$ Interrupt Pulse Width (Edge and Level Triggered)	t_{ILIL}	1.5	Note 2	t_{CYC}
PA0 through PA3 Interrupt Pulse Width High (Edge-Triggered)	t_{IHIL}	1.5	—	t_{CYC}
PA0 through PA3 Interrupt Pulse Width (Edge and Level Triggered)	t_{IHIH}	1.5	Note 2	t_{CYC}
OSC1 Pulse Width	t	400	—	ns

NOTES:

1. V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = –40°C to +85°C, unless otherwise noted
2. The maximum width t_{ILIL} or t_{IHIH} should not be more than the number of cycles it takes to execute the interrupt service routine plus 19 t_{CYC} or the interrupt service routine will be re-entered.

11

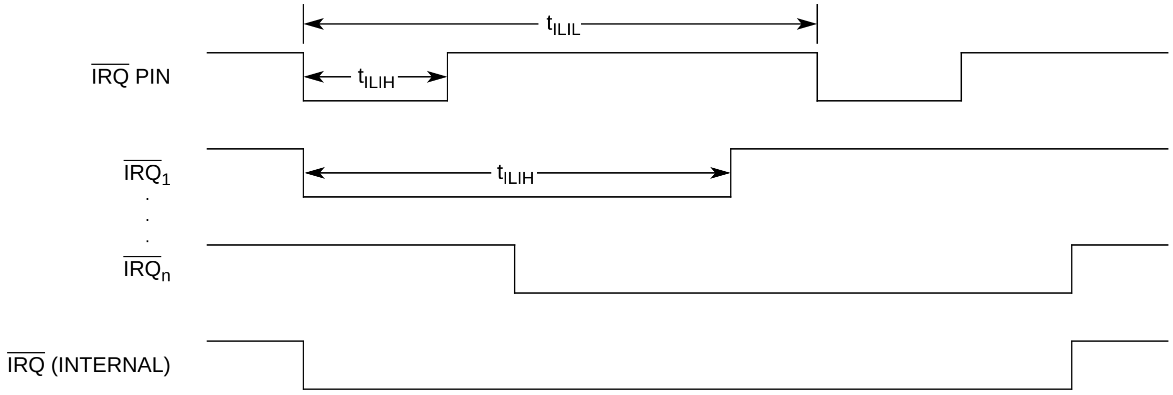
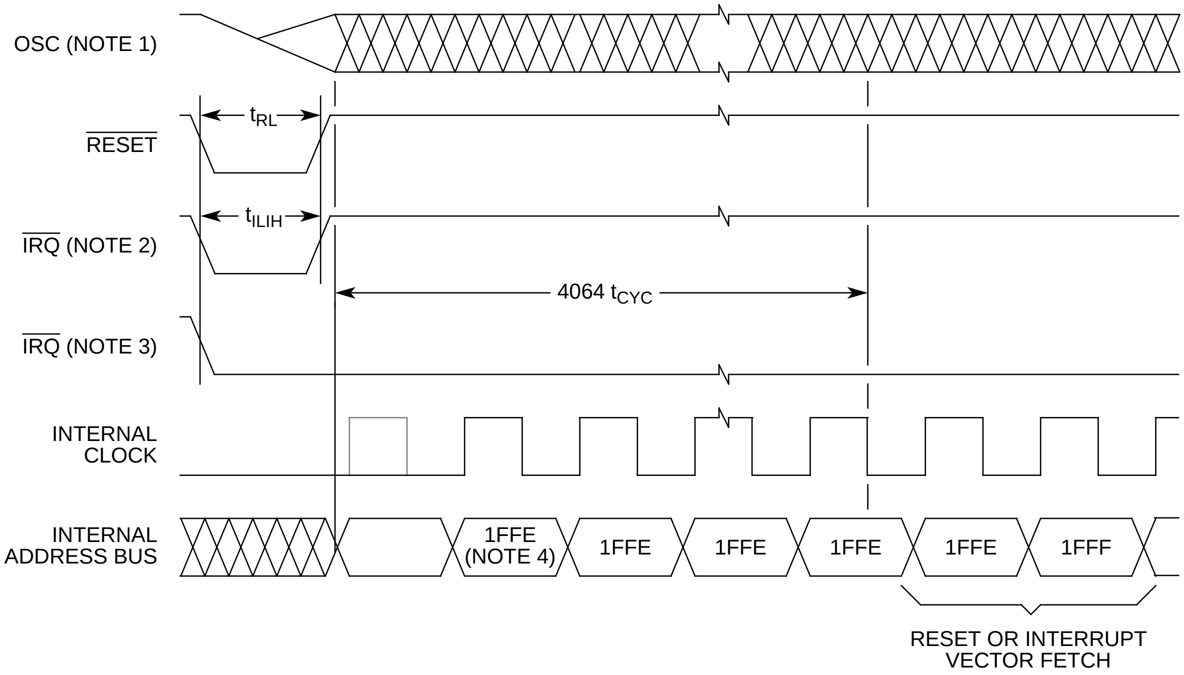


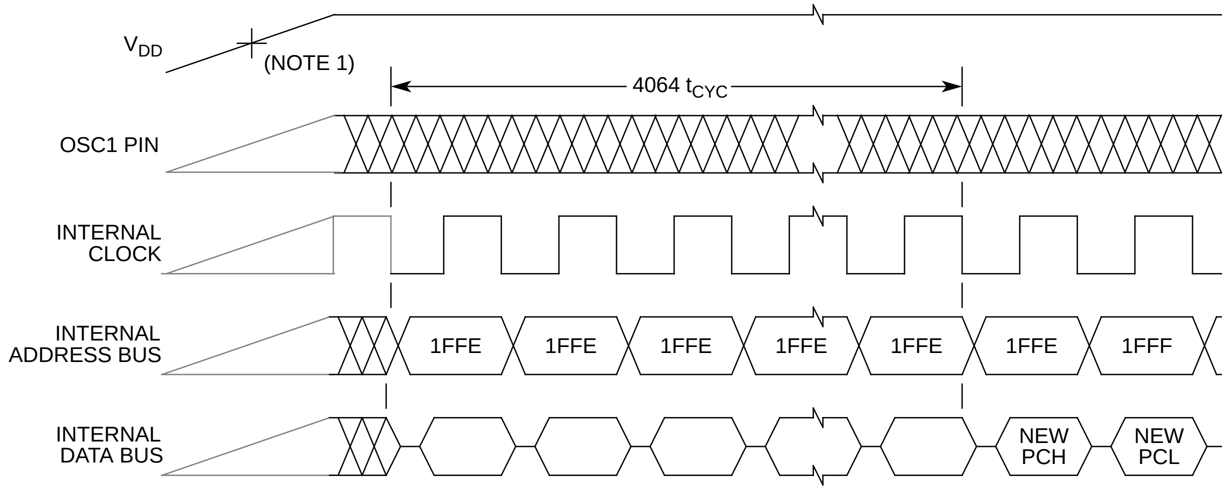
Figure 11-6. External Interrupt Timing



NOTES:

1. Internal clocking from OSC1 pin.
2. Edge-triggered external interrupt mask option.
3. Edge- and level-triggered external interrupt mask option.
4. Reset vector shown as example.

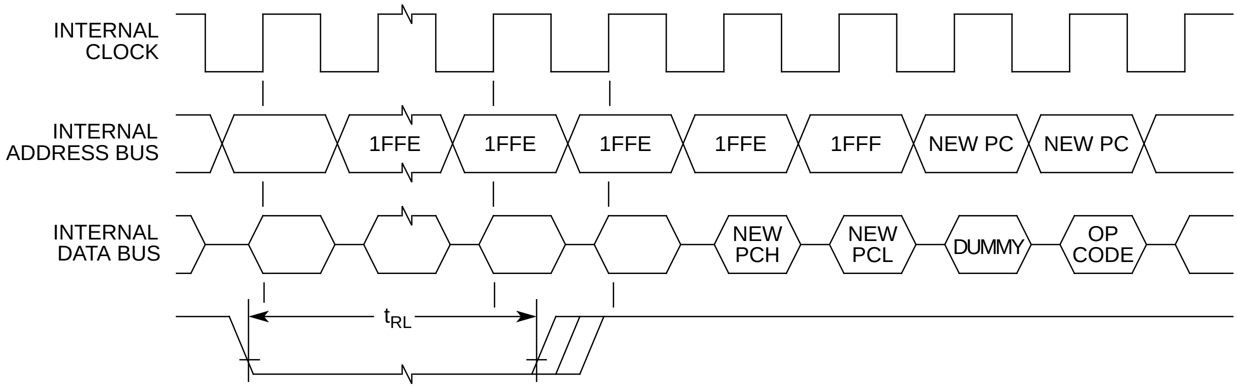
Figure 11-7. Stop Mode Recovery Timing



NOTES:

1. Power-on reset threshold is typically between 1 V and 2 V.
2. Internal clock, internal address bus, and internal data bus are not available externally.

Figure 11-8. Power-On Reset Timing



NOTES:

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

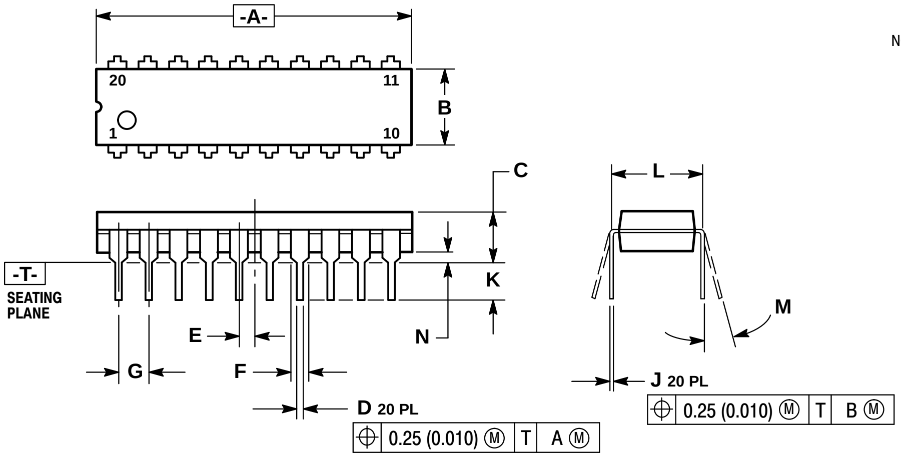
Figure 11-9. External Reset Timing

SECTION 12
MECHANICAL SPECIFICATIONS

12.1 Introduction

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

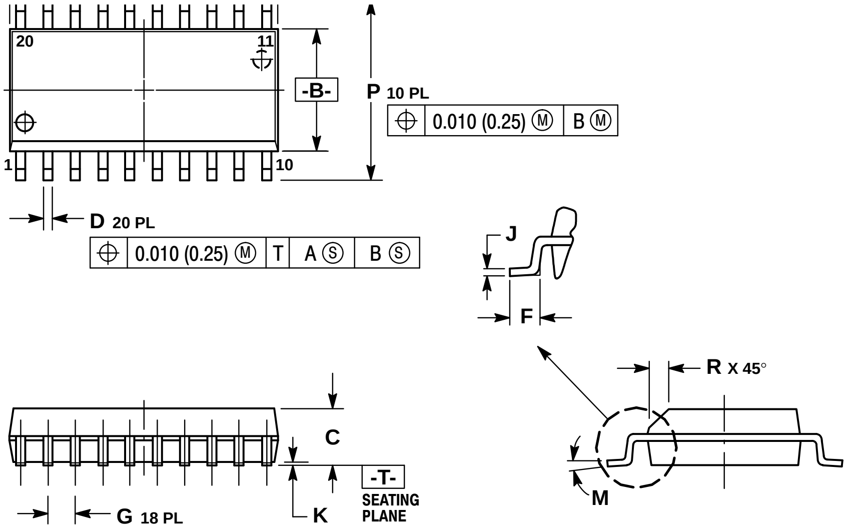
12.2 PDIP Package (Case 738-03)



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

12.3 SOIC Package (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