MC68HC705K1/D
REV. 1

HC05

MC68HC705K1

TECHNICAL
DATA

MOTOROLA

MC68HC705K1

HCMOS MICROCONTROLLER UNIT

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

SECTION 1
GENERAL DESCRIPTION

The MC68HC705K1 is a member of the low-cost, high-performance MCHC05 Family of 8-bit microcontroller units (MCUs). The M68HC05 Family is based on the customer-specified integrated circuit (CSIC) design strategy. All MCUs in the family use the popular M68HC05 central processor unit (CPU) and are available with a variety of subsystems, memory sizes and types, and package types.

On-chip memory of the MC68HC705K1 includes 504 bytes of erasable, programmable ROM (EPROM). In packages without the transparent window for EPROM erasure, the 504 EPROM bytes serve as one-time programmable ROM (OTPROM).

1.1 Features

Features of the MCU include the following:

- Popular M68HC05 CPU
- Memory-Mapped Input/Output (I/O) Registers
- 504 Bytes of EPROM/OTPROM Including 8 User Vector Locations
- 32 Bytes of User RAM
- 64-bit Personality EPROM
- 10 Bidirectional I/O Pins with the following features:
  - Software Programmable Pulldown Devices
  - 4 I/O Pins with 8 mA Current Sinking Capability
  - 4 I/O Pins with Maskable External Interrupt Capability
- Hardware Mask and Flag for External Interrupts
- Fully Static Operation with no Minimum Clock Speed
- On-Chip Oscillator with Connections for a Crystal or Ceramic Resonator or for a Mask-Optional Two-Pin or Three-Pin Resistor-Capacitor (RC) Oscillator

- Computer Operating Properly (COP) Watchdog
- 15-Bit Multifunction Timer with Real-Time Interrupt Circuit
- Power-Saving Stop, Wait, Halt, and Data-Retention Modes
- 8×8 Unsigned Multiply Instruction
- Illegal Address Reset
- Low Voltage Reset
- 16-Pin Plastic Dual In-Line Package (PDIP)
- 16-Pin Small Outline Integrated Circuit Package (SOIC)
- 16-Pin Ceramic DIP (Cerdip)

1.2 Mask Options

The following MC68HC705K1 options are programmable in the mask option register:

- Enabled or disabled COP watchdog
- Edge-triggered or edge- and level-triggered external interrupt pins
- Enabled or disabled port A external interrupt function
- Enabled or disabled low-voltage reset function
- Enabled or disabled STOP instruction
- Oscillator driven by crystal or ceramic resonator or oscillator driven by RC circuit
- Two-pin RC-driven oscillator or three-pin RC-driven oscillator
- Enabled or disabled port A and port B programmable pulldown devices

The mask option register is an EPROM/OTPROM byte at location $0017. **SECTION 9 EPROM/OTPROM** describes the mask option register and the EPROM/OTPROM programming procedure.

1.3 MCU Structure

Figure 1-1 shows the structure of the MC68HC705K1 MCU.

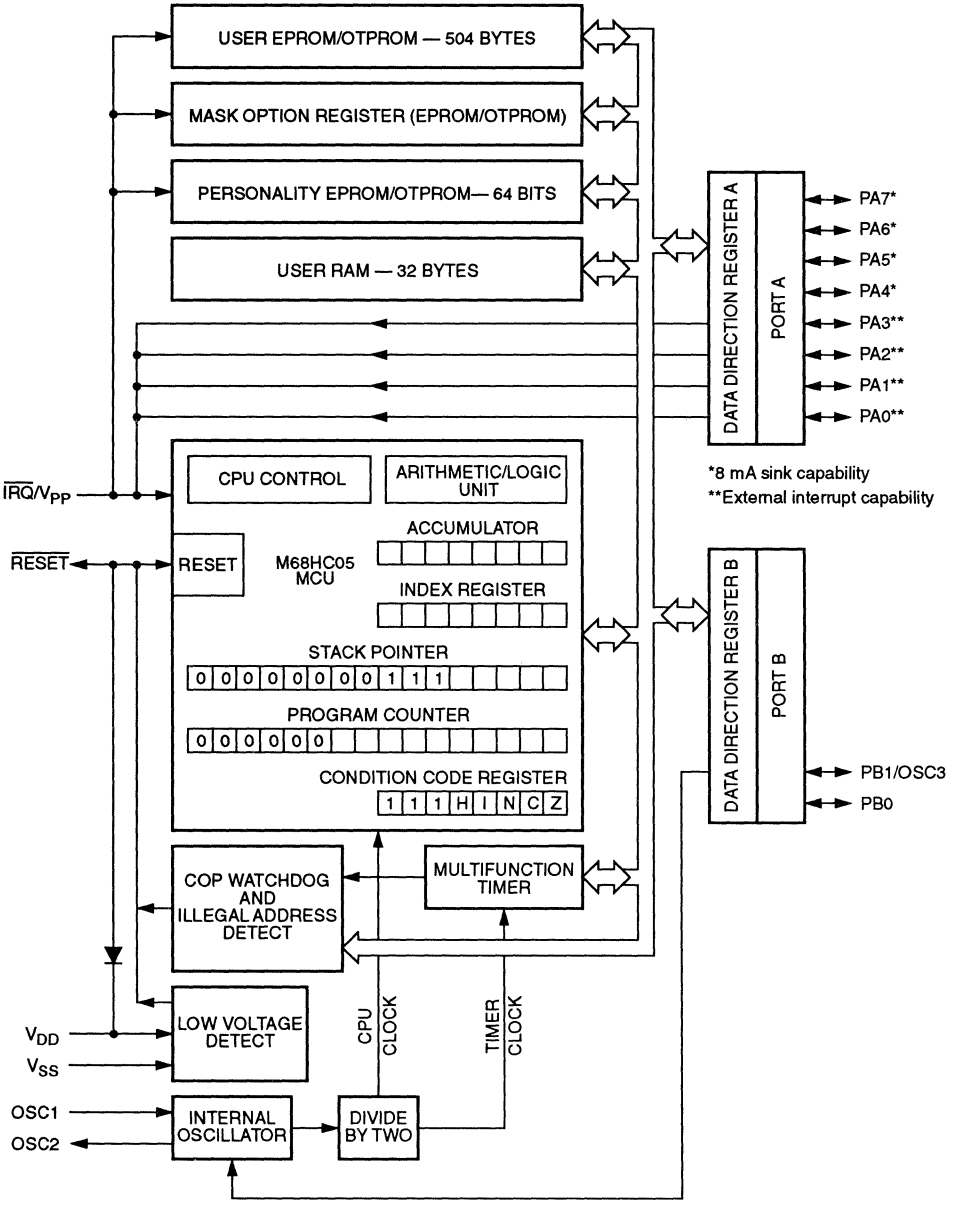


Figure 1-1 . MC68HC705K1 Block Diagram

1.4 Pin Assignments

Figure 1-2 shows the MC68HC705K1 pin assignments.

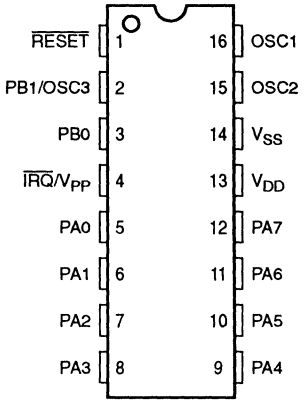


Figure 1-2 . Pin Assignments

1.4.1 V_{DD} and V_{SS}

V_{DD} and V_{SS} are the power supply and ground pins. The MCU operates from a single 5-V power supply.

Very fast signal transitions occur on the MCU pins, placing high short-duration current demands on the power supply. To prevent noise problems, take special care to provide good power supply bypassing at the MCU. Place bypass capacitors as close to the MCU as possible, as Figure 1-3 shows.

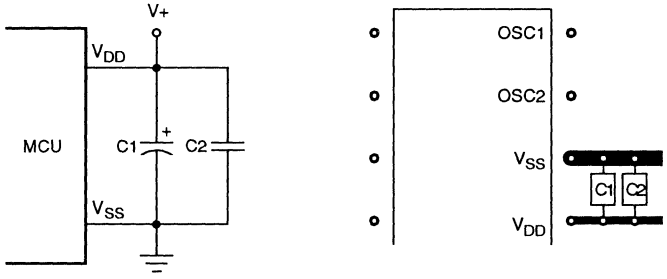


Figure 1-3 . Bypassing Layout Recommendation

1.4.2 OSC1 and OSC2

The OSC1, OSC2, and PB1/OSC3 pins are the control connections for the two-pin or three-pin on-chip oscillator. The oscillator can be driven by any of the following:

- Crystal
- Ceramic resonator
- Resistor-capacitor network
- External clock signal

The frequency of the internal oscillator is f_{OSC}. The MCU divides the internal oscillator output by two to produce the internal clock with a frequency of f_{OP}.

1.4.2.1 Crystal

The circuit in Figure 1-4 shows a typical crystal oscillator circuit for an AT-cut, parallel resonant crystal. Follow the crystal supplier's recommendations, as the crystal parameters determine the external component values required to provide 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.

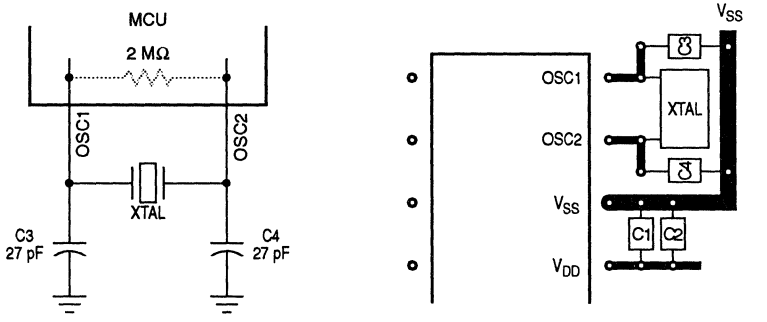


Figure 1-4 . Crystal Connections

NOTE

Use an AT-cut crystal and not a strip or tuning fork crystal. The MCU may overdrive or have the wrong characteristic impedance for a strip or tuning fork crystal.

To use the crystal-driven oscillator, the RC and PIN3 bits in the mask option register must be clear. See **9.4 Mask Option Register (MOR)**. Clearing the RC bit connects an internal 2-MΩ start-up resistor between OSC1 and OSC2.

1.4.2.2 Ceramic Resonator

To reduce cost, use a ceramic resonator in place of the crystal. Use the circuit in Figure 1-5 for a two-pin ceramic resonator or Figure 1-6 for a three-pin ceramic resonator, and follow the resonator manufacturer's recommendations. 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 layout capacitances. To minimize output distortion, mount the resonator and capacitors as close as possible to the pins.

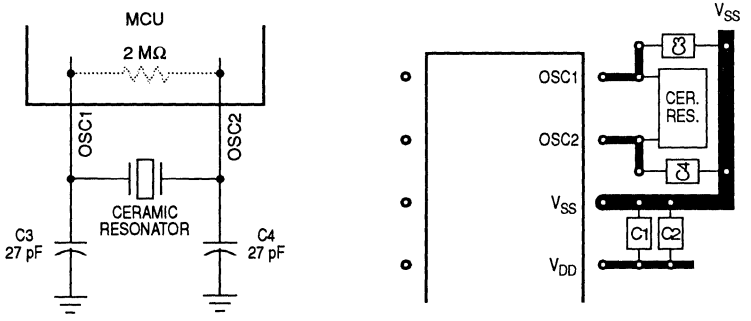


Figure 1-5 . Two-Pin Ceramic Resonator Connections

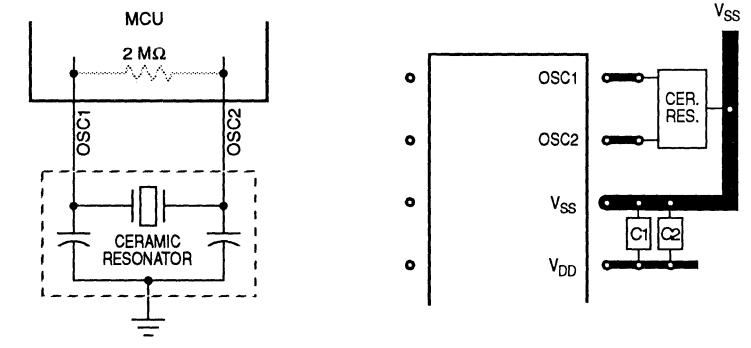


Figure 1-6 . Three-Pin Ceramic Resonator Connections

To use the resonator-driven oscillator, the RC bit in the mask option register must be clear. See **9.4 Mask Option Register (MOR)**. Clearing the RC bit connects an internal 2-MΩ start-up resistor between OSC1 and OSC2.

1.4.2.3 Two-Pin RC Oscillator

For maximum cost reduction, use the two-pin RC oscillator configuration shown in Figure 1-7. The OSC2 signal is a square wave, and the signal on OSC1 is a triangular wave. The optimum frequency for the two-pin oscillator configuration is 2 MHz.

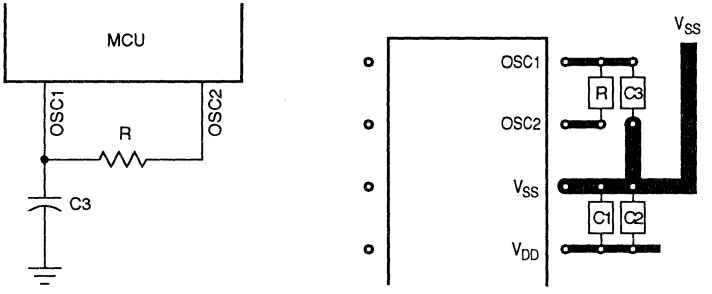


Figure 1-7 . Two-Pin RC Oscillator Connections

To use the two-pin RC oscillator configuration, the RC bit in the mask option register must be programmed to a logic one. Setting the RC bit disconnects the internal start-up resistor. The PIN3 bit in the mask option register must remain erased (logic zero). The PIN3 bit selects the three-pin RC oscillator configuration. See **9.4 Mask Option Register (MOR)**.

1.4.2.4 Three-Pin RC Oscillator

Another low-cost option is the three-pin RC oscillator configuration shown in Figure 1-8. The three-pin oscillator is more stable than the two-pin oscillator. The OSC2 and PB1/OSC3 signals are square waves, and the signal on OSC1 is a triangular wave. The three-pin RC oscillator configuration is recommended for frequencies of 1 MHz and less.

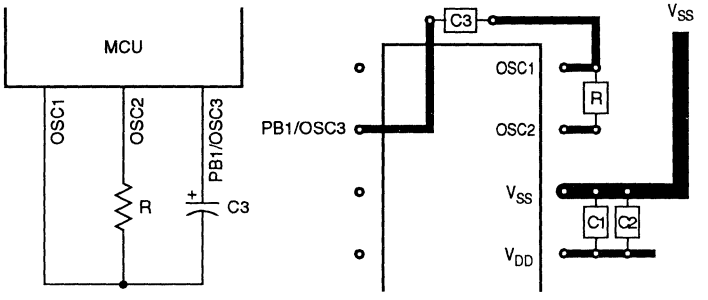


Figure 1-8 . Three-Pin RC Oscillator Connections

To use the three-pin RC oscillator configuration, both the RC and PIN3 bits in the mask option register must be programmed to logic ones. See **9.4 Mask Option Register (MOR)**.

1.4.2.5 External Clock

An external clock from another CMOS-compatible device can drive the OSC1 input, with the OSC2 pin unconnected, as Figure 1-9 shows.

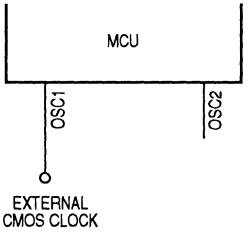


Figure 1-9 . External Clock ConnectionsRESET

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

A logic zero on the $\overline{\text{RESET}}$ pin forces the MCU to a known start-up state. See **5.1.2 Exernal Reset** for more information.

1.4.4 $\overline{\text{IRQ}}/V_{PP}$

The $\overline{\text{IRQ}}/V_{PP}$ pin has the following functions:

- Applying asynchronous external interrupt signals (See **4.1.2 External Interrupts**.)
- Applying the EPROM/OTPROM programming voltage (See **9.2 EPROM/OTPROM Programming**.)

1.4.5 PA7–PA0

PA7–PA0 are the pins of port A, a general-purpose bidirectional I/O port. See **7.2 Port A**.

1.4.6 PB1/OSC3 and PB0

PB1/OSC3 and PB0 are the pins of port B, a general-purpose bidirectional I/O port. See **7.3 Port B**.

SECTION 2
MEMORY

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

2.1 Memory Map

The CPU can address 1 Kbyte of memory space. The program counter typically advances one address at a time through the memory, reading the program instructions and data. 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.

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

2.2 Input/Output Section

The first 32 addresses of the memory space, $0001–$001F, are the I/O section. These are the addresses of the I/O control registers, status registers, and data registers. See Figure 2-2.

2.3 RAM

The 32 addresses from $00E0 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.

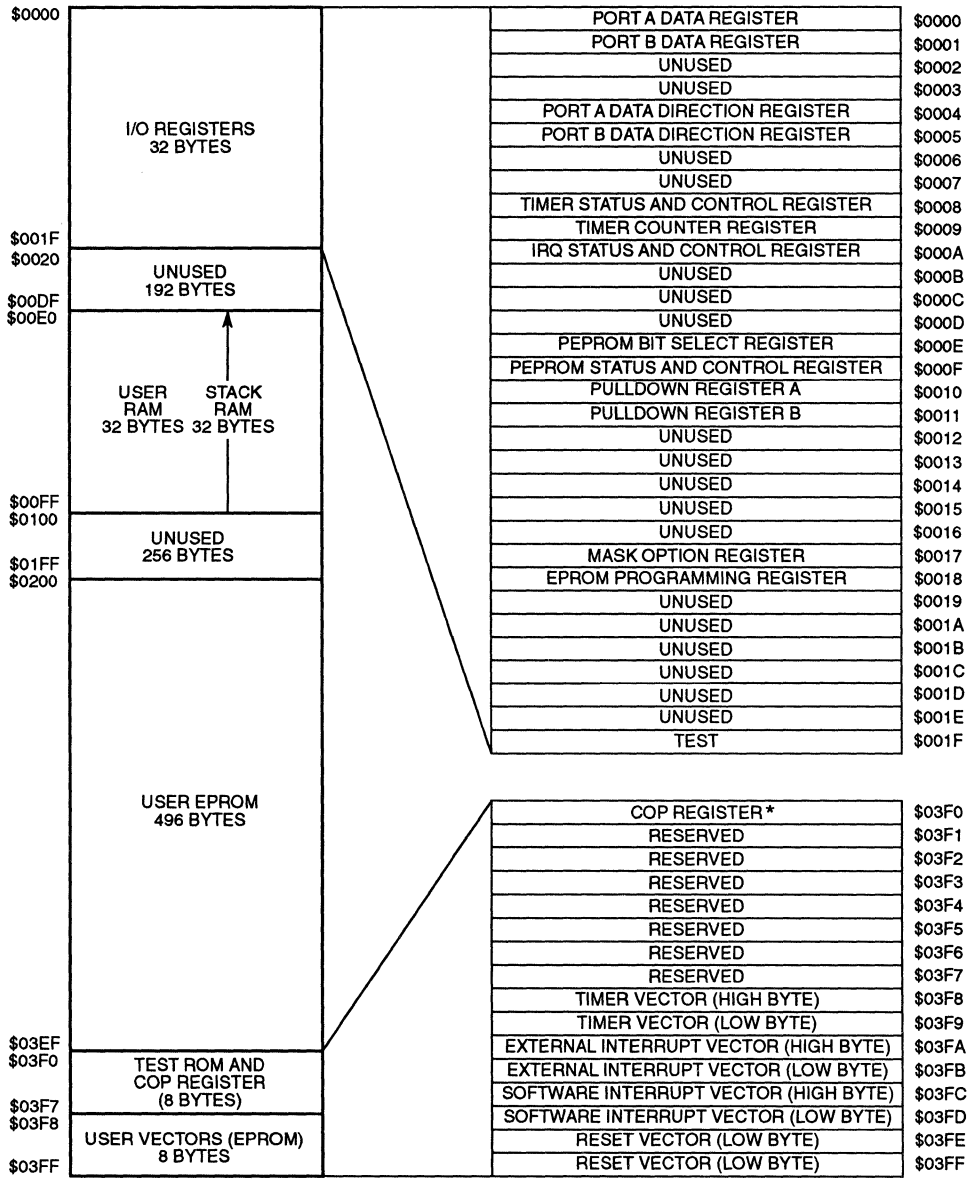
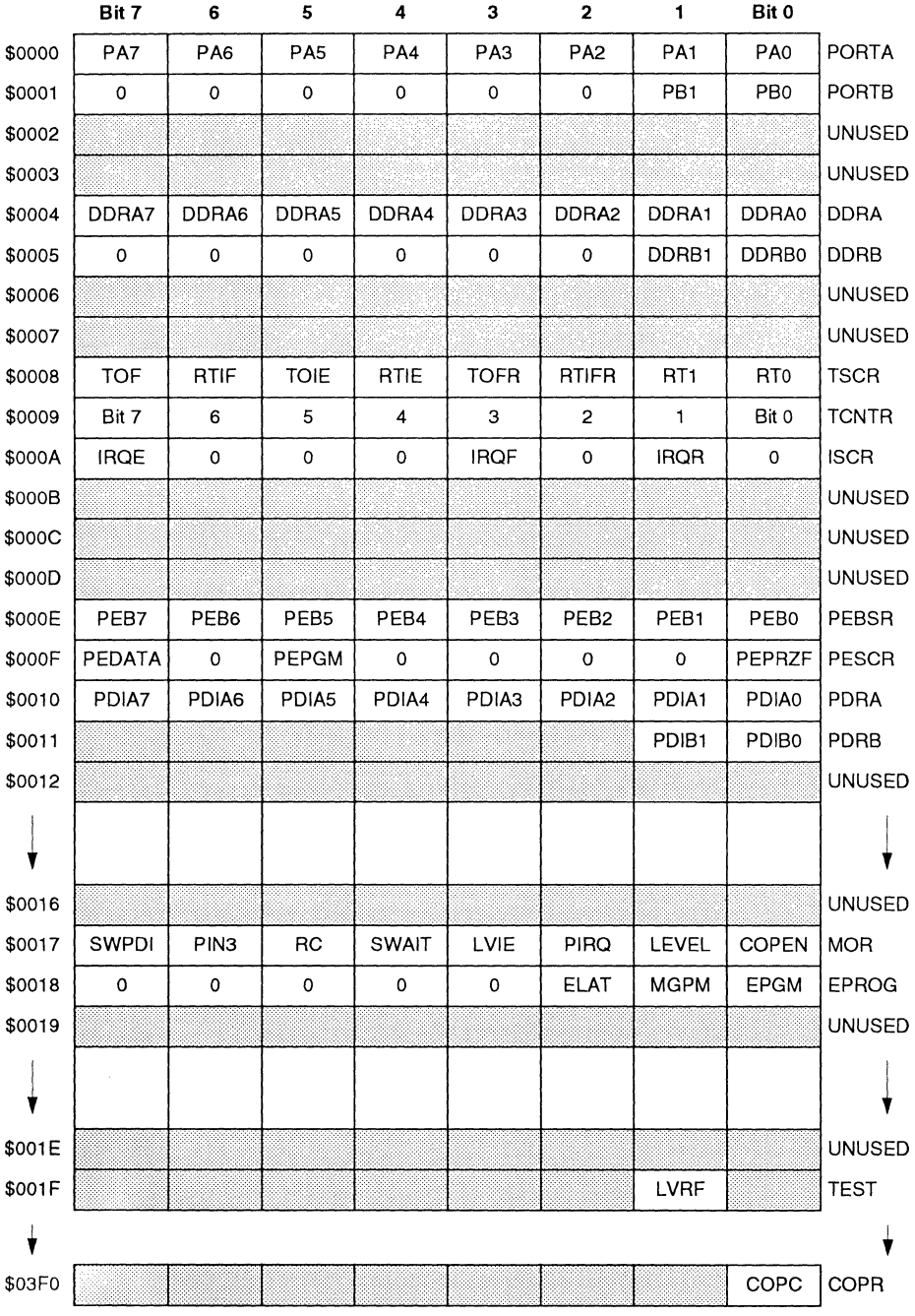


Figure 2-1. Memory Map



	Bit 7	6	5	4	3	2	1	Bit 0	
$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA
$0001	0	0	0	0	0	0	PB1	PB0	PORTB
$0002									UNUSED
$0003									UNUSED
$0004	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0	DDRA
$0005	0	0	0	0	0	0	DDRB1	DDRB0	DDRB
$0006									UNUSED
$0007									UNUSED
$0008	TOF	RTIF	TOIE	RTIE	TOFR	RTIFR	RT1	RT0	TSCR
$0009	Bit 7	6	5	4	3	2	1	Bit 0	TCNTR
$000A	IRQE	0	0	0	IRQF	0	IRQR	0	ISCR
$000B									UNUSED
$000C									UNUSED
$000D									UNUSED
$000E	PEB7	PEB6	PEB5	PEB4	PEB3	PEB2	PEB1	PEB0	PEBSR
$000F	PEDATA	0	PEPGM	0	0	0	0	PEPRZF	PESCR
$0010	PDIA7	PDIA6	PDIA5	PDIA4	PDIA3	PDIA2	PDIA1	PDIA0	PDRA
$0011							PDIB1	PDIB0	PDRB
$0012									UNUSED
$0016									UNUSED
$0017	SWPDI	PIN3	RC	SWAIT	LVIE	PIRQ	LEVEL	COPEN	MOR
$0018	0	0	0	0	0	ELAT	MGPM	EPGM	EPROG
$0019									UNUSED
$001E									UNUSED
$001F							LVRF		TEST
$03F0								COPC	COPR

Figure 2-2. I/O Registers

2.4 EPROM/OTPROM

An MCU with a quartz window has 504 bytes of erasable, programmable ROM (EPROM). The quartz window allows EPROM erasure with ultraviolet light. In an MCU without the quartz window, the EPROM cannot be erased and serves as 504 bytes of one-time programmable ROM (OTPROM). Addresses $0020–$03EF contain 496 bytes of user EPROM/OTPROM. The eight addresses from $03F8 to $03FF are EPROM/OTPROM locations reserved for interrupt vectors and reset vectors.

2.5 Personality EPROM/OTPROM

An MCU with a quartz window has a 64-bit array of erasable, programmable ROM (EPROM) to serve as a personality EPROM. The quartz window allows EPROM erasure with ultraviolet light. In an MCU without the quartz window, the personality EPROM cannot be erased and serves as a 64-bit array of one-time programmable ROM (OTPROM).

SECTION 3
CENTRAL PROCESSOR UNIT (CPU)

This section describes the CPU registers.

3.1 CPU Registers

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

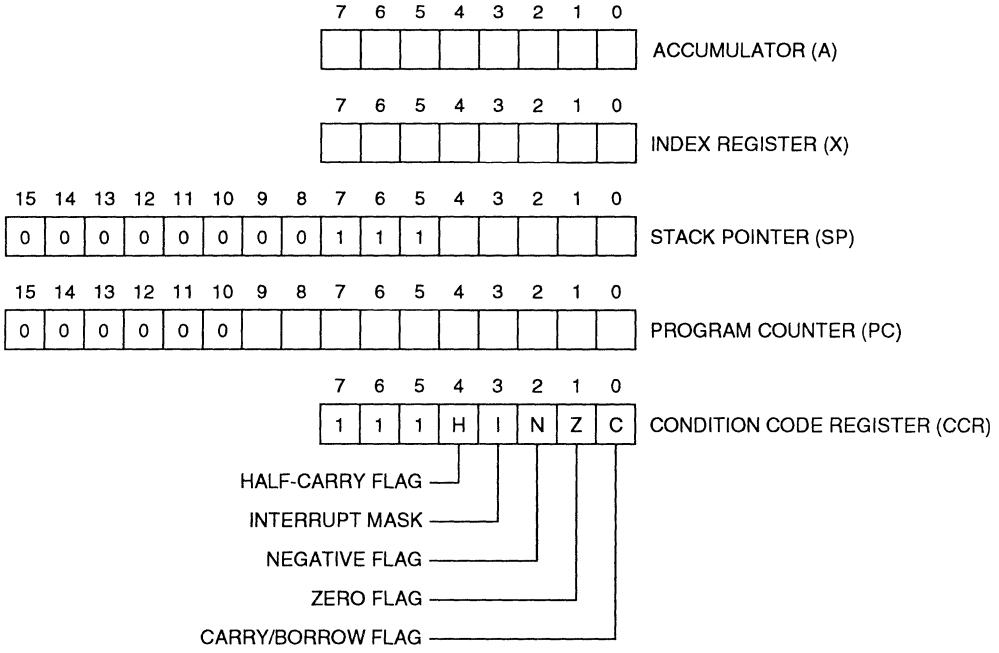


Figure 3-1. Programming Model

3.1.1 Accumulator (A)

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

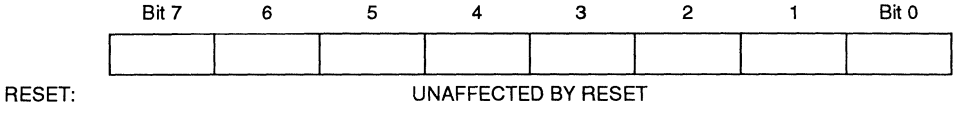


Figure 3-2. Accumulator (A)

3.1.2 Index Register (X)

In the indexed addressing modes, the CPU uses the byte in the index register to determine the conditional address of the operand. (See **11.1.5 Indexed, No Offset**, **11.1.6 Indexed, 8-Bit Offset**, and **11.1.7 Indexed, 16-Bit Offset**.)

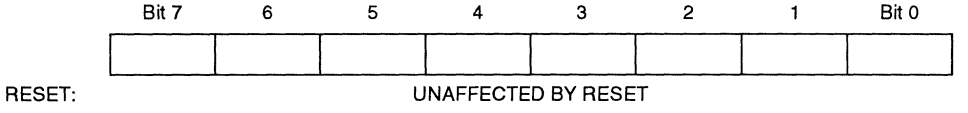


Figure 3-3. Index Register (X)

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

3.1.3 Stack Pointer (SP)

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 initializes 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.

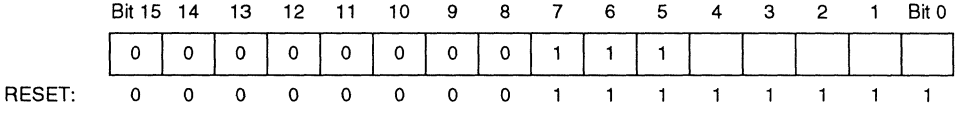


Figure 3-4. Stack Pointer (SP)

The eleven most significant bits of the stack pointer are permanently fixed at 00000000111, so the stack pointer produces addresses from $00E0 to $00FF. If subroutines and interrupts use more than 32 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.

3.1.4 Program Counter (PC)

The program counter is a 16-bit register that contains the address of the next instruction or operand to be fetched. The six most significant bits of the program counter are ignored internally and appear as 000000.

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	0										
RESET:	0	0	0	0	0	0	LOADED WITH VECTOR FROM $03FE AND $03FF									

Figure 3-5. Program Counter (PC)

3.1.5 Condition Code Register (CCR)

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 prior instructions. 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 (CCR)

3.1.5.1 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. Reset has no effect on the half-carry flag.

3.1.5.2 Interrupt Mask (I)

Setting the interrupt mask disables interrupts. If an interrupt request occurs while the interrupt mask is a 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. 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 a reset, the interrupt mask is set and can be cleared only by a CLI instruction.

3.1.5.3 Negative Flag (N)

The CPU sets the negative flag when an arithmetic operation, logical operation, or data manipulation produces a negative result. Reset has no effect on the negative flag.

3.1.5.4 Zero Flag (Z)

The CPU sets the zero flag when an arithmetic operation, logical operation, or data manipulation produces a result of $00. Reset has no effect on the zero flag.

3.1.5.5 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. Reset has no effect on the carry/borrow flag.

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

This section describes how interrupts temporarily change the normal processing sequence.

4.1 Interrupt Types

The following conditions generate interrupts:

- SWI instruction (software interrupt)
- A logic zero applied to the $\overline{IRQ}/V_{PP}$ pin (external interrupt)
- A logic one applied to one of the PA3–PA0 pins when port A external interrupts are enabled (external interrupt)
- A timer overflow (timer interrupt)
- Expiration of the real-time interrupt period (timer interrupt)

An interrupt temporarily stops normal program execution to process a particular event. An interrupt does not stop the execution of the instruction in progress, but takes effect when the current instruction completes its execution. Interrupt processing automatically saves the CPU registers on the stack and loads the program counter with a user-defined vector address.

4.1.1 Software Interrupt

The software interrupt (SWI) instruction causes a nonmaskable interrupt.

4.1.2 External Interrupts

The following sources can generate external interrupts:

- $\overline{IRQ}/V_{PP}$ pin
- PA3–PA0 pins when port A external interrupts are enabled

Setting the I bit in the condition code register or clearing the IRQE bit in the interrupt status and control register disables external interrupts.

4.1.2.1 $\overline{IRQ}/V_{PP}$ Pin

An interrupt signal on the $\overline{IRQ}/V_{PP}$ pin latches an external interrupt request. After completing the current instruction, the CPU tests the following bits:

- The IRQ latch
- The IRQE bit in the interrupt status and control register
- The I bit in the condition code register

If both the IRQ latch and the IRQE bit are set, and the I bit is clear, the CPU then begins the interrupt sequence. The CPU clears the IRQ latch while it fetches the interrupt vector, so that another external interrupt request can be latched during the interrupt service routine. As soon as the I bit is cleared during the return from interrupt, the CPU can recognize the new interrupt request. Figure 4-1 shows the logic for external interrupts.

The $\overline{IRQ}/V_{PP}$ pin is negative edge-triggered only or negative edge- and low-level-triggered, depending on the state of the LEVEL bit in the mask option register. See **9.4 Mask Option Register (MOR)**.

Programming the LEVEL bit to a logic one selects the edge- and level-sensitive trigger option. When LEVEL = 1:

- A falling edge or a low level on the $\overline{IRQ}/V_{PP}$ pin latches an external interrupt request.
- As long as the $\overline{IRQ}/V_{PP}$ is low, an external interrupt request is present, and the CPU continues to execute the interrupt service routine. The edge- and level-sensitive trigger option allows connection to the $\overline{IRQ}/V_{PP}$ pin of multiple wired-OR interrupt sources.

Programming the LEVEL bit to a logic zero selects the edge-sensitive-only trigger option. When LEVEL = 0:

- A falling edge on the $\overline{IRQ}/V_{PP}$ pin latches an external interrupt request.
- A subsequent interrupt request can be latched only after the voltage level on the $\overline{IRQ}/V_{PP}$ pin returns to logic one and then falls again to logic zero.

NOTE

If the $\overline{IRQ}/V_{PP}$ pin is not in use, connect it to the V_{DD} pin.

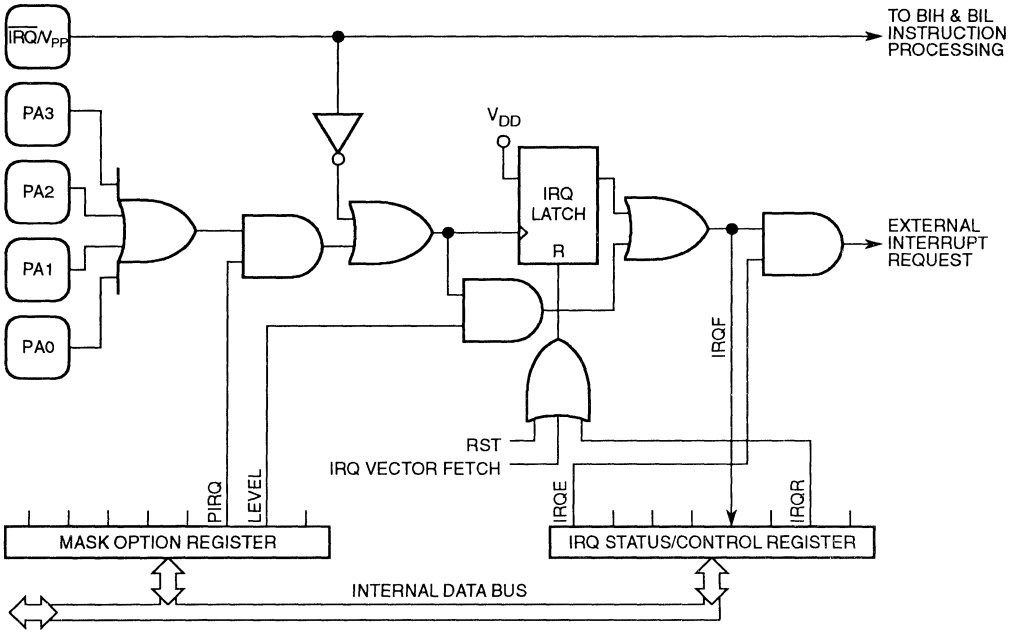


Figure 4-1. External Interrupt Logic

4.1.2.2 PA3–PA0 Pins

Programming the PIRQ bit in the mask option register to a logic one enables pins PA3–PA0 to serve as additional external interrupt sources. See **9.4 Mask Option Register (MOR)**. An interrupt signal on a PA3–PA0 pin latches an external interrupt request. After completing the current instruction, the CPU tests the following bits:

- The IRQ latch
- The IRQE bit in the IRQ status and control register
- The I bit in the condition code register.

If both the IRQ latch and the IRQE bit are set, and the I bit is clear, the CPU then begins the interrupt sequence. The CPU clears the IRQ latch while it fetches the interrupt vector, so that another external interrupt request can be latched during the interrupt service routine. As soon as the I bit is cleared during the return from interrupt, the CPU can recognize the new interrupt request.

The PA3–PA0 pins are edge-triggered only or both edge- and level-triggered, depending on the state of the LEVEL bit in the MOR.

Programming the LEVEL bit to a logic one selects the edge- and level-sensitive trigger option. When LEVEL = 1:

- A rising edge or a high level on a PA3–PA0 pin latches an external interrupt request if and only if all other PA3–PA0 pins are low and the $\overline{IRQ}/V_{PP}$ pin is high.
- A falling edge or a low level on the $\overline{IRQ}/V_{PP}$ pin latches an external interrupt request if and only if all of the PA3–PA0 pins are low.
- As long as any PA3–PA0 pin is high or the $\overline{IRQ}/V_{PP}$ pin is low, an external interrupt request is present, and the CPU continues to execute the interrupt service routine.

Programming the LEVEL bit to a logic zero selects the edge-sensitive only trigger option. When LEVEL = 0:

- A rising edge on a PA3–PA0 pin latches an external interrupt request if and only if all other PA3–PA0 pins are low and the $\overline{IRQ}/V_{PP}$ pin is high.
- A falling edge on the $\overline{IRQ}/V_{PP}$ pin latches an external interrupt request if and only if all of the PA3–PA0 pins are low.
- A subsequent PA3–PA0 pin interrupt request can be latched only after the voltage level of the previous PA3–PA0 interrupt signal returns to a logic zero and then rises again to a logic one.
- A subsequent $\overline{IRQ}/V_{PP}$ pin interrupt request can be latched only after the voltage level of the previous $\overline{IRQ}/V_{PP}$ interrupt signal returns to a logic one and then falls again to a logic zero.

4.1.2.3 IRQ Status and Control Register (ISCR)

The IRQ status and control register contains an external interrupt mask, an external interrupt flag, and a flag reset bit. Unused bits read as logic zeros.

ISCR — IRQ Status and Control Register **$000A**

	Bit 7	6	5	4	3	2	1	Bit 0
	IRQE	0	0	0	IRQF	0	IRQR	0
RESET:	1	0	0	0	0	0	—	0

Figure 4-2. IRQ Status and Control Register (ISCR)

IRQE — External Interrupt Request Enable

This read/write bit enables external interrupts. Reset sets the IRQE bit.

1 = External interrupt processing enabled
0 = External interrupt processing disabled

IRQF — External Interrupt Request Flag

The IRQF bit is a clearable, read-only flag that is set when an external interrupt request is pending. Reset clears the IRQF bit.

1 = Interrupt request pending
0 = No interrupt request pending

The following conditions set the IRQF bit:

- An external interrupt signal on the $\overline{IRQ}/V_{PP}$ pin
- An external interrupt signal on pin PA3, PA2, PA1, or PA0 when PA3–PA0 are enabled to serve as external interrupt sources

The CPU clears the IRQF bit when fetching the interrupt vector. Writing to the IRQF bit has no effect. Writing a logic one to the IRQR bit clears the IRQF bit.

IRQR — Interrupt Request Reset

Writing a logic one to this write-only bit clears the IRQF bit. Writing a logic zero to IRQR has no effect. Reset has no effect on IRQR.

1 = IRQF bit cleared
0 = No effect

4.1.3 Timer Interrupts

The multifunction timer can generate the following interrupts:

- Timer overflow interrupt
- Real-time interrupt

Setting the I bit in the condition code register disables all timer interrupts.

4.1.3.1 Timer Overflow Interrupt

A timer overflow interrupt request occurs if the timer overflow flag, TOF, becomes set while the timer overflow interrupt enable bit, TOIE, is also set. See **8.1 Timer Status and Control Register (TSCR)**.

4.1.3.2 Real-Time Interrupt

A real-time interrupt request occurs if the real-time interrupt flag, RTIF, becomes

set while the real-time interrupt enable bit, RTIE, is also set. See **8.1 Timer Status and Control Register (TSCR)**.

4.2 Interrupt Processing

The CPU does the following things to begin servicing an interrupt:

- Stores the CPU registers on the stack in the order shown in Figure 4-3
- Sets the I bit in the condition code register to prevent further interrupts
- Loads the program counter with the contents of the appropriate interrupt vector locations:
  - $03FC and $03FD (software interrupt vector)
  - $03FA and $03FB (external interrupt vector)
  - $03F8 and $03F9 (timer interrupt vector)

The return from interrupt (RTI) instruction causes the CPU to recover the CPU registers from the stack as shown in Figure 4-3.

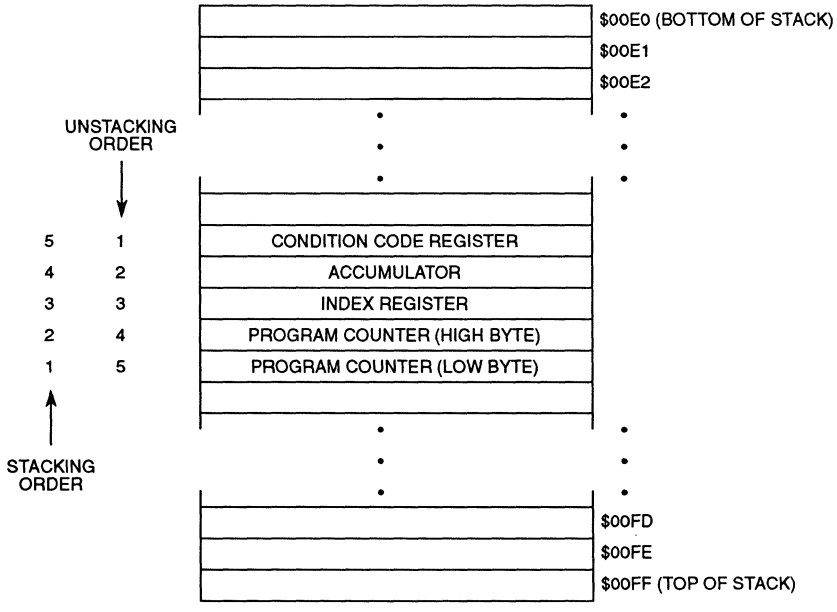


Figure 4-3. Interrupt Stacking Order

Table 4-1 summarizes the reset and interrupt sources and vector assignments.

Table 4-1. Reset/Interrupt Vector Addresses

Function	Source	Local Mask	Global Mask	MOR Control Bit	Priority (1 = Highest)	Vector Address
Reset	Power-On Logic	None	None	None	1	$03FE–$03FF
	$\overline{\text{RESET}}$ Pin			None		
	COP Watchdog			COPEN[1]		
	Low Voltage Detect			LVIE[2]		
	Illegal Address Logic			None		
Software Interrupt (SWI)	User Code	None	None	None	Same Priority As Instruction	$03FC–$03FD
External Interrupts	$\overline{\text{IRQ}}/V_{PP}$ Pin	IRQE Bit	I Bit	None	2	$03FA–$03FB
	PA3 Pin			PIRQ[3]		
	PA2 Pin			PIRQ[3]		
	PA1 Pin			PIRQ[3]		
	PA0 Pin			PIRQ[3]		
Timer Interrupts	TOF Bit	TOFE Bit	I Bit	None	3	$03F8–$03F9
	RTIF Bit	RTIE Bit				

1. COPEN enables the COP watchdog.
2. LVIE enables low-voltage resets.
3. PIRQ enables port A external interrupts.

NOTE

If more than one interrupt request is pending, the CPU fetches the vector of the higher priority interrupt first. A higher priority interrupt does not interrupt a lower priority interrupt service routine unless the lower priority interrupt service routine clears the I bit.

Figure 4-4 shows the sequence of events caused by an interrupt.

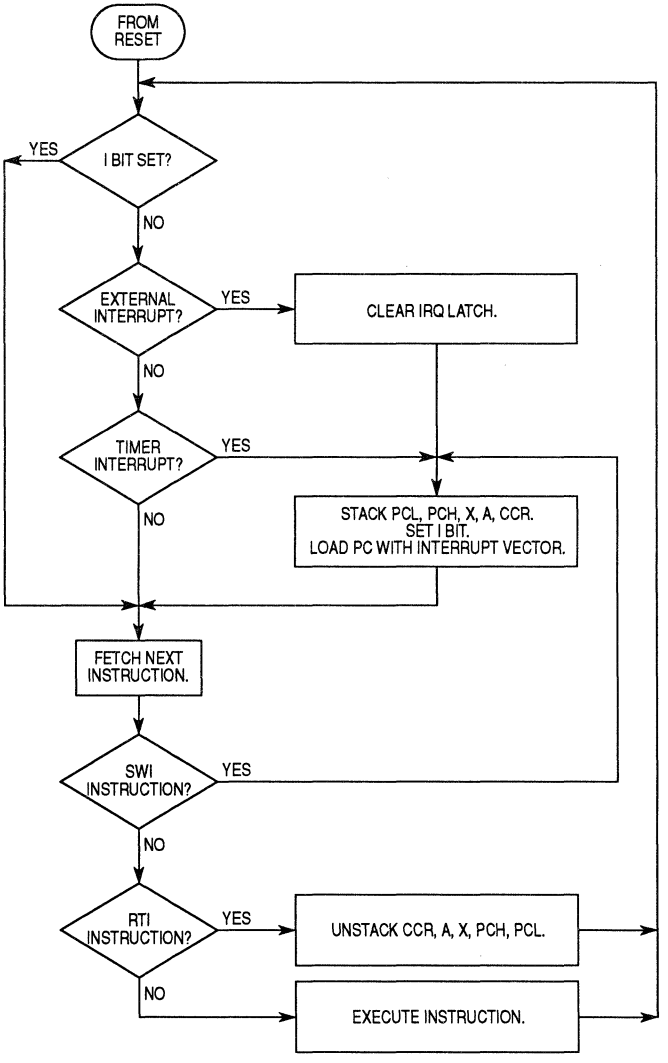


Figure 4-4. Interrupt Flowchart

SECTION 5
RESETS

This section describes the five reset sources and how they initialize the MCU.

5.1 Reset Types

A reset immediately stops the operation of the instruction being executed, initializes certain control bits, and loads the program counter with a user-defined reset vector address. The following conditions produce a reset:

- Initial power-up (power-on reset)
- A logic zero applied to the $\overline{\text{RESET}}$ pin (external reset)
- Timeout of the COP watchdog (COP reset)
- An opcode fetch from an address not in the memory map (illegal address reset)
- V_{DD} voltage below nominal 3.5 V (low voltage reset)

5.1.1 Power-On Reset

A positive transition on the V_{DD} pin generates a power-on reset. The power-on reset is strictly for power-up conditions and cannot be used to detect drops in power supply voltage.

A 4064 t_{CYC} (internal clock cycle) delay after the oscillator becomes active allows the clock generator to stabilize. If the $\overline{\text{RESET}}$ pin is at logic zero at the end of 4064 t_{CYC}, the MCU remains in the reset condition until the signal on the $\overline{\text{RESET}}$ pin goes to logic one.

5.1.2 External Reset

A logic zero applied to the $\overline{\text{RESET}}$ pin for one and one-half t_{CYC} generates an external reset. A Schmitt trigger senses the logic level at the $\overline{\text{RESET}}$ pin.

A COP reset or an illegal address reset pulls the $\overline{\text{RESET}}$ pin low for one internal clock cycle. A low voltage reset pulls the $\overline{\text{RESET}}$ pin low for as long as the low voltage condition exists.

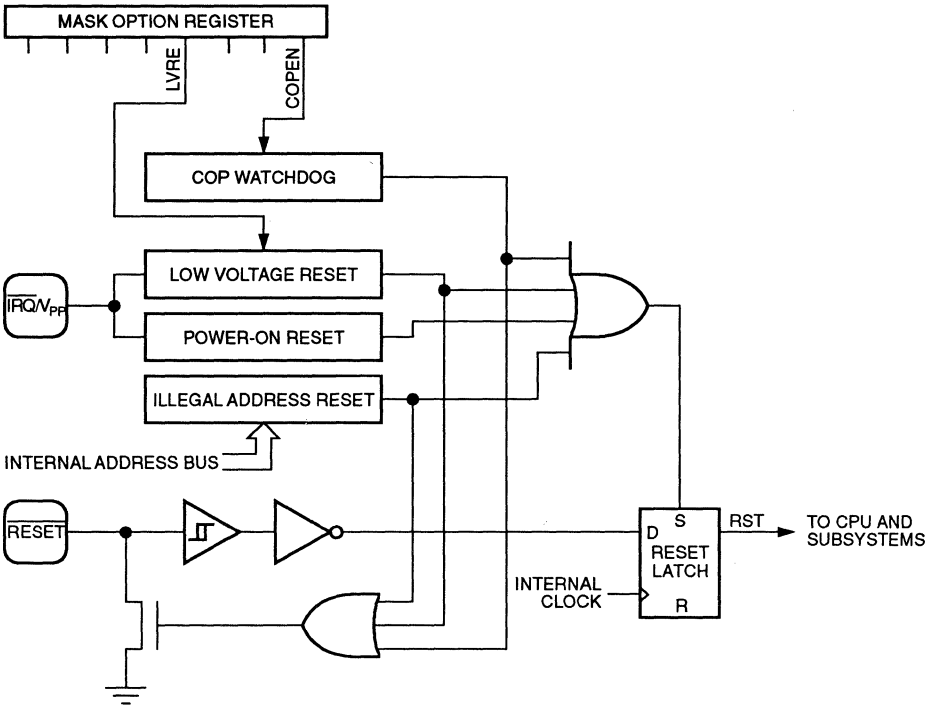


Figure 5-1. Reset Sources

NOTE

To avoid overloading some power supply designs, do not connect the $\overline{\text{RESET}}$ pin directly to V_{DD}. Use a pullup resistor of 10 kΩ or more.

5.1.3 Computer Operating Properly (COP) Reset

A timeout of the COP watchdog generates a COP reset. The COP watchdog is part of a software error detection system and must be cleared periodically to start a new timeout period. (See **8.3 COP Watchdog**.) To clear the COP watchdog and prevent a COP reset, write a logic zero to bit 0 (COPC) of the COP register at location $03F0. The COP register is a write-only register that returns the contents of a ROM location when read.

COPR — COP Register **$03F0**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	—	—	—	—	—	COPC
RESET:	—	—	—	—	—	—	—	0

Figure 5-2. COP Register (COPR)

COPC — COP Clear

COPC is a write-only bit. Periodically writing a logic zero to COPC prevents the COP watchdog from resetting the MCU. Writing a logic one has no effect. Reset clears the COPC bit.

5.1.4 Illegal Address Reset

An opcode fetch from an address that is not in the EPROM (locations $0200–$03FF) or the RAM (locations $00E0–$00FF) generates an illegal address reset. An illegal address reset pulls the $\overline{\text{RESET}}$ pin low for one cycle of the internal clock.

5.1.5 Low Voltage Reset

The low voltage reset circuit generates a reset signal if the voltage on the V_{DD} pin falls below 3.5 V (nominal). V_{DD} must be set at 5 V ±10% while the low voltage reset circuit is enabled.

Programming the LVRE bit to a logic one enables the low voltage reset function. When erased, the LVRE bit in the mask option register disables the low voltage reset circuit. See **9.4 Mask Option Register (MOR)**.

A low voltage reset pulls the $\overline{\text{RESET}}$ pin low for as long as the low voltage condition exists.

The state of the low voltage reset circuit is readable in the test register at location $001F. Bit 1 of the test register is the low-voltage reset flag (LVRF). Regardless of the LVRE bit in the mask option register, the low voltage reset circuit is active in all modes except Stop Mode.

5.2 Reset States

The following paragraphs describe how resets initialize the MCU.

5.2.1 CPU

A reset has the following effects on the CPU:

- Loads the stack pointer with $FF
- Sets the I bit in the condition code register, inhibiting interrupts
- Sets the IRQE bit in the interrupt status and control register
- Loads the program counter with the user-defined reset vector from locations $03FE and $03FF
- Clears the stop latch, enabling the CPU clock
- Clears the wait latch, waking the CPU from the Wait Mode

5.2.2 I/O Port Registers

A reset has the following effects on I/O port registers:

- Clears bits DDRA7–DDRA0 in data direction register A so that port A pins are inputs
- Clears bits PDIA7–PDIA0 in pulldown register A so that port A pulldown devices are enabled (if the SWPDI bit in the mask option register is programmed to a logic zero)
- Clears bits DDRB1 and DDRB0 in data direction register B so that port B pins are inputs (if the SWPDI bit in the mask option register is programmed to logic zero)
- Clears bits PDIB1 and PDIB0 in pulldown register B so that port B pulldown devices are enabled
- Has no effect on port A or port B data registers

5.2.3 Multifunction Timer

A reset has the following effects on the multifunction timer:

- Clears the timer status and control register
- Clears the timer counter register

5.2.4 COP Watchdog

A reset clears the COP watchdog timeout counter.

SECTION 6
LOW POWER MODES

This section describes the four low power modes:

- Stop Mode
- Wait Mode
- Halt Mode
- Data Retention Mode

6.1 Stop Mode

If the SWAIT bit in the mask option register is programmed to a logic zero, the STOP instruction puts the MCU in its lowest power-consumption mode and has the following effects on the MCU:

- Clears TOF and RTIF, the timer interrupt flags in the timer status and control register, removing any pending timer interrupts
- Clears TOIE and RTIE, the timer interrupt enable bits in the timer status and control register, disabling further timer interrupts
- Clears the multifunction timer counter register
- Sets the IRQE bit in the IRQ status and control register to enable external interrupts
- Clears the I bit in the condition code register, enabling interrupts
- Stops the internal oscillator, turning off the CPU clock and the timer clock, including the COP watchdog

The STOP instruction does not affect any other registers or any I/O lines.

The following conditions bring the MCU out of Stop Mode:

- An external interrupt signal on the $\overline{IRQ}/V_{PP}$ pin — A high-to-low transition on the $\overline{IRQ}/V_{PP}$ pin loads the program counter with the contents of locations $03FA and $03FB.

- An external interrupt signal on a port A external interrupt pin — If the PIRQ bit in the mask option register is programmed to a logic one, a low-to-high transition on a PA3–PA0 pin loads the program counter with the contents of locations $03FA and $03FB.

- External reset — A logic zero on the $\overline{\text{RESET}}$ pin resets the MCU and loads the program counter with the contents of locations $03FE and $03FF.

When the MCU exits Stop Mode, processing resumes after a stabilization delay of 4064 oscillator cycles.

6.2 Wait Mode

The WAIT instruction puts the MCU in an intermediate power-consumption mode and has the following effects on the MCU:

- Clears the I bit in the condition code register, enabling interrupts

- Sets the IRQE bit in the IRQ status and control register, enabling external interrupts

- Stops the CPU clock, but allows the internal oscillator and timer clock to continue to run

The WAIT instruction does not affect any other registers or any I/O lines.

The following conditions restart the CPU clock and bring the MCU out of Wait Mode:

- An external interrupt signal on the $\overline{\text{IRQ}}/V_{PP}$ pin — A high-to-low transition on the $\overline{\text{IRQ}}/V_{PP}$ pin loads the program counter with the contents of locations $03FA and $03FB.

- An external interrupt signal on a port A external interrupt pin — If the PIRQ bit in the mask option register is programmed to a logic one, a low-to-high transition on a PA3–PA0 pin loads the program counter with the contents of locations $03FA and $03FB.

- A timer interrupt — A timer overflow or a real-time interrupt request loads the program counter with the contents of locations $03F8 and $03F9.

- A COP watchdog reset — A timeout of the COP watchdog resets the MCU and loads the program counter with the contents of locations $03FE and $03FF. Software can enable real-time interrupts so that the MCU can periodically exit Wait Mode to reset the COP watchdog.

- External reset — A logic zero on the $\overline{\text{RESET}}$ pin resets the MCU and loads the program counter with the contents of locations $03FE and $03FF.

6.3 Halt Mode

The STOP instruction puts the MCU in Halt Mode if the SWAIT bit in the mask option register is programmed to a logic one. Halt Mode is identical to Wait Mode, except that a recovery delay of 1–4064 internal clock cycles occurs when the MCU exits Halt Mode. When the SWAIT bit is set, the COP watchdog cannot be inadvertently turned off by a STOP instruction.

Figure 6-1 shows the sequence of events in Stop, Wait, and Halt Modes.

6.4 Data Retention Mode

In Data Retention Mode, the MCU retains RAM contents and CPU register contents at V_{DD} voltages as low as 2.0 Vdc. Data Retention Mode allows the MCU to remain in a low power-consumption state during which it retains data, but the CPU cannot execute instructions.

To put the MCU in Data Retention Mode:

1. Drive the $\overline{\text{RESET}}$ pin to a logic zero.
2. Lower the V_{DD} voltage. The $\overline{\text{RESET}}$ pin must remain low continuously during Data Retention Mode.

To take the MCU out of Data Retention Mode:

1. Return V_{DD} to normal operating voltage.
2. Return the $\overline{\text{RESET}}$ pin to a logic one.

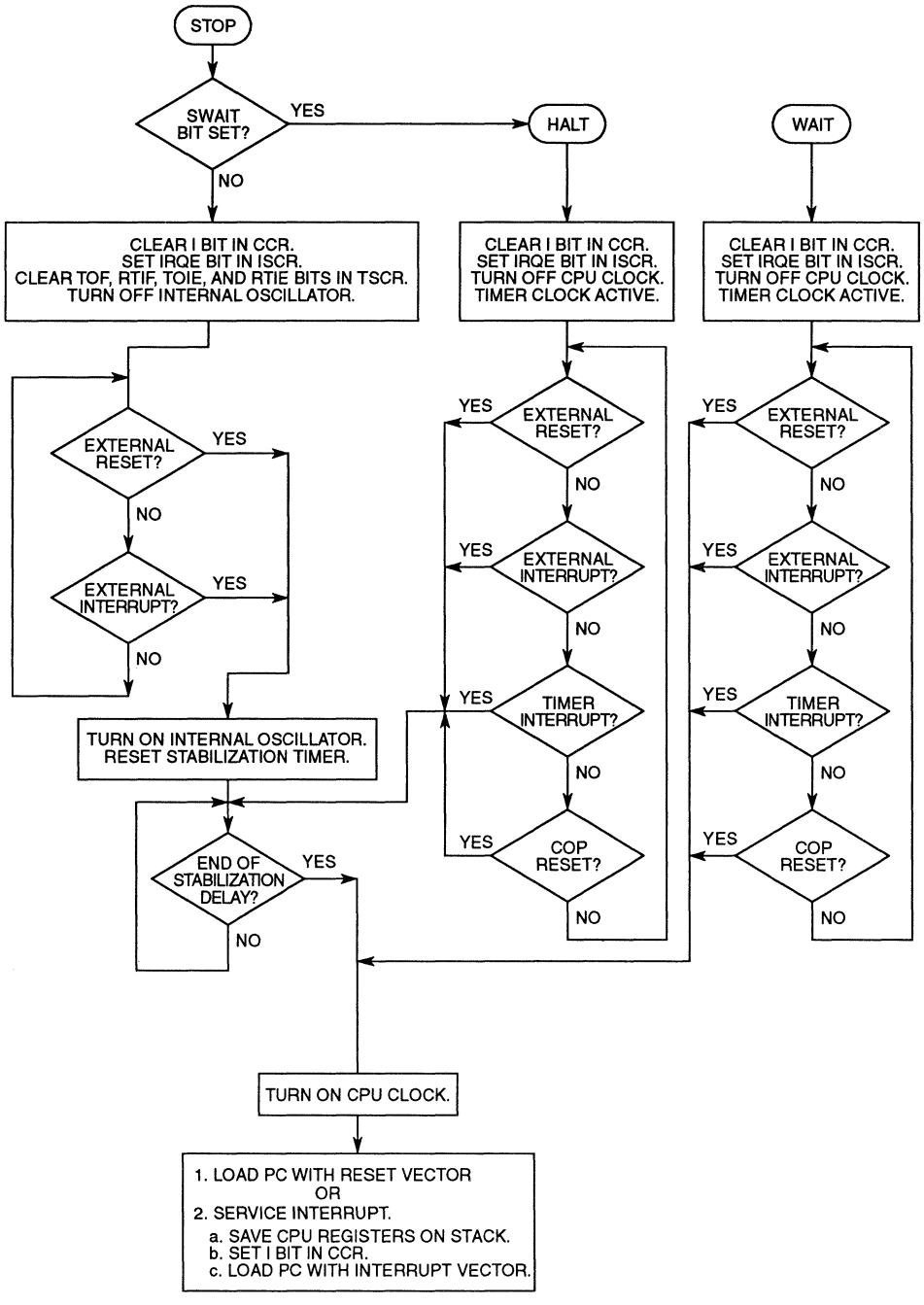


Figure 6-1. STOP/WAIT/HALT Flowchart

SECTION 7
PARALLEL I/O

This section describes the two bidirectional I/O ports.

7.1 I/O Port Function

The ten bidirectional I/O pins form two parallel I/O ports. Each I/O pin is programmable as an input or an output. The contents of the data direction registers determine the data direction of each I/O pin.

All ten I/O pins have software-programmable pulldown devices.

7.2 Port A

Port A is an 8-bit general-purpose bidirectional I/O port with the following features:

- Programmable pulldown devices
- 8 mA current sinking capability (pins PA7–PA4)
- External interrupt capability (pins PA3–PA0)

7.2.1 Port A Data Register (PORTA)

The port A data register contains a bit for each of the port A pins. When a port A pin is programmed to be an output, the state of its data register bit determines the state of the output pin. When a port A pin is programmed to be an input, reading the port A data register returns the logic state of the pin.

PORTA — Port A Data Register **$0000**

	Bit 7	6	5	4	3	2	1	Bit 0
	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
RESET:	UNAFFECTED BY RESET							

Figure 7-1. Port A Data Register (PORTA)

PA7–PA0 — Port A Data Bits

These read/write bits are software-programmable. Data direction of each bit is under the control of the corresponding bit in data direction register A. Reset has no effect on port A data.

7.2.2 Data Direction Register A (DDRA)

The contents of data direction register A determine whether each port A pin is an input or an output. Writing a logic one to a DDRA bit enables the output buffer for the associated port A pin; a logic zero disables the output buffer. A reset initializes all DDRA bits to logic zeros, configuring all port A pins as inputs. If the pulldown devices are enabled, setting a DDRA bit to a logic one turns off the pulldown device for that pin.

DDRA — Data Direction Register A **$0004**

	Bit 7	6	5	4	3	2	1	Bit 0
	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
RESET:	0	0	0	0	0	0	0	0

Figure 7-2. Data Direction Register A (DDRA)

DDRA7–DDRA0 — Port A Data Direction Bits

These read/write bits control port A data direction. Reset clears bits DDRA7–DDRA0.

1 = Corresponding port A pin configured as output
0 = Corresponding port A pin configured as input

NOTE

Avoid glitches on port A pins by writing to the port A data register before changing DDRA bits from logic zero to logic one.

7.2.3 Pulldown Register A (PDRA)

Programming the SWPDI bit in the mask option register to a logic zero enables the port A and port B pulldown devices. The port A pulldown devices sink approximately 100 µA and are under the control of the PDIA7–PDIA0 bits in pulldown register A. Clearing the PDIA7–PDIA0 bits turns on the pulldown devices of the port A pins that are configured as inputs. A pulldown device can be turned on only when its pin is an input. When SWPDI is a logic zero, reset initializes all port A pins as inputs with pulldown devices turned on.

Programming the SWPDI bit to a logic one disables the port A and port B pulldown devices. Reset initializes all port A pins as inputs with pulldown devices disabled when the SWPDI bit is programmed to a logic one.

PDRA — Pulldown Register A **$0010**

	Bit 7	6	5	4	3	2	1	Bit 0
	PDIA7	PDIA6	PDIA5	PDIA4	PDIA3	PDIA2	PDIA1	PDIA0
RESET:	0	0	0	0	0	0	0	0

Figure 7-3. Pulldown Register A (PDRA)

PDIA7–PDIA0 — Port A Pulldown Inhibit Bits 7–0

Writing logic zeros to these write-only bits turns on the port A pulldown devices. Reading pulldown register A returns undefined data. Reset clears bits PDIA7–PDIA0.

1 = Corresponding port A pin pulldown device turned off
0 = Corresponding port A pin pulldown device turned on

NOTE

Avoid a floating port A input by clearing its pulldown register bit before changing its DDRA bit from logic one to logic zero.

NOTE

Do not use read-modify-write instructions on pulldown register A.

7.2.4 Port A External Interrupts

Programming the PIRQ bit in the mask option register to a logic one enables the PA3–PA0 pins to serve as external interrupt pins in addition to the $\overline{IRQ}/V_{PP}$ pin. The active interrupt state for the PA3–PA0 pins is a logic one or a rising edge. The active interrupt state for the $\overline{IRQ}/V_{PP}$ pin is a logic zero or a falling edge. The state of the LEVEL bit in the mask option register determines whether external interrupt inputs are edge-sensitive only or both edge- and level-sensitive.

NOTE

When testing for external interrupts, the BIH and BIL instructions test the voltage on the $\overline{IRQ}/V_{PP}$ pin, not the state of the internal IRQ signal. Therefore, BIH and BIL cannot test the port A external interrupt pins.

7.2.5 Port A Logic

Figure 7-4 shows the port A I/O logic.

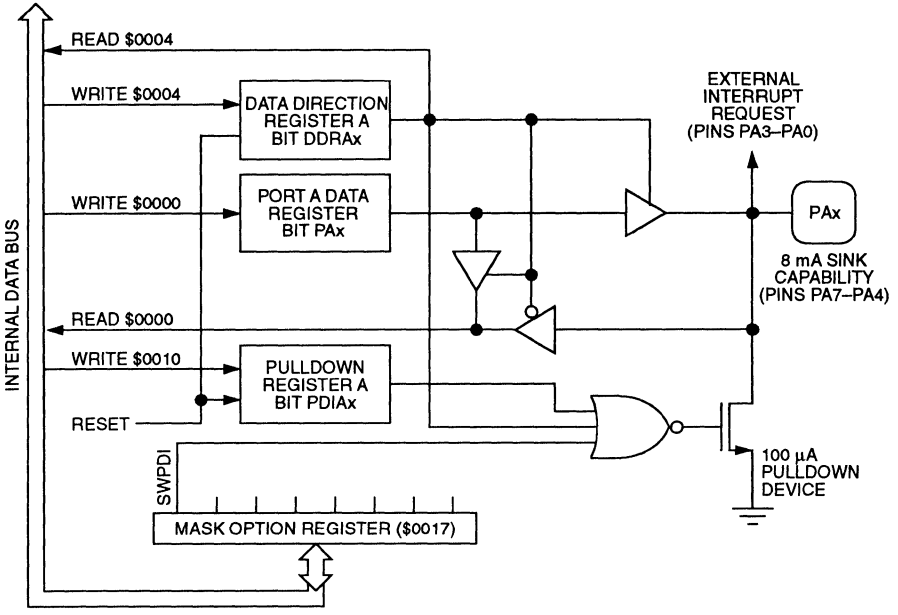


Figure 7-4. Port A I/O Circuit

When a port A pin is programmed as an output, reading the port bit actually reads the value of the data latch and not the voltage on the pin itself. When a port A pin is programmed as an input, reading the port bit reads the voltage level on the pin. The data latch can always be written, regardless of the state of its DDR bit. Table 7-1 summarizes the operations of the port A pins.

Table 7-1. Port A Pin Functions

Pulldown Mask Option	Control Bits		I/O Pin Mode	Accesses to PDRA		Accesses to DDRA	Accesses to PORTA	
	PDIAx	DDRAx		Read	Write	Read/Write	Read	Write
No	X(1)	0	Input, Hi-Z	U(2)	PDIA7-0	DDRA7-0	Pin	PA7-0
No	X	1	Output	U	PDIA7-0	DDRA7-0	PA7-0	PA7-0
Yes	0	0	Input, Pulldown On	U	PDIA7-0	DDRA7-0	Pin	PA7-0
Yes	0	1	Output	U	PDIA7-0	DDRA7-0	PA7-0	PA7-0
Yes	1	0	Input, Hi-Z	U	PDIA7-0	DDRA7-0	Pin	PA7-0
Yes	1	1	Output	U	PDIA7-0	DDRA7-0	PA7-0	PA7-0

1. X = Don't care
2. U = Undefined

7.3 Port B

Port B is a 2-bit general-purpose bidirectional I/O port with the following features:

- Programmable pulldown devices
- Oscillator output for three-pin RC oscillator configuration

7.3.1 Port B Data Register (PORTB)

The port B data register contains a bit for each of the port B pins. When a port B pin is programmed to be an output, the state of its data register bit determines the state of the output pin. When a port B pin is programmed to be an input, reading the port B data register returns the logic state of the pin. Reset has no effect on port B data.

PORTB — Port B Data Register **$0001**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	0	0	0	0	PB1/OSC3	PB0
RESET:	UNAFFECTED BY RESET							
ALTERNATE FUNCTION:							OSC3	

Figure 7-5. Port B Data Register (PORTB)

PB1/OSC3 — Port B Data Bit 1/Oscillator Output

This read/write data bit is software programmable. Data direction of PB1 bit is under the control of the DDRB1 bit in data direction register B.

When both the RC and PIN3 bits in the mask option register are set, PB1/OSC3 can be used as an oscillator output in the three-pin RC oscillator configuration. Using PB1/OSC3 as an oscillator output affects port B in the following ways:

- Bit PB1 can be used as a read/write storage location without affecting the oscillator. Reset has no effect on bit PB1.
- Bit DDRB1 in data direction register B can be used as a read/write storage location without affecting the oscillator. Reset clears DDRB1.
- The PB1/OSC3 pulldown device is disabled, regardless of the state of the SWPDI bit in the mask option register.

PB0 — Port B Data Bit 0

This read/write data bit is software programmable. Data direction of PB0 is under the control of the DDRB0 bit in data direction register B.

Bits 7–2 — Not used

Bits 7–2 always read as logic zeros. Writes to these bits have no effect.

7.3.2 Data Direction Register B (DDRB)

The contents of DDRB determine whether each port B pin is an input or an output. Writing a logic one to a DDRB bit enables the output buffer for the associated port B pin; a logic zero disables the output buffer. A reset initializes all DDRB bits to logic zero, configuring all port B pins as inputs. Setting a DDRB bit to a logic one turns off the pulldown device for that pin.

DDRB — Data Direction Register B **$0005**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	0	0	0	0	DDRB1	DDRB0
RESET:	0	0	0	0	0	0	0	0

Figure 7-6. Data Direction Register B (DDRB)

DDRB1 and DDRB0 — Data Direction Bits 1 and 0

These read/write bits control port B data direction.

1 = Corresponding port B pin configured as output
0 = Corresponding port B pin configured as input

Bit 7–2 — Not used

Bits 7–2 always read as logic zeros. Writes to these bits have no effect.

NOTE

Avoid glitches on port B pins by writing to the port B data register before changing DDRB bits from logic zero to logic one.

7.3.3 Pulldown Register B (PDRB)

Programming the SWPDI bit in the mask option register to a logic zero enables the port A and port B pulldown devices. The port B pulldown devices sink approximately 100 µA and are under the control of the PDIB1 and PDIB0 bits in pulldown register B. Clearing PDIB1 and PDIB0 turns on the port B pulldown devices if they are configured as inputs. A pulldown device can be turned on only when its pin is an input. When SWPDI is a logic zero, reset initializes both port B pins as inputs with pulldown devices turned on.

Programming the SWPDI bit to a logic one disables both of the port B pulldown devices. Reset initializes both port B pins as inputs with pulldown devices disabled when the SWPDI bit is programmed to a logic one.

PDRB — Pulldown Register B **$0011**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	—	—	—	—	PDIB1	PDIB0
RESET:	0	0	0	0	0	0	0	0

Figure 7-7. Pulldown Register B (PDRB)

PDIB1 and PDIB0 — Port B Pulldown Inhibit Bits 1 and 0
Writing logic zeros to these write-only bits turns on the port B pulldown devices. Reading pulldown register B returns undefined data. Reset clears PDIB1 and PDIB0.
1 = Corresponding port B pin pulldown device turned off
0 = Corresponding port B pin pulldown device turned on

Bits 7–2 — Not used
Bits 7–2 always read as logic zeros.

Programming the SWPDI bit in the mask option register to logic one turns off all port A and port B pulldown devices and disables software control of the pulldown devices. Reset has no effect on the pulldown devices when the SWPDI bit is set to a logic one.

NOTE

Avoid a floating port B input by clearing its pulldown register bit before changing its DDRB bit from logic one to logic zero.

NOTE

Do not use read-modify-write instructions on pulldown register B.

7.3.4 Port B Logic

Figure 7-8 shows the port B I/O logic.

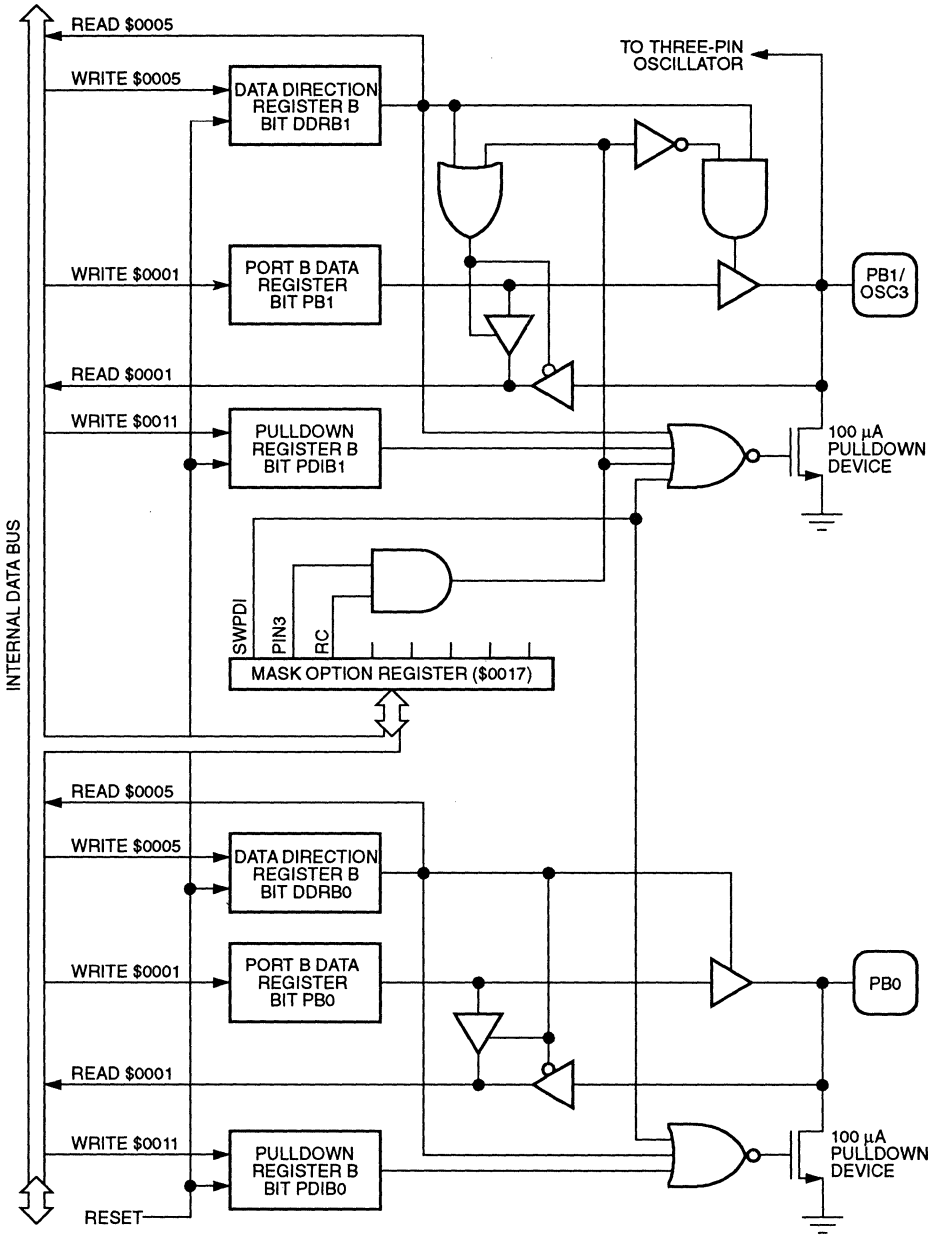


Figure 7-8. Port B I/O Circuit

When a port B pin is programmed as an output, reading the port bit reads the value of the data latch and not the voltage on the pin itself. When a port B pin is programmed as an input, reading the port bit reads the voltage level on the pin. The data latch can always be written, regardless of the state of its DDR bit. Table 7-2 summarizes the operation of the PB0 pin.

Table 7-2. PB0 Pin Functions

Control Bits			PB0 Pin Mode	Accesses to PDRB		Accesses to DDRB	Accesses to PORTB	
SWPDI	PDIB0	DDRB0		Read	Write	Read/Write	Read	Write
1	X[1]	0	Input, Hi-Z	U[2]	PDIB0	DDRB0	Pin	PB0
1	X	1	Output	U	PDIB0	DDRB0	PB0	PB0
0	0	0	Input, Pulldown On	U	PDIB0	DDRB0	Pin	PB0
0	0	1	Output	U	PDIB0	DDRB0	PB0	PB0
0	1	0	Input, Hi-Z	U	PDIB0	DDRB0	Pin	PB0
0	1	1	Output	U	PDIB0	DDRB0	PB0	PB0

1. X = Don't care
2. U = Undefined

Programming the RC and PIN3 bits to logic one disables the PB1/OSC3 output buffer and pulldown device. The PB1/OSC3 bit becomes an output from the three-pin RC oscillator. The DDRB1 and PB1 bits are available as read/write storage locations; reset clears DDRB1 but does not affect PB1. Table 7-3 summarizes the operation of the PB1/OSC3 pin.

Table 7-3. PB1/OSC3 Pin Functions

Control Bits					PB1/OSC3 Pin Mode	Accesses to PDRB		Accesses to DDRB	Accesses to PORTB	
RC	PIN3	SWPDI	PDIB1	DDRB1		Read	Write	Read/Write	Read	Write
0	X[1]	1	X	0	Input, Hi-Z	U[2]	PDIB1	DDRB1	Pin	PB1
0	X	1	X	1	Output	U	PDIB1	DDRB1	PB1	PB1
0	X	0	0	0	Input, Pulldown On	U	PDIB1	DDRB1	Pin	PB1
0	X	0	0	1	Output	U	PDIB1	DDRB1	PB1	PB1
0	X	0	1	0	Input, Hi-Z	U	PDIB1	DDRB1	Pin	PB1
0	X	0	1	1	Output	U	PDIB1	DDRB1	PB1	PB1
1	0	1	X	0	Input, Hi-Z	U	PDIB1	DDRB1	Pin	PB1
1	0	1	X	1	Output	U	PDIB1	DDRB1	PB1	PB1
1	0	0	0	0	Input, Pulldown On	U	PDIB1	DDRB1	Pin	PB1
1	0	0	0	1	Output	U	PDIB1	DDRB1	PB1	PB1
1	0	0	1	0	Input, Hi-Z	U	PDIB1	DDRB1	Pin	PB1
1	0	0	1	1	Output	U	PDIB1	DDRB1	PB1	PB1
1	1	X	X	X	3-Pin RC Oscillator Output	U	PDIB1	DDRB1	PB1	PB1

1. X = don't care
2. U = undefined

SECTION 8
MULTIFUNCTION TIMER

This section describes the operation of the multifunction timer and the COP watchdog. Figure 8-1 shows the organization of the timer subsystem.

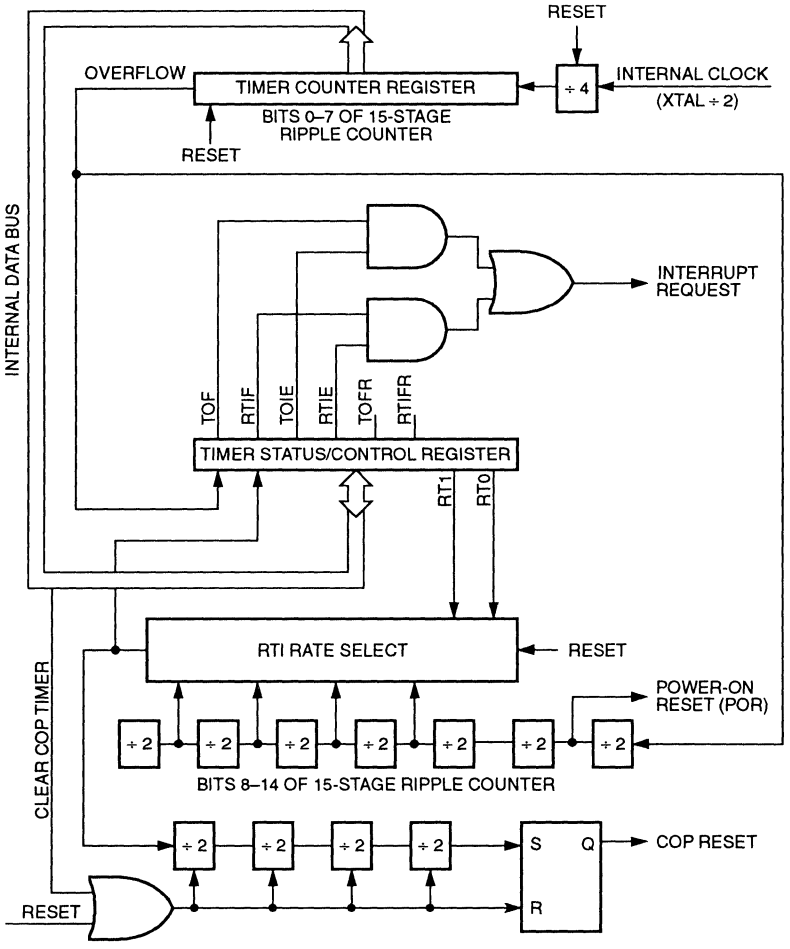


Figure 8-1. Multifunction Timer Block Diagram

8.1 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

TSCR — Timer Status and Control Register **$0008**

	Bit 7	6	5	4	3	2	1	Bit 0
	TOF	RTIF	TOIE	RTIE	TOFR	RTIFR	RT1	RT0
RESET:	0	0	0	0	—	—	1	1

Figure 8-2. Timer Status and Control Register (TSCR)

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.

TOIE — Timer Overflow Interrupt Enable
This read/write bit enables timer overflow interrupts. Reset clears TOIE.
1 = Timer overflow interrupts enabled
0 = Timer overflow interrupts disabled

RTIE — Real-Time Interrupt Enable
This read/write bit enables real-time interrupts. Reset clears RTIE.
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 a logic zero. Reset does not affect TOFR.

RTIFR — Real-Time Interrupt Flag Reset

Writing a logic one to this write-only bit clears the RTIF bit. RTIFR always reads as a logic zero. Reset does not affect RTIFR.

RT1, 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, selecting the longest COP timeout period and real-time interrupt period.

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	8 × RTI Period	65.5 ms
0 1	$f_{OP} \div 2^{15}$	16.4 ms	8 × RTI Period	131.1 ms
1 0	$f_{OP} \div 2^{16}$	32.8 ms	8 × RTI Period	262.1 ms
1 1	$f_{OP} \div 2^{17}$	65.5 ms	8 × RTI Period	524.3 ms

8.2 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.

TCNTR — Timer Counter Register **$0009**

	Bit 7	6	5	4	3	2	1	Bit 0
RESET:	0	0	0	0	0	0	0	0

Figure 8-3. Timer Counter Register (TCNTR)

Power-on clears the entire counter chain and begins clocking the counter. After 4064 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 1024 internal clock cycles.

Each count of the timer counter register takes eight oscillator cycles or four cycles of the internal clock.

8.3 COP Watchdog

Four counter stages at the end of the timer make up the mask-optional computer operating properly (COP) watchdog. 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.

COPR — COP Register **$03F0**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	—	—	—	—	—	COPC
RESET:	—	—	—	—	—	—	—	0

Figure 8-4. COP Register (COPR)

COPC — COP Clear

This write-only bit resets the COP watchdog. Reading address $03F0 returns the ROM data at that address.

The COP watchdog is active in the Run, Wait, and Halt Modes of operation if the COPEN bit in the mask option register is set.

The STOP instruction disables the COP watchdog by clearing the counter and turning off its clock source. In applications that depend on the COP watchdog, the STOP instruction can be disabled by programming the SWAIT bit to a logic one in the mask option register. In applications that have wait cycles longer than the COP timeout period, the COP watchdog can be disabled by not programming the COPEN bit to a logic one in the mask option register.

NOTE

If the voltage on the $\overline{IRQ}/V_{PP}$ pin exceeds $2 \times V_{DD}$, the COP watchdog turns off and remains off until the $\overline{IRQ}/V_{PP}$ voltage falls below $2 \times V_{DD}$.

Table 8-2 summarizes recommended conditions for enabling and disabling the COP watchdog.

Table 8-2. COP Watchdog Recommendations

Voltage on $\overline{IRQ}/V_{PP}$ Pin	SWAIT Bit[1]	Wait/Halt Time	Recommended COP Watchdog Condition
Less than $2 \times V_{DD}$	1	Less than COP Timeout Period	Enabled[2]
Less than $2 \times V_{DD}$	1	Greater than COP Timeout Period	Disabled
Less than $2 \times V_{DD}$	0	X[3]	Disabled
More than $2 \times V_{DD}$	X	X	Automatically Disabled

1. The SWAIT bit in the mask option register converts STOP instructions to HALT instructions.
2. Reset the COP watchdog immediately before executing the WAIT/HALT instruction.
3. X = don't care

SECTION 9
EPROM/OTPROM

This section describes how to program the 504-byte EPROM/OTPROM.

NOTE

In packages with no quartz window, the 504 bytes of EPROM function as one-time programmable ROM (OTPROM).

9.1 EPROM Programming Register (EPROG)

The EPROM programming register contains the control bits for programming the EPROM/OTPROM. In normal operation, the EPROM programming register is a read-only register that contains all logic zeros.

EPROG — EPROM Programming Register **$0018**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	0	0	0	ELAT	MPGM	EPGM
RESET:	—	—	—	—	—	0	0	0

Figure 9-1. EPROM Programming Register (EPROG)

ELAT — EPROM Bus Latch

This read/write bit configures address and data buses for programming the EPROM/OTPROM array. EPROM/OTPROM data cannot be read when ELAT is set. Clearing the ELAT bit also clears the EPGM bit. Reset clears ELAT.

1 = Address and data buses configured for EPROM/OTPROM programming
0 = Address and data buses configured for normal operation

MPGM — Mask Option Register (MOR) Programming

This read/write bit applies programming power from the $\overline{IRQ}/V_{PP}$ pin to the MOR. Reset clears MPGM.

1 = MOR programming power switched on
0 = MOR programming power switched off

EPGM — EPROM Programming

This read/write bit applies the voltage from the $\overline{IRQ}/V_{PP}$ pin to the EPROM/OTPROM. To write the EPGM bit, the ELAT bit must already be set. Reset clears EPGM.

1 = EPROM/OTPROM programming power switched on
0 = EPROM/OTPROM programming power switched off

NOTE

Writing logic ones to both the ELAT and EPGM bits with a single instruction sets ELAT and clears EPGM. ELAT must be set first by a separate instruction.

Bits 7–3 — Reserved

Bits 7–3 are factory test bits that always read as logic zeros.

9.2 EPROM/OTPROM Programming

Factory-provided software for programming the EPROM/OTPROM is available through the Motorola Freeware Bulletin Board Service (BBS). The number is (512) 891-FREE. After making the connection, type bbs in lowercase letters and press the return key to start the BBS software.

The programming software copies to the 496-byte space located at EPROM/OTPROM addresses $0200–$03EF, to the 8-byte space at addresses $03F8–$03FF, and to the mask option register at address $0017.

Figure 9-2 shows the circuit used to download to the on-chip EPROM/OTPROM using the factory-provided programming software.

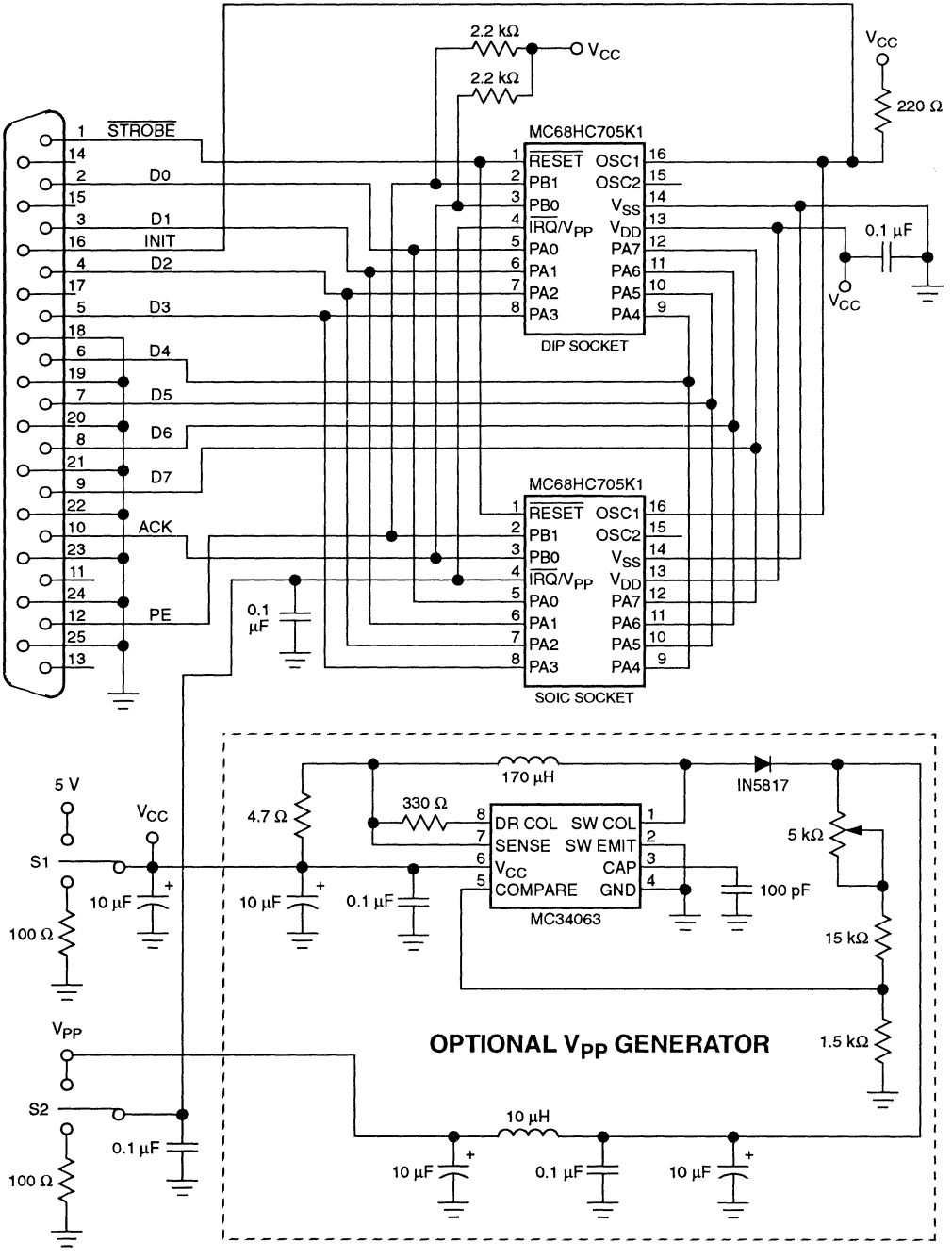


Figure 9-2. Programming Circuit

The following sequence shows the steps in programming a byte of EPROM/OTPROM:

1. Switch S1 powers up the MC68HC705K1.
2. Software synchronizes the external oscillator to the internal clock.
3. Switch S2 applies V_{PP} to the $\overline{IRQ}/V_{PP}$ pin.
4. Software sets the ELAT bit.
5. Software writes to an EPROM/OTPROM address.
6. Software sets the EPGM bit for a time t_{EPGM} to apply the programming voltage.
7. Software clears the ELAT bit.

NOTE

To program the EPROM/OTPROM, V_{DD} must be greater than 4.5 Vdc.

9.3 EPROM Erasing

MCUs with windowed packages permit EPROM erasure with ultraviolet light. Erase the EPROM by exposing it to 15 Ws/cm^2 of ultraviolet light with a wavelength of 2537 angstroms. Position the ultraviolet light source 1 inch from the window. Do not use a shortwave filter. The erased state of an EPROM bit is a logic zero.

9.4 Mask Option Register (MOR)

The mask option register is an EPROM/OTPROM byte that controls the following options:

- Port A and port B programmable pulldown devices (enable or disable)
- Oscillator connections (two-pin or three-pin RC oscillator)
- Oscillator connections (RC oscillator or crystal/ceramic resonator)
- STOP instruction (enable or disable)
- Low voltage reset (enable or disable)
- Port A external interrupt function (enable or disable)
- IRQ trigger sensitivity (edge-triggered only or both edge- and level-triggered)
- COP watchdog (enable or disable)

The mask option register is unaffected by reset. The erased state of the mask option register is $0000.

MOR — Mask Option Register **$0017**

	Bit 7	6	5	4	3	2	1	Bit 0
	SWPDI	PIN3	RC	SWAIT	LVRE	PIRQ	LEVEL	COPEN
RESET:	UNAFFECTED BY RESET							
ERASED:	0	0	0	0	0	0	0	0

Figure 9-3. Mask Option Register (MOR)

SWPDI —Software Pulldown Inhibit

This EPROM bit inhibits software control of the port A and port B pulldown devices.

1 = Software pulldown inhibited
0 = Software pulldown enabled

PIN3 — Three-Pin RC Oscillator

This EPROM bit configures the on-chip oscillator as either a three-pin oscillator or as a two-pin oscillator. The PIN3 bit should be cleared when the RC bit is clear.

1 = Three-pin oscillator configured
0 = Two-pin oscillator configured

RC — RC Oscillator

This EPROM bit configures the on-chip oscillator for an external RC network.

1 = Oscillator configured for external RC network

0 = Oscillator configured for external crystal, ceramic resonator, or clock source

SWAIT — STOP Conversion to WAIT

This EPROM bit disables the STOP instruction and prevents inadvertently turning off the COP watchdog with a STOP instruction. When the SWAIT bit is set, a STOP instruction puts the MCU in Halt Mode. Halt Mode is a low-power state similar to Wait Mode. The internal oscillator and timer clock continue to run, but the CPU clock stops. When the SWAIT bit is clear, a STOP instruction stops the internal oscillator, the internal clock, the CPU clock, and the timer clock.

1 = STOP instruction converted to WAIT instruction

0 = STOP instruction not converted to WAIT instruction

LVRE — Low Voltage Reset Enable

This EPROM bit enables the low voltage reset (LVR) circuit.

1 = LVR circuit enabled

0 = LVR circuit disabled

PIRQ — Port A IRQ Enable

This EPROM bit enables the PA3–PA0 pins to function as external interrupt sources.

1 = PA3–PA0 enabled as external interrupt sources

0 = PA3–PA0 not enabled as external interrupt sources

LEVEL — External Interrupt Sensitivity

This EPROM bit makes the external interrupt inputs level-triggered as well as edge-triggered.

1 = $\overline{IRQ}/V_{PP}$ pin negative-edge triggered and low-level triggered; PA3–PA0 pins positive-edge triggered and high-level triggered

0 = $\overline{IRQ}/V_{PP}$ pin negative-edge triggered only; PA3–PA0 pins positive-edge triggered only

COPEN — COP Watchdog Enable

This EPROM bit enables the COP watchdog.

1 = COP watchdog enabled

0 = COP watchdog disabled

SECTION 10
PERSONALITY EPROM

This section describes how to program the 64-bit personality EPROM (PEPROM). Figure 10-1 shows the structure of the PEPROM subsystem.

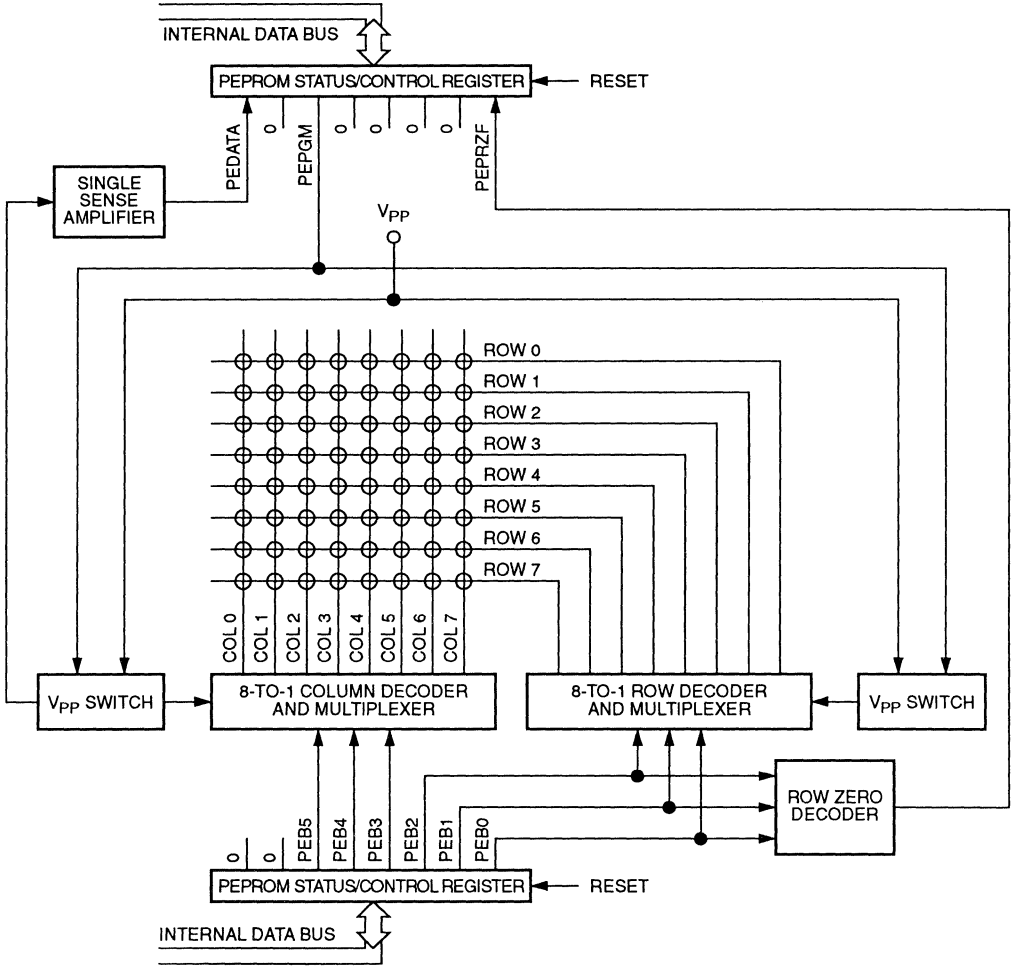


Figure 10-1. Personality EPROM

10.1 PEPROM Registers

Two I/O registers control programming and reading of the PEPROM:

- The PEPROM bit select register (PEBSR)
- The PEPROM status and control register (PESCR)

10.1.1 PEPROM Bit Select Register (PEBSR)

The PEPROM bit select register selects one of 64 bits in the PEPROM array. Reset clears all the bits in the PEPROM bit select register.

PEBSR — PEPROM Bit Select Register **$000E**

	Bit 7	6	5	4	3	2	1	Bit 0
	PEB7	PEB6	PEB5	PEB4	PEB3	PEB2	PEB1	PEB0
RESET:	0	0	0	0	0	0	0	0

Figure 10-2. PEPROM Bit Select Register (PEBSR)

PEB7 and PEB6 — Not connected to the PEPROM array
These read/write bits are available as storage locations. Reset clears PEB7 and PEB6.

PEB5–PEB0 — PEPROM Bit Select Bits
These read/write bits select one of 64 bits in the PEPROM as shown in Table 10-1. Bits PEB2–0 select the PEPROM row, and bits PEB5–3 select the PEPROM column. Reset clears PEB5–PEB0, selecting the PEPROM bit in row zero, column zero.

Table 10-1. PEPROM Bit Selection

PEBSR	PEPROM Bit Selected	
$00	Row 0	Column 0
$01	Row 1	Column 0
$02	Row 2	Column 0
↓	↓	↓
$07	Row 7	Column 0
$08	Row 0	Column 1
$09	Row 1	Column 1
$0A	Row 2	Column 1
↓	↓	↓
$0F	Row 7	Column 1
$10	Row 0	Column 2
$11	Row 1	Column 2
$12	Row 2	Column 2
↓	↓	↓
$37	Row 7	Column 6
$38	Row 0	Column 7
$39	Row 1	Column 7
$3A	Row 2	Column 7
$3B	Row 3	Column 7
$3C	Row 4	Column 7
$3D	Row 5	Column 7
$3E	Row 6	Column 7
$3F	Row 7	Column 7

10.1.2 PEPROM Status and Control Register (PESCR)

The PEPROM status and control register controls the PEPROM programming voltage. This register also transfers the PEPROM bits to the internal data bus and contains a row zero flag.

PESCR — PEPROM Status and Control Register **$000F**

	Bit 7	6	5	4	3	2	1	Bit 0
	PEDATA	0	PEPGM	0	0	0	0	PEPRZF
RESET:	—	0	0	0	0	0	0	1

Figure 10-3. PEPROM Status and Control Register (PESCR)

PEDATA — PEPROM Data

This read-only bit is the state of the PEPROM sense amplifier and shows the state of the currently selected bit. Reset does not affect the PEDATA bit.

1 = PEPROM data logic one
0 = PEPROM data logic zero

PEPGM — PEPROM Program Control

This read/write bit controls the switches that apply the programming voltage, V_{PP}, to the selected PEPROM cell. Reset clears PEPGM.

1 = Programming voltage applied
0 = Programming voltage not applied

PEPRZF — PEPROM Row Zero Flag

This read-only bit is set when the PEPROM bit select register selects the first row (row zero) of the PEPROM array. Selecting any other row clears PEPRZF. Monitoring PEPRZF can reduce the code needed to access one byte of PEPROM. Reset sets PEPRZF.

1 = Row zero selected
0 = Row zero not selected

10.2 PEPROM Programming

Factory-provided software for programming the PEPROM is available through the Motorola Freeware Bulletin Board Service (BBS). The number is (512) 891-FREE. After making the connection, type bbs in lowercase letters. Then press the return key to start the BBS software.

The circuit shown in **Figure 9-2. Programming Circuit** can be used to program the PEPROM with the factory-provided programming software.

NOTE

To program the PEPROM, V_{DD} must be greater than 4.5 Vdc.

The PEPROM can also be programmed by user software with V_{PP} applied to the $\overline{IRQ}/V_{PP}$ pin. The following sequence shows how to program each PEPROM bit:

1. Select a PEPROM bit by writing to PEBSR.
2. Set the PEPGM bit in PESCR.
3. Wait 3 ms.
4. Clear the PEPGM bit.

NOTE

While the PEPGM bit is set and V_{PP} is applied to the $\overline{IRQ}/V_{PP}$ pin, do not access bits that are to be left unprogrammed (erased).

10.3 PEPROM Reading

The following sequence shows how to read the PEPROM:

1. Select a bit by writing to PEBSR.
2. Read the PEDATA bit in PESCR.
3. Store the PEDATA bit in RAM or in a register.
4. Select another bit by changing PEBSR.
5. Continue reading and storing the PEDATA bits until the required personality EPROM data is stored.

Reading the PEPROM is easiest when each PEPROM column contains one byte. Selecting a row 0 bit selects the first bit, and incrementing the PEPROM bit select register (PEBSR) selects the next row 1 bit from the same column. Incrementing PEBSR seven more times selects the remaining bits of the column and selects the row 0 bit of the next column, setting the row 0 flag, PEPRZF.

A PEPROM byte that has been read can be transferred to the personality EPROM bit select register (PEBSR) so that subsequent reads of the PEBSR quickly yield that PEPROM byte.

10.4 PEPROM Erasing

MCUs with windowed packages permit PEPROM erasure with ultraviolet light. Erase the PEPROM by exposing it to 15 Ws/cm^2 of ultraviolet light with a wavelength of 2537 angstroms. Position the ultraviolet light source 1 inch from the window. Do not use a shortwave filter. The erased state of a PEPROM bit is a logic zero.

SECTION 11
INSTRUCTION SET

This section describes the M68HC05 addressing modes and instruction types.

11.1 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

11.1.1 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.

11.1.2 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.

11.1.3 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.

11.1.4 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.

11.1.5 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 uses RAM or I/O location.

11.1.6 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.

11.1.7 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.

11.1.8 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.

11.2 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

11.2.1 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 11-1 lists the register/memory instructions.

Table 11-1. 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

11.2.2 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 11-2 lists the read-modify-write instructions.

Table 11-2. 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

11.2.3 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 11-3 lists the jump and branch instructions.

Table 11-3. 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

11.2.4 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 11-4 lists these instructions.

Table 11-4. Bit Manipulation Instructions

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

11.2.5 Control Instructions

These register reference instructions control CPU operation during program execution. Control instructions, listed in Table 11-5, use inherent addressing.

Table 11-5. 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

11.3 Instruction Set Summary

Table 11-6 is an alphabetical list of all M68HC05 instructions and shows the effect of each instruction on the condition code register.

Table 11-6. Instruction Set Summary

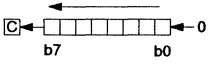
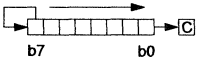
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr* 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* AND *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

Table 11-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BHCC *rel*	Branch if Half-Carry Bit Clear	PC ← (PC) + 2 + *rel* ? H = 0	—	—	—	—	—	REL	28	rr	3
BHCS *rel*	Branch if Half-Carry Bit Set	PC ← (PC) + 2 + *rel* ? H = 1	—	—	—	—	—	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? C ∨ Z = 0	—	—	—	—	—	REL	22	rr	3
BHS *rel*	Branch if Higher or Same	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BIH *rel*	Branch if $\overline{IRQ}$ Pin High	PC ← (PC) + 2 + *rel* ? $\overline{IRQ}$ = 1	—	—	—	—	—	REL	2F	rr	3
BIL *rel*	Branch if $\overline{IRQ}$ Pin Low	PC ← (PC) + 2 + *rel* ? $\overline{IRQ}$ = 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	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	5
BRN *rel*	Branch Never	PC ← (PC) + 2 + *rel* ? 1 = 0	—	—	—	—	—	REL	21	rr	3

Table 11-6. 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 ← $00	—	—	0	1	—	DIR INH INH IX1 IX	3F 4F 5F 6F 7F	dd ff 	5 3 3 6 5
CMP #*opr* CMP *opr* CMP *opr* CMP *opr*,X CMP *opr*,X CMP ,X	Compare Accumulator with Memory Byte	(A) – (M)	—	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A1 B1 C1 D1 E1 F1	ii dd hh ll ee ff ff 	2 3 4 5 4 3
COM *opr* COMA COMX COM *opr*,X COM ,X	Complement Byte (One's Complement)	M ← ($\overline{M}$) = $FF – (M) A ← ($\overline{A}$) = $FF – (M) X ← ($\overline{X}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M)	—	—	↕	↕	1	DIR INH INH IX1 IX	33 43 53 63 73	dd ff 	5 3 3 6 5
CPX #*opr* CPX *opr* CPX *opr* CPX *opr*,X CPX *opr*,X CPX ,X	Compare Index Register with Memory Byte	(X) – (M)	—	—	↕	↕	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 11-6. Instruction Set Summary (Continued)

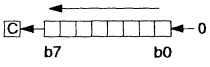
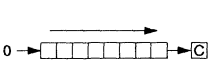
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
INC *opr* INCA INCX INC *opr*,X INC ,X	Increment Byte	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1	—	—	↕	↕	—	DIR INH INH IX1 IX	3C 4C 5C 6C 7C	dd ff 	5 3 3 6 5
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Unconditional Jump	PC ← Jump Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BC CC DC 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 CD DD ED FD	dd hh ll ee ff ff 	5 6 7 6 5
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X	Load Accumulator with Memory Byte	A ← (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	A6 B6 C6 D6 E6 F6	ii dd hh ll ee ff ff 	2 3 4 5 4 3
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X	Load Index Register with Memory Byte	X ← (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AE BE CE DE EE FE	ii dd hh ll ee ff ff 	2 3 4 5 4 3
LSL *opr* LSLA LSLX LSL *opr*,X LSL ,X	Logical Shift Left (Same as ASL)		—	—	↕	↕	↕	DIR INH INH IX1 IX	38 48 58 68 78	dd ff 	5 3 3 6 5
LSR *opr* LSRA LSRX LSR *opr*,X LSR ,X	Logical Shift Right		—	—	0	↕	↕	DIR INH INH IX1 IX	34 44 54 64 74	dd ff 	5 3 3 6 5
MUL	Unsigned Multiply	X : A ← (X) × (A)	0	—	—	—	0	INH	42		11
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X	Negate Byte (Two's Complement)	M ← –(M) = $00 – (M) A ← –(A) = $00 – (A) X ← –(X) = $00 – (X) M ← –(M) = $00 – (M) M ← –(M) = $00 – (M)	—	—	↕	↕	↕	DIR INH INH IX1 IX	30 40 50 60 70	ii ff 	5 3 3 6 5
NOP	No Operation		—	—	—	—	—	INH	9D		2

Table 11-6. Instruction Set Summary (Continued)

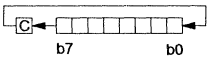
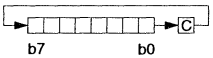
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X	Logical OR Accumulator with Memory	A ← (A) ∨ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff	2 3 4 5 4 3
ROL *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 11-6. 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	*opr*	Operand (one or two bytes)
C	Carry/borrow flag	PC	Program counter
CCR	Condition code register	PCH	Program counter high byte
dd	Direct address of operand	PCL	Program counter low byte
dd rr	Direct address of operand and relative offset of branch instruction	REL	Relative addressing mode
DIR	Direct addressing mode	*rel*	Relative program counter offset byte
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing	rr	Relative program counter offset byte
EXT	Extended addressing mode	SP	Stack pointer
ff	Offset byte in indexed, 8-bit offset addressing	X	Index register
H	Half-carry flag	Z	Zero flag
hh ll	High and low bytes of operand address in extended addressing	#	Immediate value
I	Interrupt mask	∧	Logical AND
ii	Immediate operand byte	∨	Logical OR
IMM	Immediate addressing mode	⊕	Logical EXCLUSIVE OR
INH	Inherent addressing mode	( )	Contents of
IX	Indexed, no offset addressing mode	–( )	Negation (two's complement)
IX1	Indexed, 8-bit offset addressing mode	←	Loaded with
IX2	Indexed, 16-bit offset addressing mode	?	If
M	Memory location	:	Concatenated with
N	Negative flag	↕	Set or cleared
n	Any bit	—	Not affected

Table 11-7. Opcode Map

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

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

LSB of Opcode in Hexadecimal

MSB / LSB	0
0	BRSET0 5 3 DIR

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

SECTION 12
ELECTRICAL SPECIFICATIONS

This section contains electrical and timing specifications.

12.1 Maximum Ratings

CAUTION

To avoid damage to the MCU, do not exceed the values listed in Table 12-1. Keep V_{IN} and V_{OUT} within the range $V_{SS} \leq (V_{IN}$ or $V_{OUT}) \leq V_{DD}$.

Table 12-1. Maximum Ratings[1]

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
EPROM Programming Voltage	V_{PP}	18.0	V
Current Drain Per Pin (Excluding V_{DD} and V_{SS})	I	25	mA

1. Maximum values are not guaranteed operating values.

12.2 Thermal Characteristics

Table 12-2. Thermal Characteristics

Characteristic	Symbol	Value	Unit
Maximum Junction Temperature	T_J	150	°C
Thermal Resistance MC68HC705K1P[1] MC68HC705K1DW[2]	θ_{TA}	 100 140	 °C/W °C/W
Operating Temperature Range MC68HC705K1P, DW, S[3] MC68HC705K1C[4]P, CDW, CS	T_A	 0 to +70 –40 to +85	°C
Storage Temperature Range	T_{STG}	–65 to +150	°C

1. P = Plastic dual in-line package (PDIP)
2. DW = Small outline integrated circuit (SOIC)
3. S = Ceramic dual in-line package (Cerdip)
4. C = Extended temperature range (–40 to +85 °C)

12.3 Power Considerations

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

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

where:

T_A = Ambient temperature, °C

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

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

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

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

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

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

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

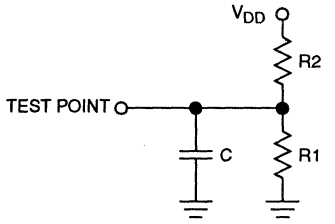
Solving equations (1) and (2) for K gives:

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

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

12.4 Equivalent Pin Loading

Figure 12-1 shows the equivalent I/O pin loading for test purposes.



PINS	V_{DD}	R1	R2	C
PA3–PA0, PB1–PB0	4.5 V	3.26 kΩ	2.38 kΩ	50 pF
PA7–PA4		470 Ω	2.38 kΩ	50 pF
PA3–PA0, PB1–PB0	3.0 V	10.91 kΩ	6.32 kΩ	50 pF

Figure 12-1. Equivalent Test Load

12.5 DC Electrical Characteristics

Table 12-3. DC Electrical Characteristics (V_{DD} = 5.0 V)[1]

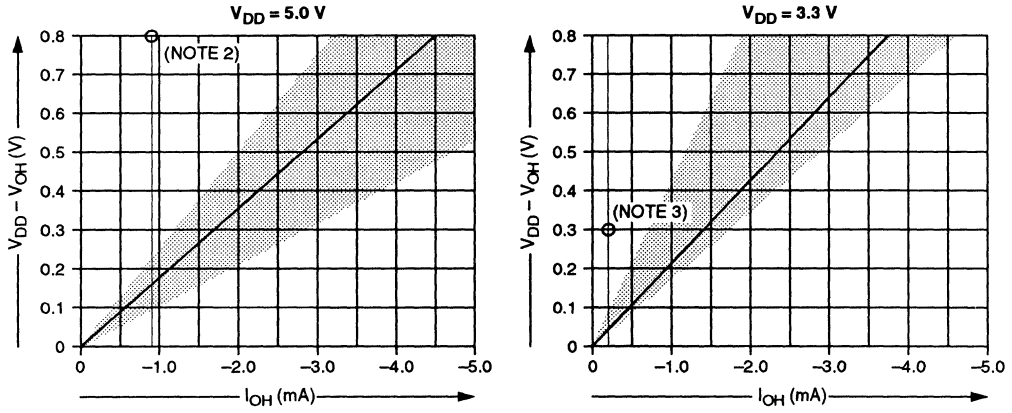
Characteristic[2]	Symbol	Min	Typ	Max	Unit
Output Voltage I_{LOAD} = 10.0 µA I_{LOAD} = –10.0 µA	 V_{OL} V_{OH}	 — V_{DD} – 0.1	 — —	 0.1 —	 V
Output High Voltage (I_{LOAD} = –0.8 mA) PA7–PA0, PB1/OSC3, PB0	V_{OH}	V_{DD} – 0.8	—	—	V
Output Low Voltage PA3–PA0, PB1/OSC3, PB0 (I_{LOAD} = 1.6 mA) PA7–PA4 (I_{LOAD} = 8.0 mA)	V_{OL}	 —	 —	 0.4 0.4	 V V
Input High Voltage PA7–PA0, PB1/OSC3, PB0, $\overline{IRQ}/V_{PP}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input Low Voltage PA7–PA0, PB1/OSC3, PB0, $\overline{IRQ}/V_{PP}$, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 × V_{DD}	V
Supply Current Run[3] Wait[4] Stop[5] 25 °C 0 °C to +70 °C (Standard) –40 °C to +85 °C (Extended)	 I_{DD}	 — — — — —	 2.6 0.9 200 700 1000	 — — — — —	 mA mA nA nA nA
I/O Ports Hi-Z Leakage Current PA7–PA0, PB1/OSC3, PB0 (Pulldown Devices Off)	I_{OZ}	—	—	±10	µA
Input Pulldown Current PA7–PA0, PB1/OSC3, PB0 (Pulldown Devices On)	I_{IL}	50	75	200	µA
Input Current $\overline{IRQ}/V_{PP}$, OSC1 $\overline{RESET}$ (Pulldown Device Off) $\overline{RESET}$ (Pulldown Device On)	I_{IN}	 — 1.0	 — 4.0	 ±1 ±1 8.0	 µA µA mA
Capacitance Ports (Input or Output) $\overline{RESET}$, $\overline{IRQ}/V_{PP}$	 C_{OUT} C_{IN}	 — —	 — —	 12 8	 pF pF
Low Voltage Reset Threshold[6]	V_{LVR}	2.8	3.5	4.5	V
Oscillator Internal Resistor (OSC1 to OSC2)	R_{OSC}	1.0	2.0	3.0	MΩ
Programming Voltage[7]	V_{PP}	17.0	17.5	18.0	V
Programming Current	I_{PP}	—	5	10	mA

1. V_{DD} = 5.0 Vdc ±10%; V_{SS} = 0 Vdc.
2. Values reflect average measurements at midpoint of voltage range at 25 °C.
3. Run (operating) I_{DD} measured using external square wave clock source (f_{OSC} = 4.2 MHz) with all inputs 0.2 V from rail; no dc loads; less than 50 pF on all outputs; C_L = 20 pF on OSC2.
4. Wait I_{DD} measured using external square wave clock source (f_{OSC} = 4.2 MHz) with all inputs 0.2 V from rail; no dc loads; less than 50 pF on all outputs; C_L = 20 pF on OSC2. All ports configured as inputs; V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V. OSC2 capacitance linearly affects wait I_{DD}.
5. Stop I_{DD} measured with OSC1 = V_{DD}. All ports configured as inputs; V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V. With low voltage reset enabled, stop I_{DD} can be as high as 25 µA.
6. All MCUs guaranteed to operate at V_{DD} = 5 V ±10%. Each MCU guaranteed to operate at its V_{LVR}.
7. Programming voltage measured at $\overline{IRQ}/V_{PP}$ pin.

Table 12-4. DC Electrical Characteristics (V_{DD} = 3.3 V)[1]

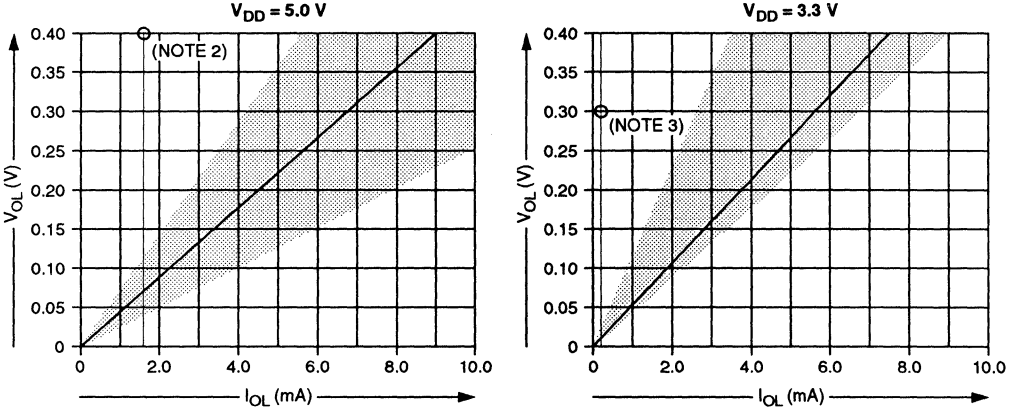
Characteristic[2]	Symbol	Min	Typ	Max	Unit
Output Voltage I_{LOAD} = 10.0 µA I_{LOAD} = –10.0 µA	 V_{OL} V_{OH}	 — V_{DD} – 0.1	 — —	 0.1 —	 V V
Output High Voltage (I_{LOAD} = –0.4 mA) PA7–PA0, PB1/OSC3, PB0	V_{OH}	V_{DD} – 0.3	—	—	V
Output Low Voltage PA3–PA0, PB1/OSC3, PB0 (I_{LOAD} = 0.4 mA) PA7–PA4 (I_{LOAD} = 3.0 mA)	V_{OL}	 —	 —	0.3 0.3	V V
Input High Voltage PA7–PA0, PB1/OSC3, PB0, $\overline{IRQ}$/V_{PP}, $\overline{RESET}$, OSC1	V_{IH}	0.7 × V_{DD}	—	V_{DD}	V
Input Low Voltage PA7–PA0, PB1/OSC3, PB0,$\overline{IRQ}$/V_{PP}, $\overline{RESET}$, OSC1	V_{IL}	V_{SS}	—	0.2 × V_{DD}	V
Supply Current Run[3] Wait[4] Stop[5] 25 °C 0 °C to 70 °C (Standard) –40 °C to +85 °C (Extended)	 I_{DD}	 — — — — —	 0.7 300 50 500 1000	 — — — — —	 mA µA nA nA nA
I/O Ports Hi-Z Leakage Current PA7–PA0, PB1/OSC3, PB0 (Pulldown Devices Off)	I_{OZ}	—	—	±10	µA
Input Pulldown Current PA7–PA0, PB1/OSC3, PB0 (Pulldown Devices On)	I_{IL}	10	20	100	µA
Input Current $\overline{IRQ}$/V_{PP}, OSC1 $\overline{RESET}$ (Pulldown Devices Off) $\overline{RESET}$ (Pulldown Devices On)	I_{IN}	 — 0.2	 — 2.0	±1 ±1 4.0	µA µA mA
Capacitance Ports (Input or Output) $\overline{RESET}$, $\overline{IRQ}$/V_{PP}	 C_{OUT} C_{IN}	 — —	 — —	 12 8	 pF pF
Oscillator Internal Resistor (OSC1 to OSC2)	R_{OSC}	1.0	2.0	3.0	MΩ

1. V_{DD} = 3.3 Vdc ±10%; V_{SS} = 0 Vdc.
2. Values reflect average measurements at midpoint of voltage range at 25 °C.
3. Run (operating) I_{DD} measured using external square wave clock source (f_{OSC} = 2.0 MHz) with all inputs 0.2 V from rail; no dc loads; less than 50 pF on all outputs; C_L = 20 pF on OSC2.
4. Wait I_{DD} measured using external square wave clock source (f_{OSC} = 2.0 MHz) with all inputs 0.2 V from rail; no dc loads; less than 50 pF on all outputs; C_L = 20 pF on OSC2. All ports configured as inputs; V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V. OSC2 capacitance linearly affects wait I_{DD}.
5. Stop I_{DD} measured with OSC1 = V_{DD} and low voltage reset disabled. All ports configured as inputs; V_{IL} = 0.2 V, V_{IH} = V_{DD} – 0.2 V.



NOTES:
1. Shaded area indicates variation in driver characteristics due to changes in temperature and for normal processing tolerances. Within the limited range of values shown, V vs I curves are approximately straight lines.
2. At V_{DD} = 5.0 V, devices are specified and tested for $V_{OL} \leq 800$ mV @ $I_{OL} = -0.8$ mA.
3. At V_{DD} = 3.3 V, devices are specified and tested for $V_{OL} \leq 300$ mV @ $I_{OL} = -0.2$ mA.

Figure 12-2. Typical High-Side Driver Characteristics



NOTES:
1. Shaded area indicates variation in driver characteristics due to changes in temperature and for normal processing tolerances. Within the limited range of values shown, V vs I curves are approximately straight lines.
2. At V_{DD} = 5.0 V, devices are specified and tested for $V_{OL} \leq 400$ mV @ $I_{OL} = 1.6$ mA.
3. At V_{DD} = 3.3 V, devices are specified and tested for $V_{OL} \leq 300$ mV @ $I_{OL} = 0.4$ mA.

Figure 12-3. Typical Low-Side Driver Characteristics

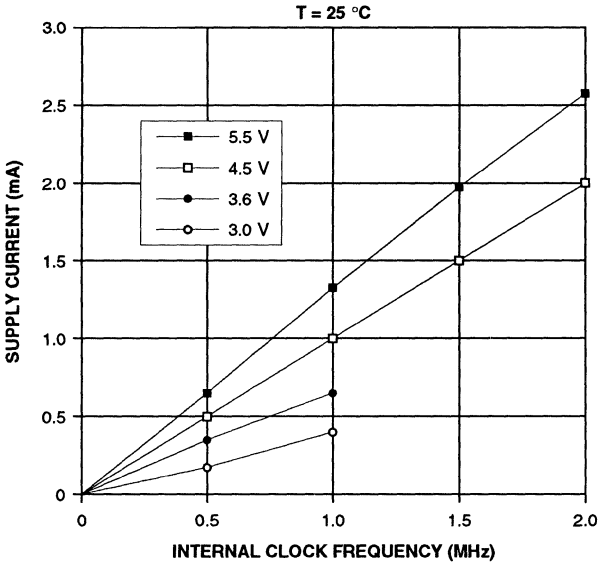


Figure 12-4. Run I_{DD} vs Internal Clock Frequency

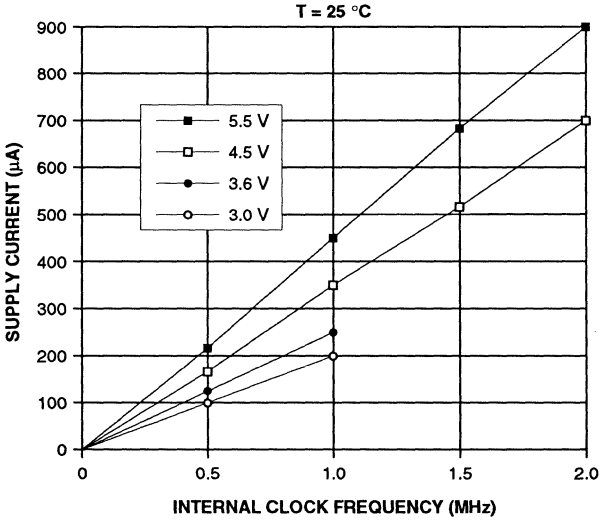


Figure 12-5. Wait I_{DD} vs Internal Clock Frequency

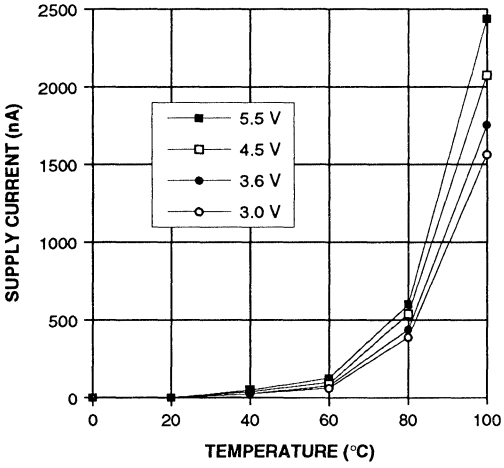


Figure 12-6. Stop I_{DD} vs Temperature

12.6 Control Timing

Table 12-5. Control Timing (V_{DD} = 5.0 V)[1]

Characteristic	Symbol	Min	Max	Unit
Oscillator Frequency 3-Pin RC Oscillator 2-Pin RC Oscillator Crystal/Ceramic Resonator[2] External Clock	f_{OSC}	 [3] [3] 0.500 dc	 1.2 2.4 4.0 4.0	MHz
Internal Operating Frequency ($f_{OSC} \div 2$) 3-Pin RC Oscillator 2-Pin RC Oscillator Crystal/Ceramic Resonator External Clock	f_{OP}	 [3] [3] 0.250 dc	 0.6 1.2 2.0 2.0	MHz
2-Pin RC Oscillator Frequency Combined Stability[4] f_{OSC} = 2.0 MHz; V_{DD} = 5.0 Vdc ±10%; T_A = –40 °C to +85 °C f_{OSC} = 2.0 MHz; V_{DD} = 5.0 Vdc ±10%; T_A = 0 °C to +40 °C	Δf_{OSC}	 — —	 ±25 ±20	%
3-Pin RC Oscillator Frequency Combined Stability[4] f_{OSC} = 1.0 MHz; V_{DD} = 5.0 Vdc ±10%; T_A = –40 °C to +85 °C f_{OSC} = 1.0 MHz; V_{DD} = 5.0 Vdc ±10%; T_A = 0 °C to +40 °C	Δf_{OSC}	 — —	 ±15 ±10	%
Cycle Time ($1 \div f_{OP}$)	t_{CYC}	500	—	ns
RC Oscillator Stabilization Time	t_{RCON}	—	1	ms
Crystal Oscillator Start-Up Time	t_{OXON}	—	100	ms
Stop Recovery Start-Up Time	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width Low	t_{RL}	1.5	—	t_{CYC}
Timer Resolution[5]	t_{RESL}	4.0	—	t_{CYC}
$\overline{\text{IRQ}}$ Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	250	—	ns
$\overline{\text{IRQ}}$ Interrupt Pulse Period	t_{ILIL}	[6]	—	t_{CYC}
PA3–PA0 Interrupt Pulse Width High (Edge-Triggered)	t_{IHIL}	250	—	ns
PA3–PA0 Interrupt Pulse Period	t_{IHIH}	[6]	—	t_{CYC}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns
Programming Time per Byte[7]	t_{EPGM}	10	15	ms

1. V_{DD} = 5.0 Vdc ±10%; V_{SS} = 0 Vdc; $T_A = T_L$ to T_H.
2. Use only AT-cut crystals.
3. Minimum oscillator frequency with RC oscillator option is limited only by size of external R and C and leakage of external C.
4. Including processing tolerances and variations in temperature and supply voltage. Excluding tolerances of external R and C.
5. The 2-bit timer prescaler is the limiting factor in determining timer resolution.
6. The minimum period t_{ILIL} or t_{IHIH} should not be less than the number of cycles it takes to execute the interrupt service routine plus 19 t_{CYC}.
7. T_{EPGM} is programming time per byte and may be accumulated during multiple programming passes.

Table 12-6. Control Timing (V_{DD} = 3.3 V)[1]

Characteristic	Symbol	Min	Max	Unit
Oscillator Frequency 3-Pin RC Oscillator 2-Pin RC Oscillator Crystal/Ceramic Resonator[2] External Clock	f_{OSC}	 [3] [3] 0.500 dc	 1.2 2.0 2.0 2.0	MHz
Internal Operating Frequency ($f_{OSC} \div 2$) 3-Pin RC Oscillator 2-Pin RC Oscillator Crystal/Ceramic Resonator External Clock	f_{OP}	 [3] [3] 0.250 dc	 0.6 1.0 1.0 1.0	MHz
2-Pin RC Oscillator Frequency Combined Stability[4] f_{OSC} = 2.0 MHz; V_{DD} = 3.3 Vdc ±10%; T_A = −40 °C to +85 °C f_{OSC} = 2.0 MHz; V_{DD} = 3.3 Vdc ±10%; T_A = 0 °C to +40 °C	Δf_{OSC}	 — —	 ±40 ±30	%
3-Pin RC Oscillator Frequency Combined Stability[4] f_{OSC} = 1.0 MHz; V_{DD} = 3.3 Vdc ±10%; T_A = −40 °C to +85 °C f_{OSC} = 1.0 MHz; V_{DD} = 3.3 Vdc ±10%; T_A = 0 °C to +40 °C	Δf_{OSC}	 — —	 ±20 ±15	%
Cycle Time ($1 \div f_{OP}$)	t_{CYC}	1000	—	ns
RC Oscillator Stabilization Time	t_{RCON}	—	1	ms
Crystal Oscillator Start-Up Time	t_{OXON}	—	100	ms
Stop Recovery Start-Up Time	t_{ILCH}	—	100	ms
$\overline{\text{RESET}}$ Pulse Width Low	t_{RL}	1.5	—	t_{CYC}
Timer Resolution[5]	t_{RESL}	4.0	—	t_{CYC}
$\overline{\text{IRQ}}$ Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	250	—	ns
$\overline{\text{IRQ}}$ Interrupt Pulse Period	t_{ILIL}	[6]	—	t_{CYC}
PA3–PA0 Interrupt Pulse Width High (Edge-Triggered)	t_{IHIL}	250	—	ns
PA3–PA0 Interrupt Pulse Period	t_{IHIH}	[6]	—	t_{CYC}
OSC1 Pulse Width	t_{OH}, t_{OL}	200	—	ns

1. V_{DD} = 3.3 Vdc ±10%; V_{SS} = 0 Vdc; $T_A = T_L$ to T_H.
2. Use only AT-cut crystals.
3. Minimum oscillator frequency with RC oscillator option is limited only by size of external R and C and leakage of external C.
4. Including processing tolerances and variations in temperature and supply voltage. Excluding tolerances of external R and C.
5. The 2-bit timer prescaler is the limiting factor in determining timer resolution.
6. The minimum period t_{ILIL} or t_{IHIH} should not be less than the number of cycles it takes to execute the interrupt service routine plus 19 t_{CYC}.

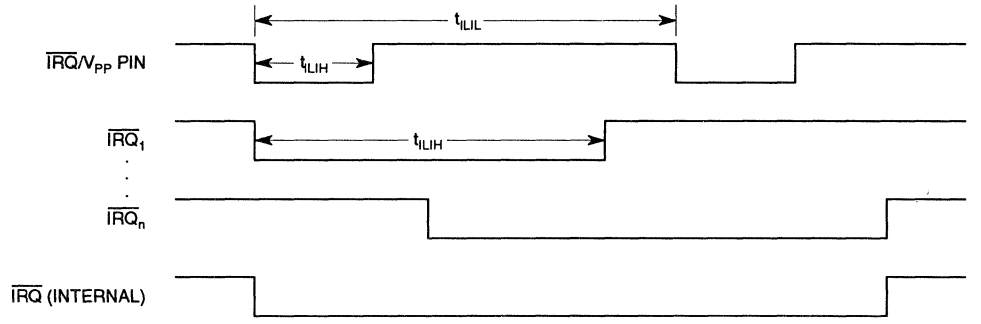
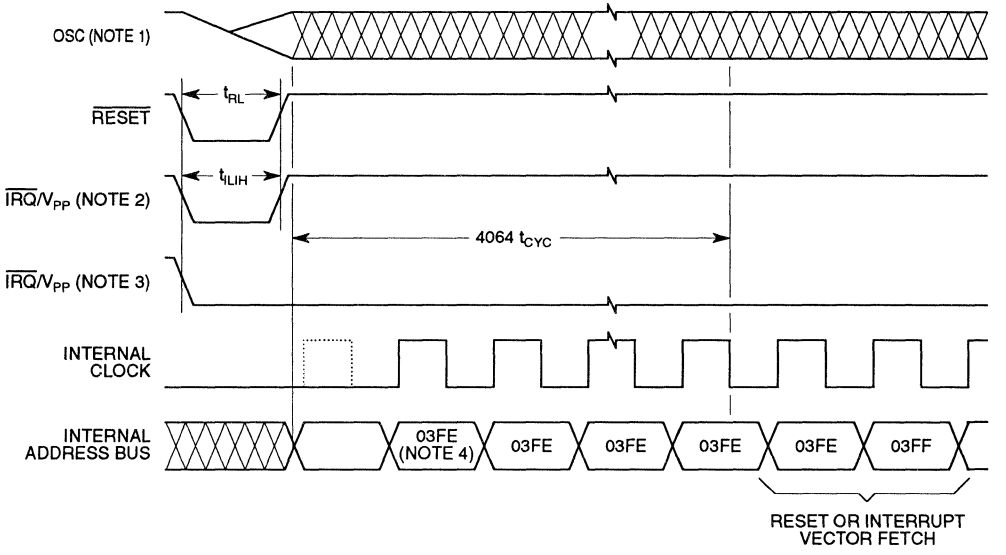


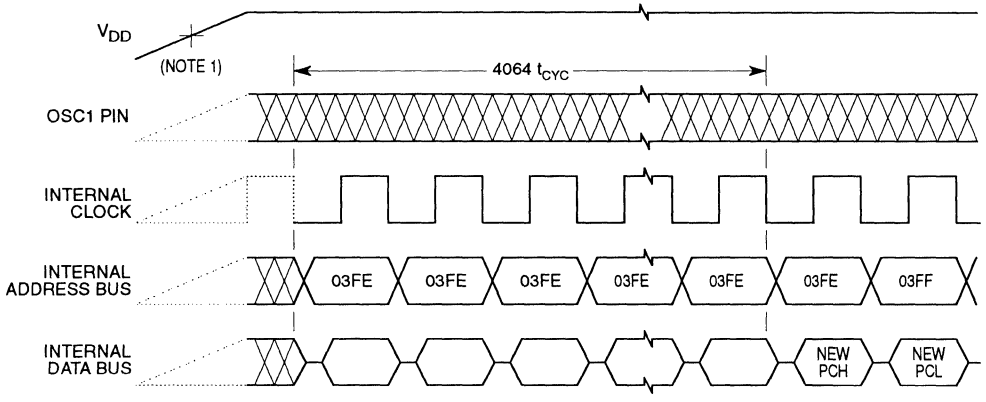
Figure 12-7. External Interrupt Timing



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.

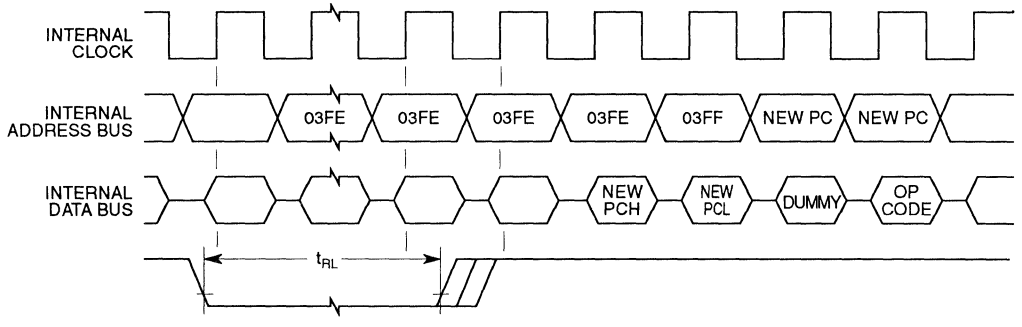
Figure 12-8. Stop Mode Recovery Timing



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 12-9. 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 12-10. External Reset Timing

12.7 Typical Oscillator Characteristics

Table 12-7. Typical Oscillator Characteristics

Parameter		V_{DD} = 3.0 V	V_{DD} = 5.0 V	Units
Oscillator Type	**Nominal Frequency**			
Frequency Variation (Part-to-Part)				
2-Pin RC Oscillator	2 MHz	±12	±7	%
3-Pin RC Oscillator	1 MHz	±5	±4	
Frequency Variation with Temperature				
2-Pin RC Oscillator	2 MHz	–2100	–1600	ppm/°C
3-Pin RC Oscillator	1 MHz	–1100	–1100	
Frequency Variation with Supply Voltage				
2-Pin RC Oscillator	2 MHz	±1.0	±0.2	Δf/ΔV
3-Pin RC Oscillator	1 MHz	±0.3	±0.1	
Cumulative Frequency Variations(1)				
2-Pin RC Oscillator	2 MHz	±36	±20	%
3-Pin RC Oscillator	1 MHz	±16	±13	

1. V_{DD} ±10%; T_A = –40 °C to +85 °C.

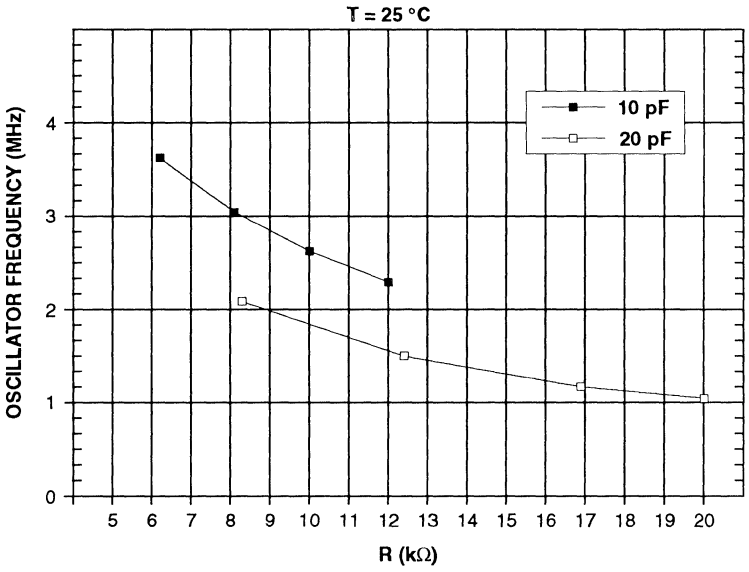


Figure 12-11. 2-Pin RC Oscillator R vs Frequency (V_{DD} = 5.0 V)

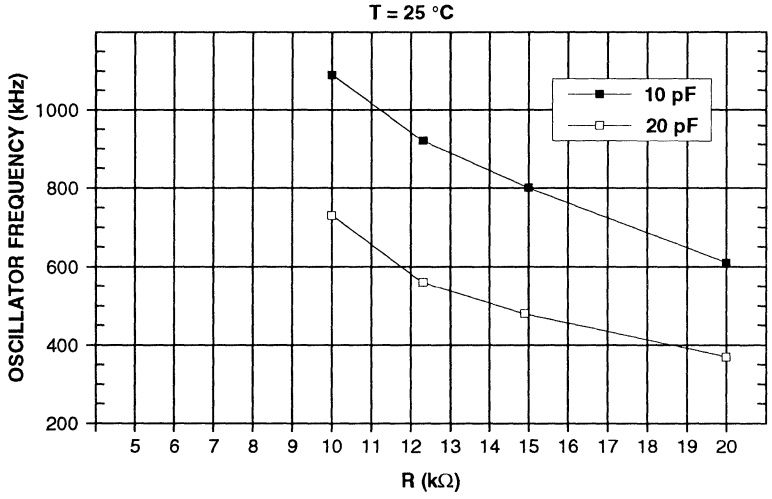


Figure 12-12. 3-Pin RC Oscillator R vs Frequency (V_{DD} = 5.0 V)

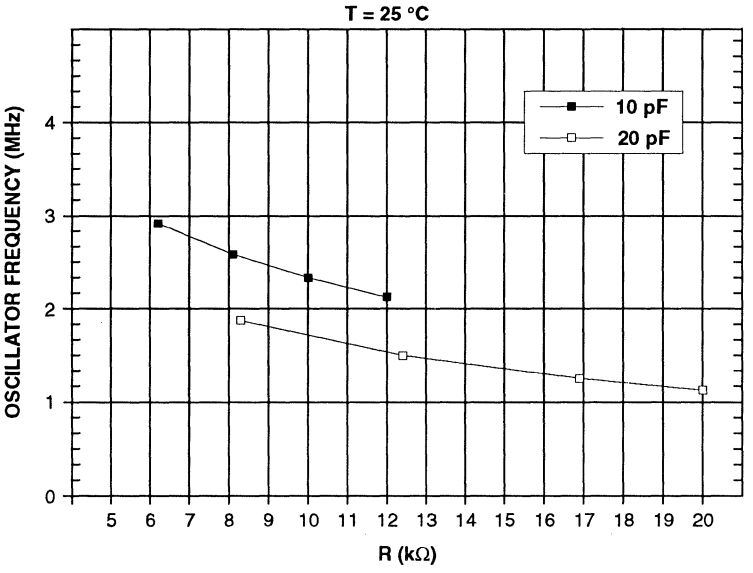


Figure 12-13. 2-Pin Oscillator R vs Frequency (V_{DD} = 3.0 V)

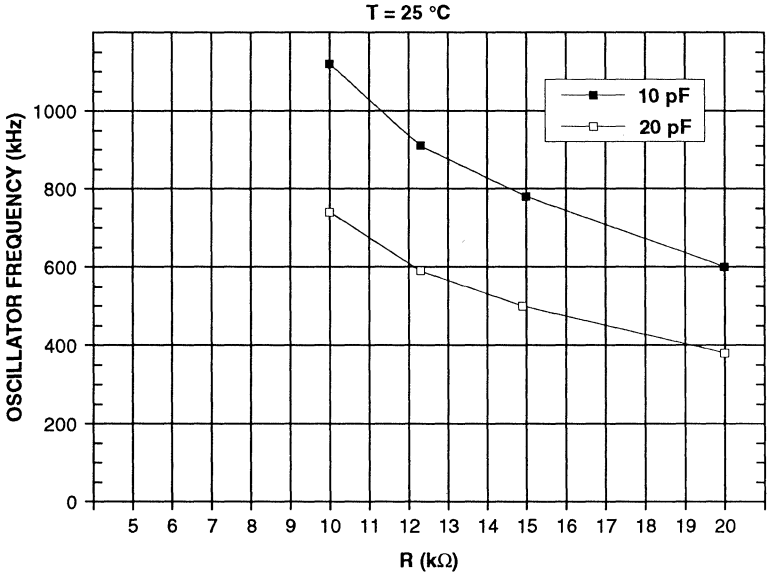


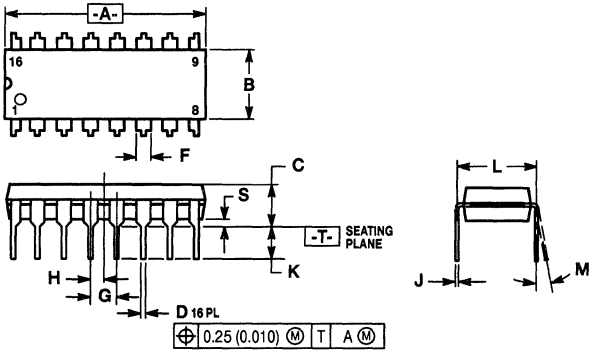
Figure 12-14. 3-Pin Oscillator R vs Frequency (V_{DD} = 3.0 V)

SECTION 13
MECHANICAL SPECIFICATIONS

This section gives the dimensions of the PDIP, SOIC, and Cerdip packages.

13.1 Plastic Dual In-Line Package (PDIP)

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



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

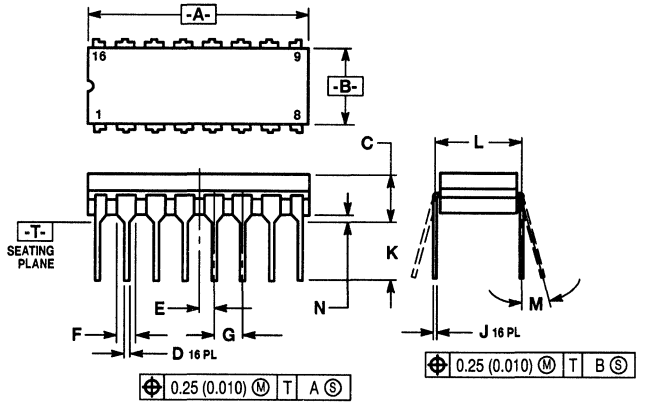
	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.740	0.770	18.80	19.55
B	0.250	0.270	6.35	6.85
C	0.145	0.175	3.69	4.44
D	0.015	0.021	0.39	0.53
F	0.040	0.070	1.02	1.77
G	0.100 BSC		2.54 BSC	
H	0.050 BSC		1.27 BSC	
J	0.008	0.015	0.21	0.38
K	0.110	0.130	2.80	3.30
L	0.295	0.305	7.50	7.74
M	0°	10°	0°	10°
S	0.020	0.040	0.51	1.01

CASE 648-08
ISSUE R

DATE 05/18/88

Figure 13-1. MC68HC705K1P

13.2 Small Outline Integrated Circuit (SOIC)



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

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. DIM F MAY NARROW TO 0.76 (0.030) WHERE THE LEAD ENTERS THE CERAMIC BODY.

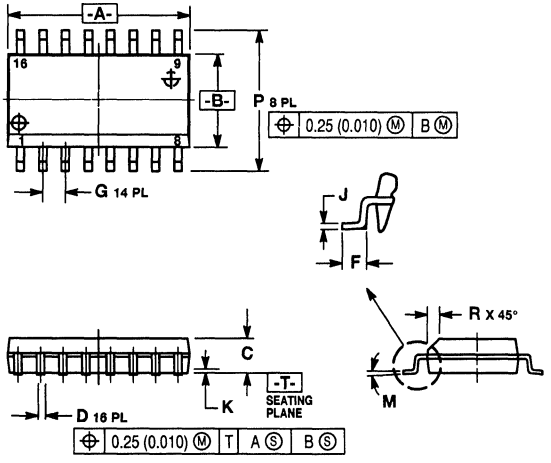
	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.750	0.770	19.05	19.55
B	0.240	0.290	6.10	7.36
C	—	0.165	—	4.19
D	0.015	0.021	0.39	0.53
E	0.050 BSC		1.27 BSC	
F	0.055	0.070	1.40	1.77
G	0.100 BSC		2.54 BSC	
J	0.009	0.011	0.23	0.27
K	—	0.200	—	5.08
L	0.300 BSC		7.62 BSC	
M	0°	15°	0°	15°
N	0.015	0.035	0.39	0.88

CASE 620-09
ISSUE V

DATE 11/05/87

Figure 13-2. MC68HC705K1DW

13.3 Ceramic DIP (Cerdip)



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

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	10.15	10.45	0.400	0.411
B	7.40	7.60	0.292	0.299
C	2.35	2.65	0.093	0.104
D	0.35	0.49	0.014	0.019
F	0.50	0.90	0.020	0.035
G	1.27 BSC		0.050 BSC	
J	0.25	0.32	0.010	0.012
K	0.10	0.25	0.004	0.009
M	0°	7°	0°	7°
P	10.05	10.55	0.395	0.415
R	0.25	0.75	0.010	0.029

CASE 751G-02
ISSUE A

DATE 06/22/93

Figure 13-3. MC68HC705K1S

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

MOTOROLA

1ATX31062-2 PRINTED IN USA 10/93 IMPERIAL LITHO 94189 10,000 MCU YGACAA

MC68HC705K1/D