MC68HC705K1/D
Rev. 2.0

HC05

MC68HC705K1

HCMOS Microcontroller Unit

TECHNICAL DATA

Technical Data

List of Sections

List of Sections

Table of Contents

Section 1. General Description

Section 2. Memory

Section 3. Central Processor Unit (CPU)

Section 4. Interrupts

Section 5. Resets

Section 6. Low-Power Modes

Section 7. Parallel Input/Output (I/O)

Section 8. Multifunction Timer

Section 9. EPROM/OTPROM

Section 10. Personality EPROM (PEPROM)

Section 11. Instruction Set

Section 12. Electrical Specifications

Section 13. Mechanical Specifications

Section 13. Ordering Information

List of Figures

List of Figures

List of Tables

List of Tables

Section 1. General Description

1.1 Contents

1.2 Introduction

The MC68HC705K1 is a member of the low-cost, high-performance M68HC05 Family of 8-bit microcontroller units (MCU). 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 read-only memory (EPROM). In packages without the transparent window for EPROM erasure, the 504 EPROM bytes serve as one-time programmable read-only memory (OTPROM).

1.3 Features

Features of the MCU include:

- Popular M68HC05 CPU
- Memory-mapped input/output (I/O) registers
- 504 bytes of EPROM/OTPROM, including eight user vector locations
- 32 bytes of user random-access memory (RAM)
- 64-bit personality EPROM
- 10 bidirectional I/O pins with these features:
  - Software programmable pulldown devices
  - Four I/O pins with 8-mA current sinking capability
  - Four 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:
  - Crystal or ceramic resonator
  - Mask-optional 2-pin or 3-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)

1.4 Mask Options

These MC68HC705K1 options are programmable in the mask option register (MOR):

- 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
- 2-pin RC-driven oscillator or 3-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.

1.5 MCU Structure

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

General Description

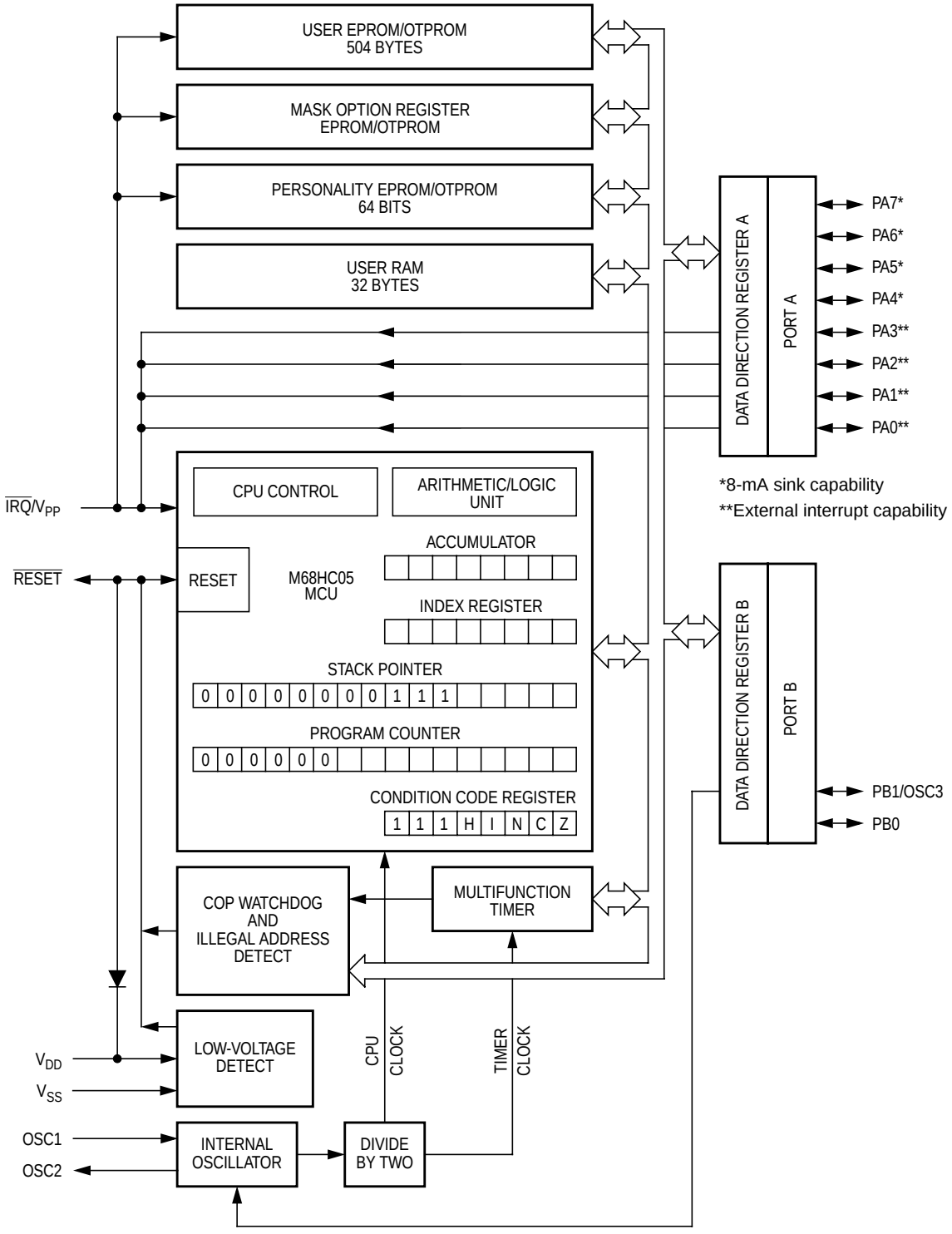


Figure 1-1. MC68HC705K1 Block Diagram

1.6 Pin Assignments

Figure 1-2 shows the MC68HC705K1 pin assignments.

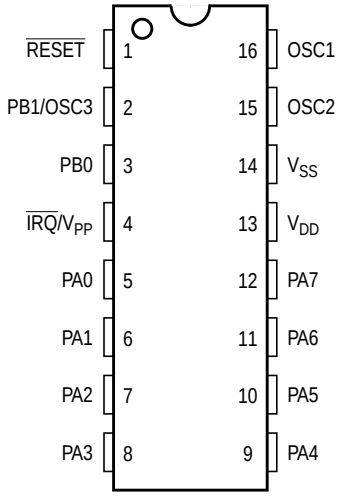


Figure 1-2. Pin Assignments

1.6.1 V_{DD} and V_{SS}

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

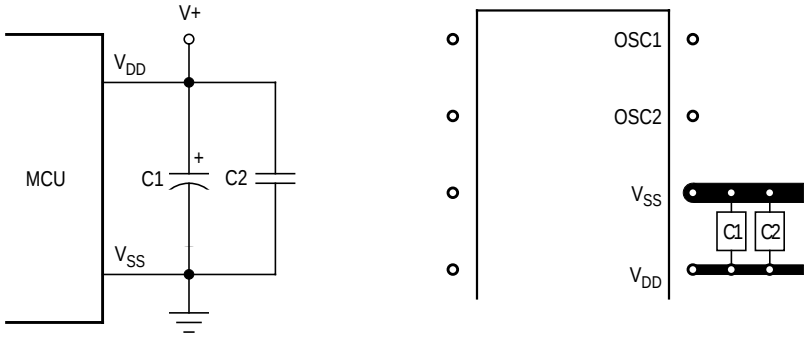


Figure 1-3. Bypassing Layout Recommendation

1.6.2 OSC1 and OSC2

The OSC1, OSC2, and PB1/OSC3 pins are the control connections for the 2-pin or 3-pin on-chip oscillator. The oscillator can be driven by any of these:

- 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}.

1.6.2.1 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.

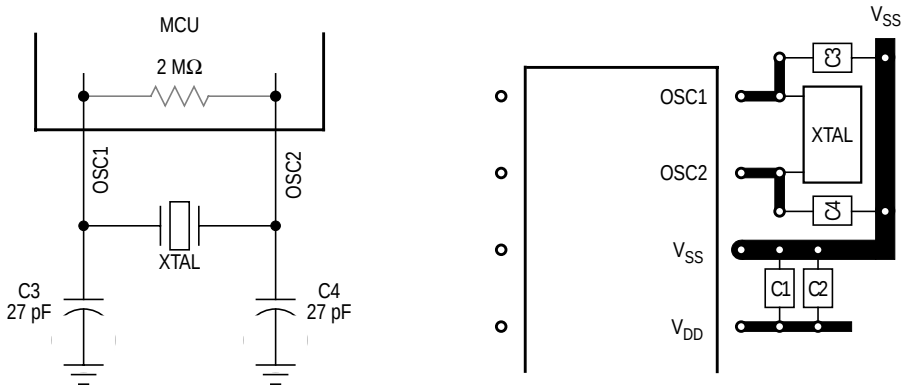


Figure 1-4. Crystal Connections

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 **9.6 Mask Option Register**. Clearing the RC bit connects an internal 2-MΩ startup resistor between OSC1 and OSC2.

1.6.2.2 Ceramic Resonator

To reduce cost, use a ceramic resonator in place of the crystal. Use the circuit in **Figure 1-5** for a 2-pin ceramic resonator or **Figure 1-6** for a 3-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.

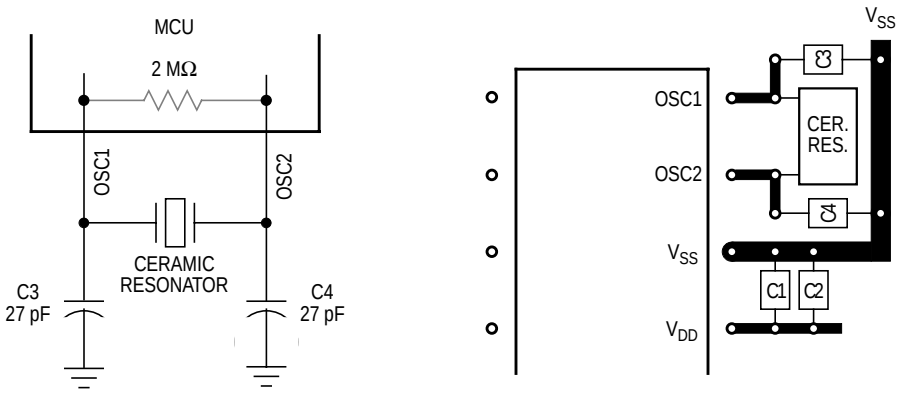


Figure 1-5. 2-Pin Ceramic Resonator Connections

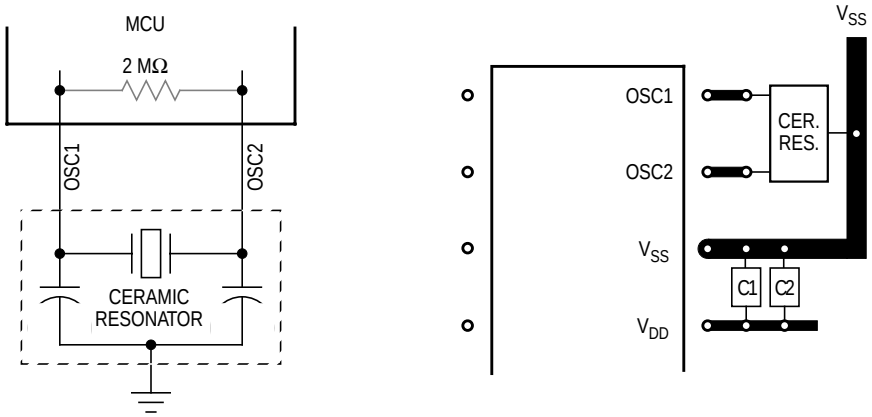


Figure 1-6. 3-Pin Ceramic Resonator Connections

To use the resonator-driven oscillator, the RC bit in the mask option register must be clear. See **9.6 Mask Option Register**. Clearing the RC bit connects an internal 2-MΩ startup resistor between OSC1 and OSC2.

1.6.2.3 2-Pin RC Oscillator

For maximum cost reduction, use the 2-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 2-pin oscillator configuration is 2 MHz.

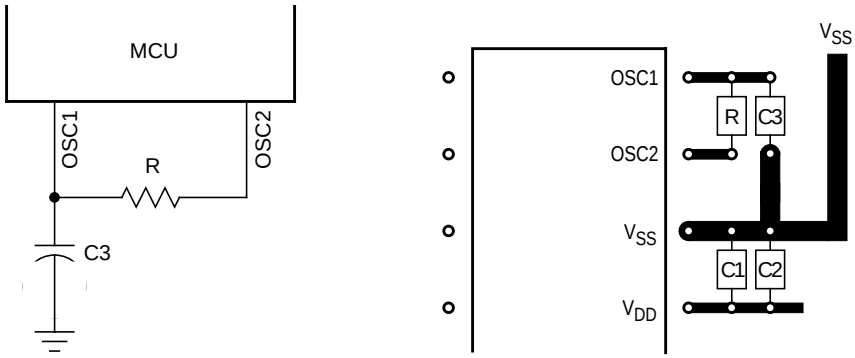


Figure 1-7. 2-Pin RC Oscillator Connections

To use the 2-pin RC oscillator configuration, the RC bit in the mask option register must be programmed to a logic 1. Setting the RC bit

disconnects the internal startup resistor. The PIN3 bit in the mask option register must remain erased (logic 0). The PIN3 bit selects the 3-pin RC oscillator configuration. See **9.6 Mask Option Register**.

1.6.2.4 3-Pin RC Oscillator

Another low-cost option is the 3-pin RC oscillator configuration shown in **Figure 1-8**. The 3-pin oscillator is more stable than the 2-pin oscillator. The OSC2 and PB1/OSC3 signals are square waves, and the signal on OSC1 is a triangular wave. The 3-pin RC oscillator configuration is recommended for frequencies of 1 MHz and less.

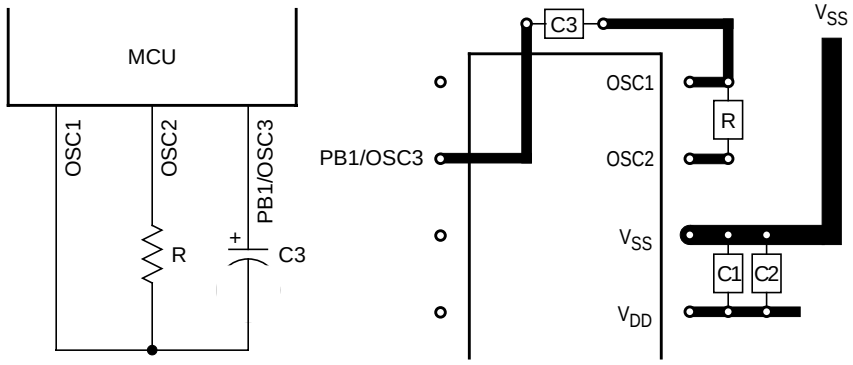


Figure 1-8. 3-Pin RC Oscillator Connections

To use the 3-pin RC oscillator configuration, both the RC and PIN3 bits in the mask option register must be programmed to logic 1s. See **9.6 Mask Option Register**.

NOTE: *In 3-pin RC oscillator configurations, the personality EPROM (PEPROM) cannot be programmed by user software. If the voltage on $\overline{IRQ}/V_{PP}$ is raised above V_{DD}, the oscillator will revert to a 2-pin oscillator configuration and device operation will be disrupted.*

1.6.2.5 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.

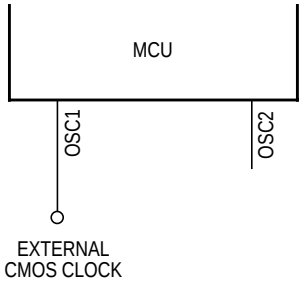


Figure 1-9. External Clock Connections

1.6.3 $\overline{\text{RESET}}$

A logic 0 on the $\overline{\text{RESET}}$ pin forces the MCU to a known startup state. See **5.3.2 External Reset** for more information.

1.6.4 $\overline{\text{IRQ}}/V_{PP}$

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

- Applying asynchronous external interrupt signals, see **4.3.2 External Interrupts**
- Applying the EPROM/OTPROM programming voltage, see **9.4 EPROM/OTPROM Programming**

1.6.5 PA7–PA0

PA7–PA0 are the pins of port A, a general-purpose, bidirectional I/O port. See **7.4 Port A**.

1.6.6 PB1/OSC3 and PB0

PB1/OSC3 and PB0 are the pins of port B, a general-purpose, bidirectional I/O port. See **7.5 Port B**.

Section 2. Memory

2.1 Contents

2.2 Introduction

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

2.3 Memory Map

The central processor unit (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 erasable, programmable read-only memory (EPROM) portion of memory holds the program instructions, fixed data, user-defined vectors, and interrupt service routines. The random-access memory (RAM) portion of memory holds variable data. Input/output (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 microcontroller unit (MCU). Refer to **Figure 2-2** for a more detailed memory map of the 32-byte I/O register section.

2.4 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**.

2.5 Random-Access Memory (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.*

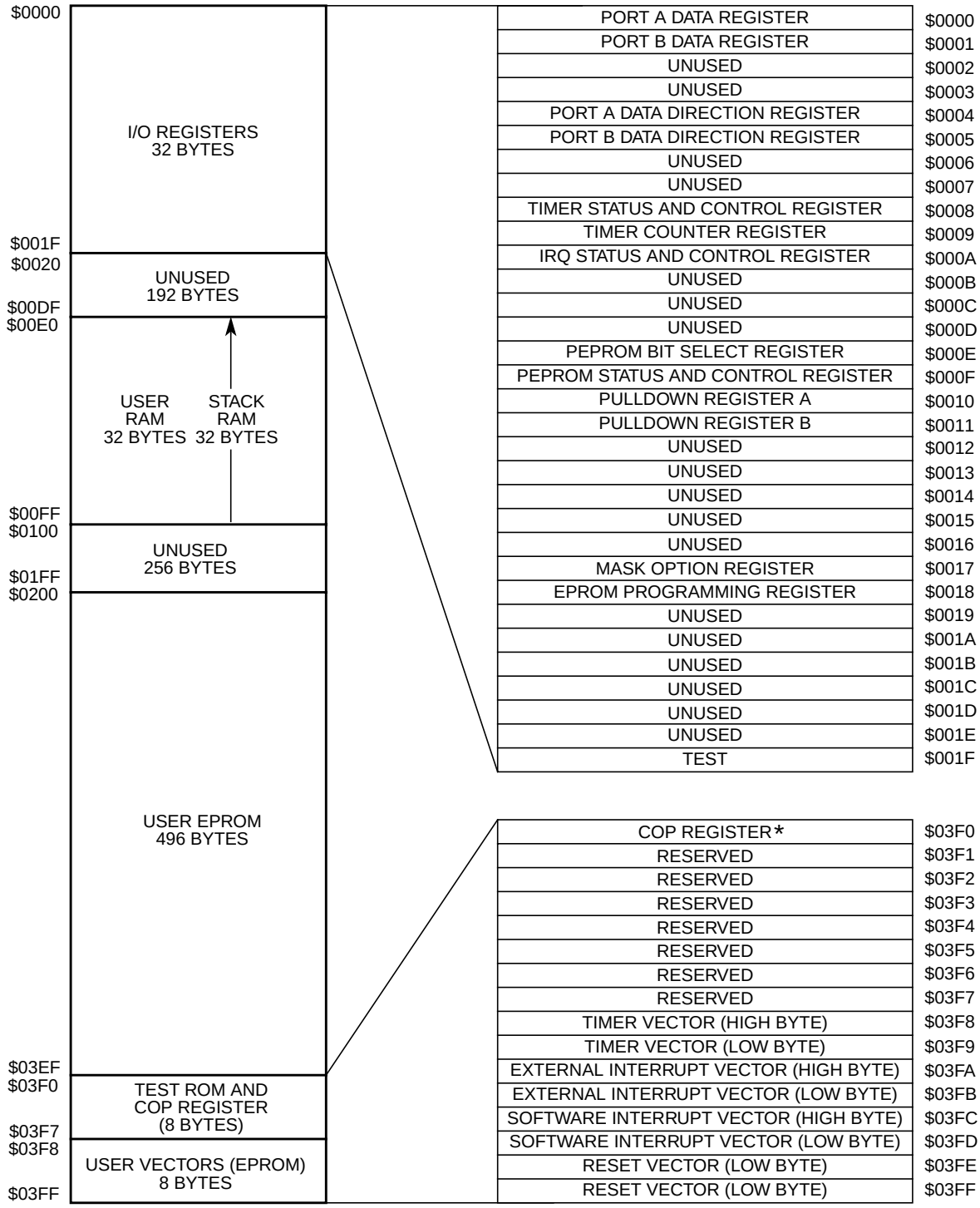


Figure 2-1. Memory Map

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PORTA) See page 64.	Read:	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
		Write:								
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PORTB) See page 69.	Read:	0	0	0	0	0	0	PB1/ OSC3	PB0
		Write:								
		Reset:	Unaffected by reset							
$0002	Unimplemented									
$0003	Unimplemented									
$0004	Data Direction Register A (DDRA) See page 65.	Read:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0005	Data Direction Register B (DDRB) See page 70.	Read:	0	0	0	0	0	0	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0006	Unimplemented									
$0007	Unimplemented									
$0008	Timer Status and Control Register (TSCR) See page 77.	Read:	TOF	RTIF	TOIE	RTIE			RT1	RT0
		Write:					TOFR	RTIFR		
		Reset:	0	0	0	0	U	U	1	1
$0009	Timer Counter Register (TCNTR) See page 79.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000A	IRQ Status and Control Register (ISCR) See page 45.	Read:	IRQE	0	0	0	IRQF	0		0
		Write:							IRQR	
		Reset:	1	0	0	0	0	0	U	0

= Unimplemented R = Reserved U = Unaffected

Figure 2-2. I/O Register Summary (Sheet 1 of 3)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000B	Unimplemented									
↓										
$000D	Unimplemented									
$000E	PEPROM Bit Select Register (PEBSR) See page 93.	Read: / Write:	PEB7	PEB6	PEB5	PEB4	PEB3	PEB2	PEB1	PEB0
		Reset:	0	0	0	0	0	0	0	0
$000F	PEPROM Status and Control Register (PESCR) See page 95.	Read:	PEDATA	0	PEPGM	0	0	0	0	PEPRZF
		Write:								
		Reset:	U	0	0	0	0	0	0	1
$0010	Pulldown Register A (PDRA) See page 66.	Read:								
		Write:	PDIA7	PDIA6	PDIA5	PDIA4	PDIA3	PDIA2	PDIA1	PDIA0
		Reset:	0	0	0	0	0	0	0	0
$0011	Pulldown Register B (PDRB) See page 71.	Read:								
		Write:	0	0	0	0	0	0	PDIB1	PDIB0
		Reset:	0	0	0	0	0	0	0	0
$0012	Unimplemented									
↓										
$0016	Unimplemented									
$0017	Mask Option Register (MOR) See page 87.	Read: / Write:	SWPDI	PIN3	RC	SWAIT	LVRE	PIRQ	LEVEL	COPEN
		Reset:	Unaffected by reset							
$0018	EPROM Programming Register (EPROG) See page 84.	Read: / Write:	R	R	R	R	R	ELAT	MPGM	EPGM
		Reset:	U	U	U	U	U	0	0	0

= Unimplemented R = Reserved U = Unaffected

Figure 2-2. I/O Register Summary (Sheet 2 of 3)

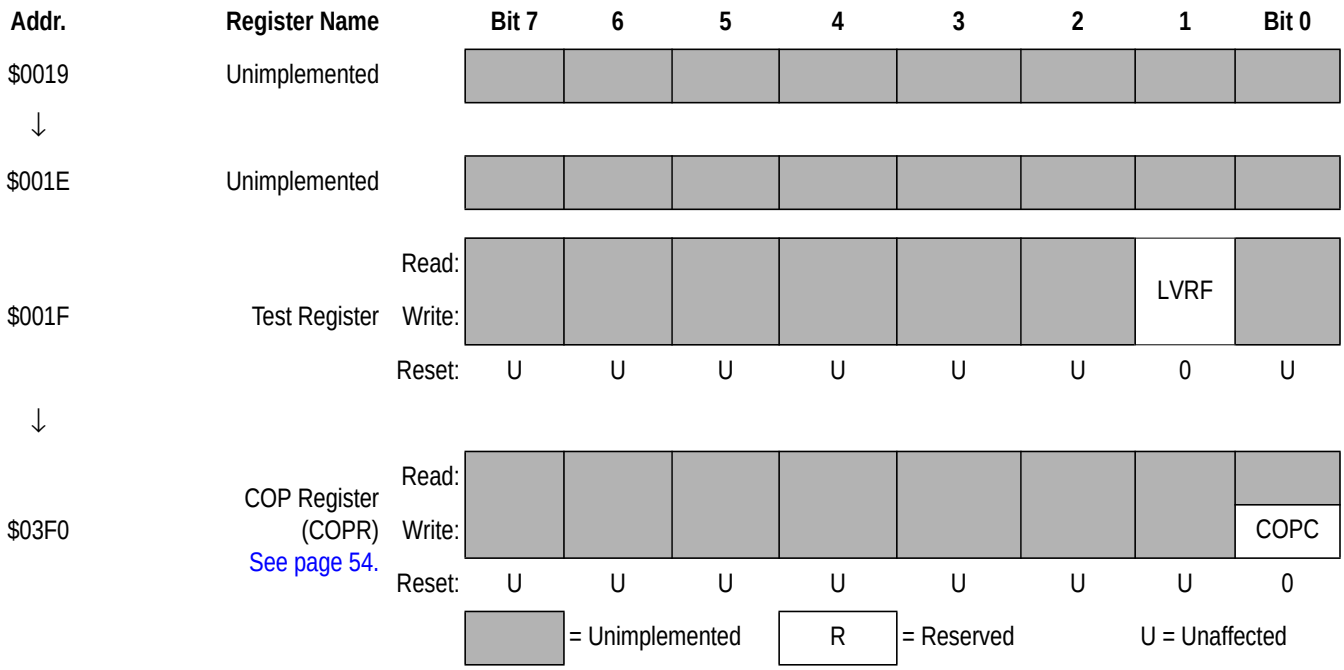


Figure 2-2. I/O Register Summary (Sheet 3 of 3)

2.6 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.

2.7 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 OTPROM.

Section 3. Central Processor Unit (CPU)

3.1 Contents

3.2 Introduction

This section describes the central processor unit (CPU) registers.

3.3 CPU Registers

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

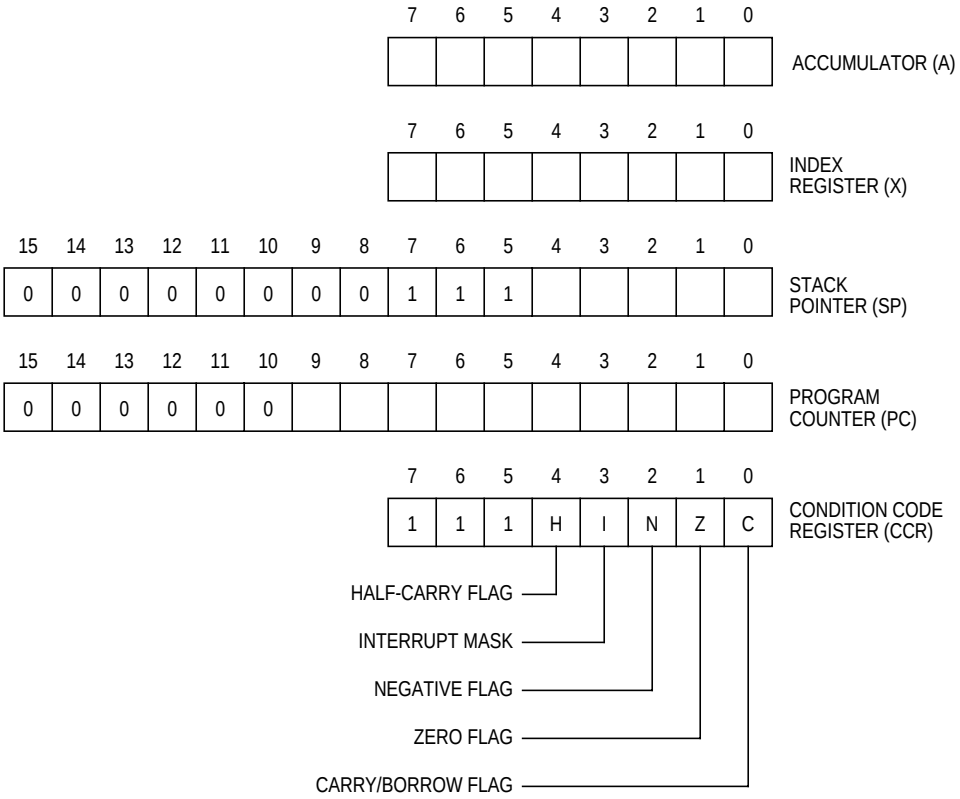


Figure 3-1. Programming Model

3.3.1 Accumulator

The accumulator (A) shown in **Figure 3-2** is a general-purpose 8-bit register. The CPU uses the accumulator to hold operands and results of arithmetic and non-arithmetic operations.

Figure 3-2. Accumulator (A)

3.3.2 Index Register

In the indexed addressing modes, the CPU uses the byte in the index register (X) shown in **Figure 3-3** to determine the conditional address of the operand. See **11.3.5 Indexed, No Offset**, **11.3.6 Indexed, 8-Bit Offset**, and **11.3.7 Indexed, 16-Bit Offset** for more information on indexed addressing.

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

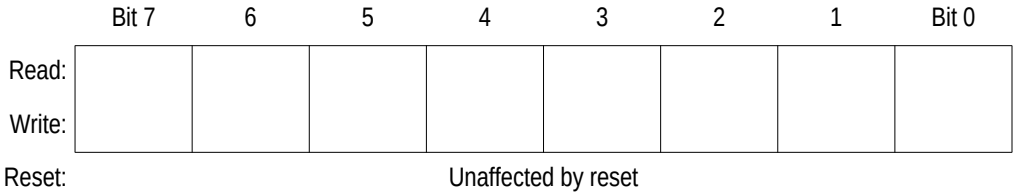


Figure 3-3. Index Register (X)

3.3.3 Stack Pointer

The stack pointer (SP) shown in Figure 3-4 is a 16-bit register that contains the address of the next free location on the stack. During a reset or after the reset stack pointer (RSP) instruction, the stack pointer 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.

The 11 most significant bits of the stack pointer are fixed permanently 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.

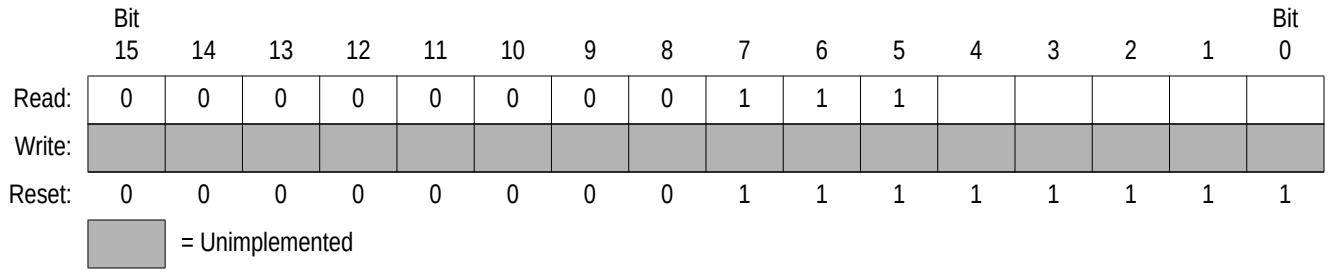


Figure 3-4. Stack Pointer (SP)

3.3.4 Program Counter

The program counter (PC) shown in **Figure 3-5** 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.

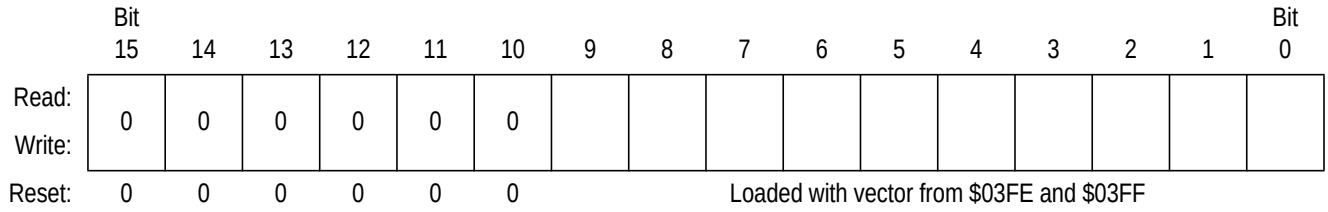


Figure 3-5. Program Counter (PC)

3.3.5 Condition Code Register

The condition code register (CCR) shown in **Figure 3-6** 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.

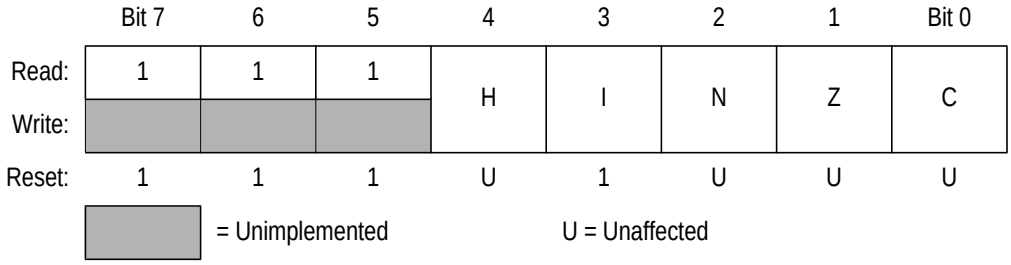


Figure 3-6. Condition Code Register (CCR)

H — Half-Carry Flag

The CPU sets the half-carry flag when a carry occurs between bits 3 and 4 of the accumulator during an add without carry (ADD) or add with carry (ADC) operation. The half-carry bit is required for binary-coded decimal (BCD) arithmetic operations. Reset has no affect on the half-carry flag.

I — Interrupt Mask Flag

Setting the interrupt mask (I) disables interrupts. If an interrupt request occurs while the interrupt mask is a logic 0, 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 clear interrupt mask bit (CLI), STOP, or WAIT instruction.

N — Negative Flag

The CPU sets the negative flag when an arithmetic operation, logical operation, or data manipulation produces a negative result (bit 7 in the results is a logic 1). Reset has no effect on the negative flag.

Z — Zero Flag

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.

C — Carry/Borrow Flag

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

3.4 Arithmetic/Logic Unit (ALU)

The arithmetic/logic unit (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 requires 11 internal clock cycles to complete this chain of operations.

Section 4. Interrupts

4.1 Contents

4.2 Introduction

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

4.3 Interrupt Types

These conditions generate interrupts:

- SWI instruction (software interrupt)
- A logic 0 applied to the $\overline{IRQ}/V_{PP}$ pin (external interrupt)
- A logic 1 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 central processor unit (CPU) registers on the stack and loads the program counter with a user-defined vector address.

4.3.1 Software Interrupt

The software interrupt (SWI) instruction causes a non-maskable interrupt.

4.3.2 External Interrupts

These 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.

4.3.2.1 $\overline{IRQ}/V_{PP}$ Pin

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

- IRQ latch
- IRQE bit in the interrupt status and control register
- 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 (MOR). See **9.6 Mask Option Register**.

Programming the LEVEL bit to a logic 1 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 0 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 1 and then falls again to logic 0.

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

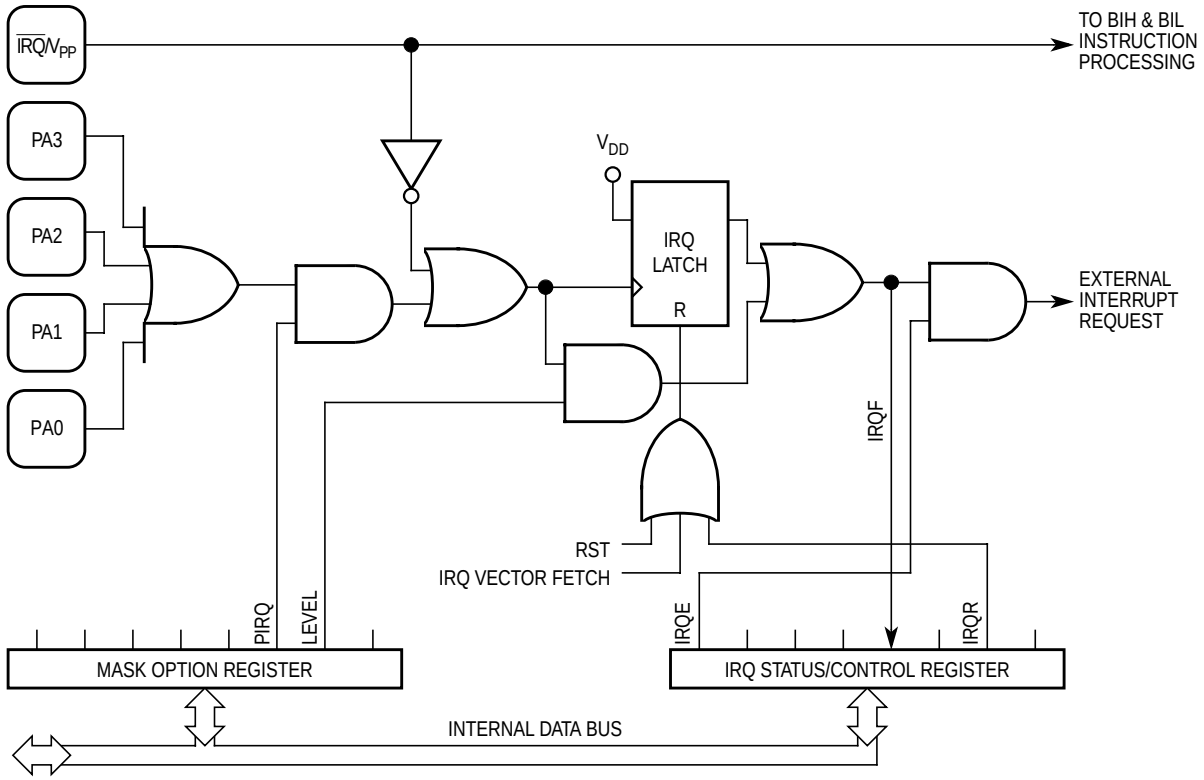


Figure 4-1. External Interrupt Logic

4.3.2.2 PA3–PA0 Pins

Programming the PIRQ bit in the mask option register to a logic 1 enables pins PA3–PA0 to serve as additional external interrupt sources. See **9.6 Mask Option Register**. An interrupt signal on a PA3–PA0 pin latches an external interrupt request. After completing the current instruction, the CPU tests these bits:

- IRQ latch
- IRQE bit in the IRQ status and control register
- 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 1 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 0 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 0 and then rises again to a logic 1.
- 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 1 and then falls again to a logic 0.

4.3.2.3 IRQ Status and Control Register

The IRQ status and control register (ISCR) contains an external interrupt mask, an external interrupt flag, and a flag reset bit. Unused bits read as logic 0s.

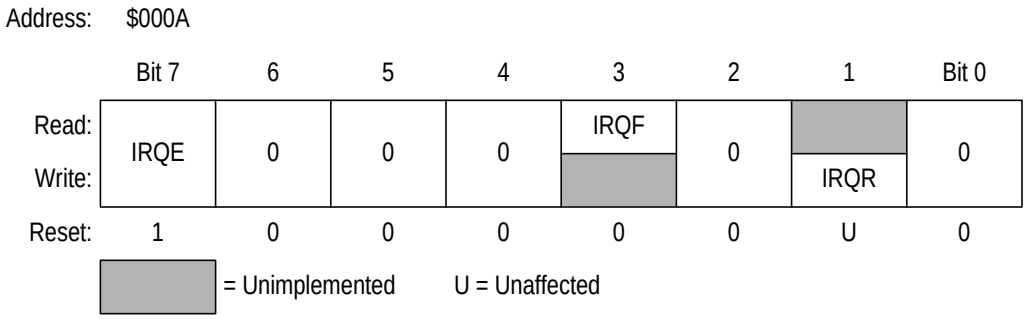


Figure 4-2. IRQ Status and Control Register (ISCR)

IRQE — External Interrupt Request Enable Bit

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

These conditions set the IRQF bit:

a. An external interrupt signal on the $\overline{IRQ}/V_{PP}$ pin

b. 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 1 to the IRQR bit clears the IRQF bit.

IRQR — Interrupt Request Reset Bit

Writing a logic 1 to this write-only bit clears the IRQF bit. Writing a logic 0 to IRQR has no effect. Reset has no effect on IRQR.

1 = IRQF bit cleared
0 = No effect

4.3.3 Timer Interrupts

The multifunction timer can generate these interrupts:

- Timer overflow interrupt
- Real-time interrupt

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

4.3.3.1 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 **8.3 Timer Status and Control Register**.

4.3.3.2 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 **8.3 Timer Status and Control Register**.

4.4 Interrupt Processing

The CPU does these 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**.

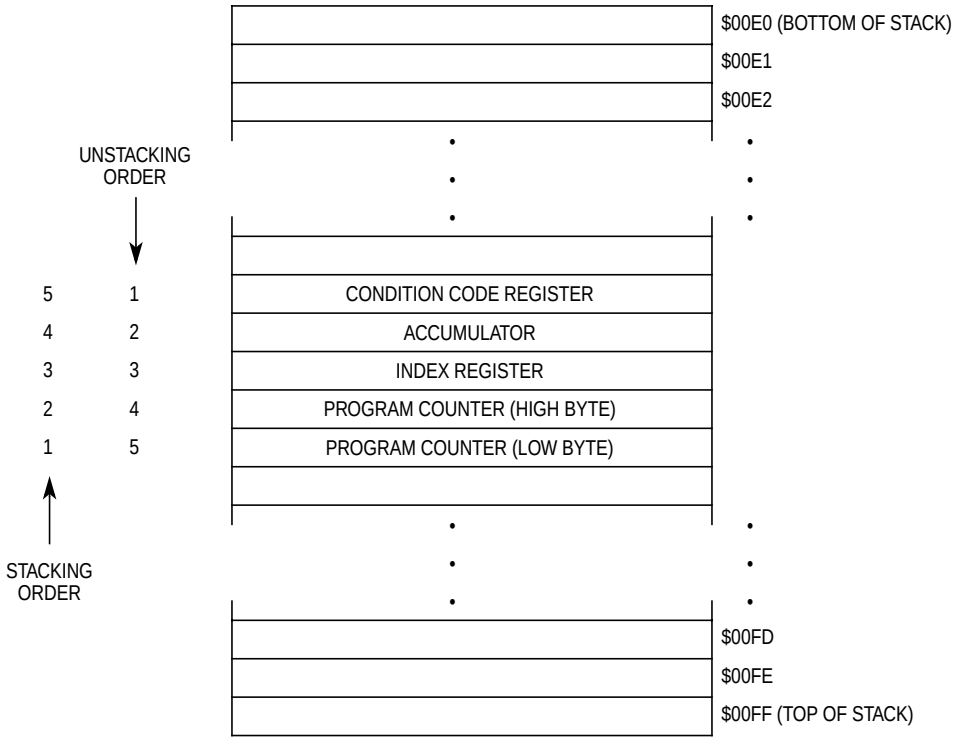


Figure 4-3. Interrupt Stacking Order

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			PIRQ3		
	PA1 pin			PIRQ3		
	PA0 pin			PIRQ3		
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.

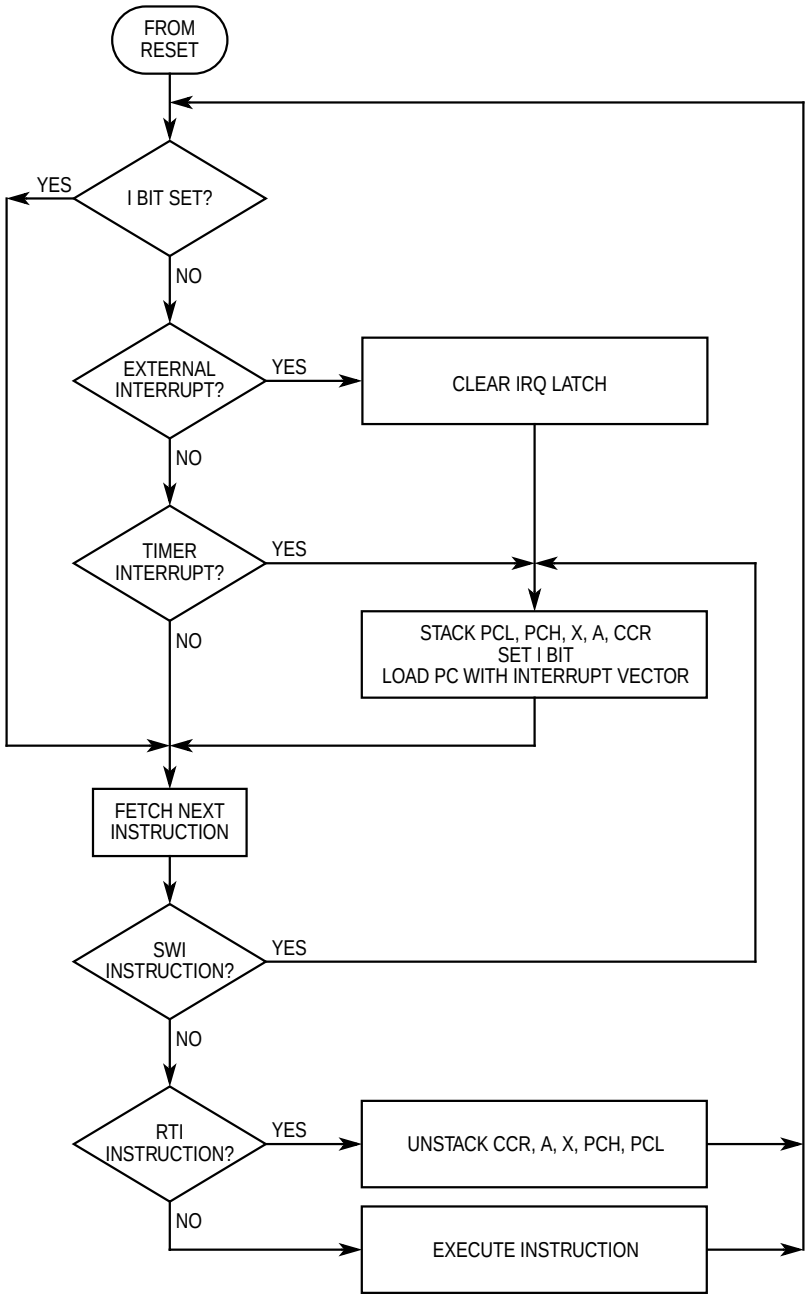


Figure 4-4. Interrupt Flowchart

Section 5. Resets

5.1 Contents

5.2 Introduction

This section describes the five reset sources and how they initialize the microcontroller unit (MCU).

5.3 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. These conditions produce a reset:

- Initial power-up (power-on reset)
- A logic 0 applied to the $\overline{\text{RESET}}$ pin (external reset)
- Timeout of the computer operating properly (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 volts (low-voltage reset)

5.3.1 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 0 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 1.

5.3.2 External Reset

A logic 0 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.

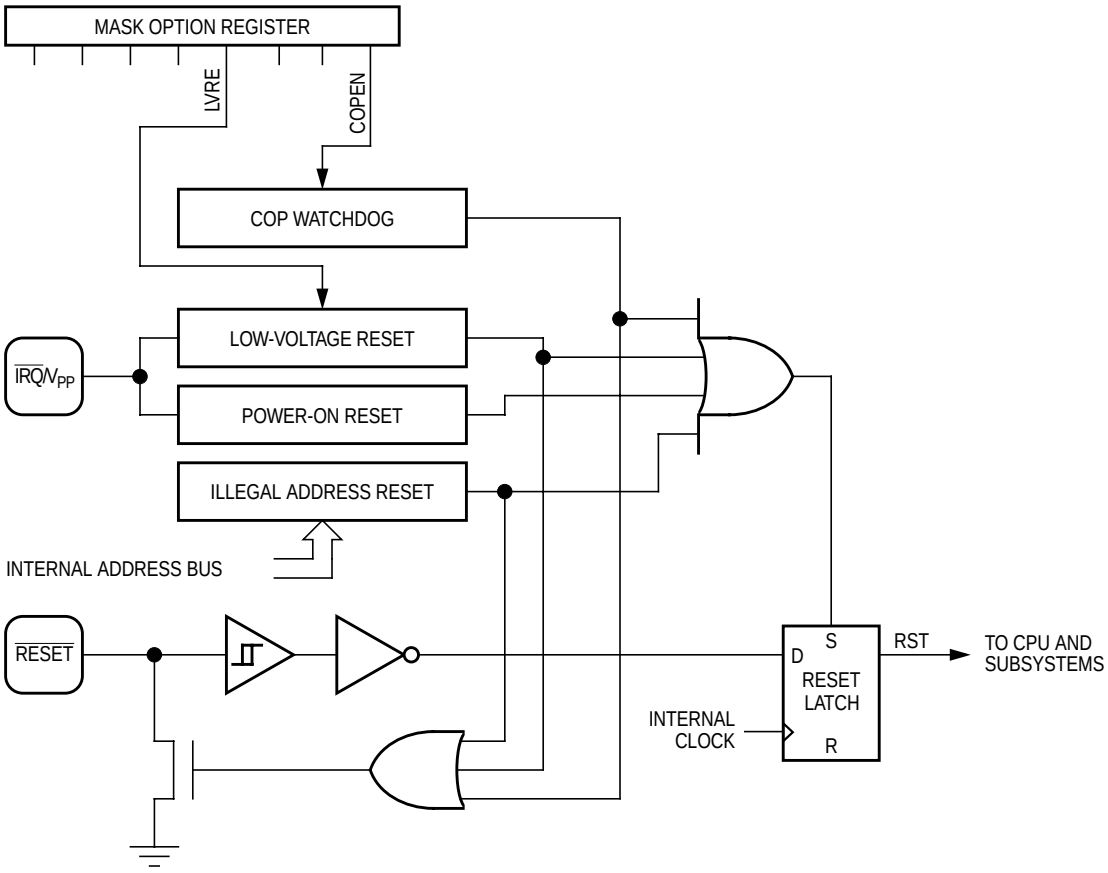


Figure 5-1. Reset Sources

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

5.3.3 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 **8.5 COP Watchdog**.) To clear the COP watchdog and prevent a COP reset, write a logic 0 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.

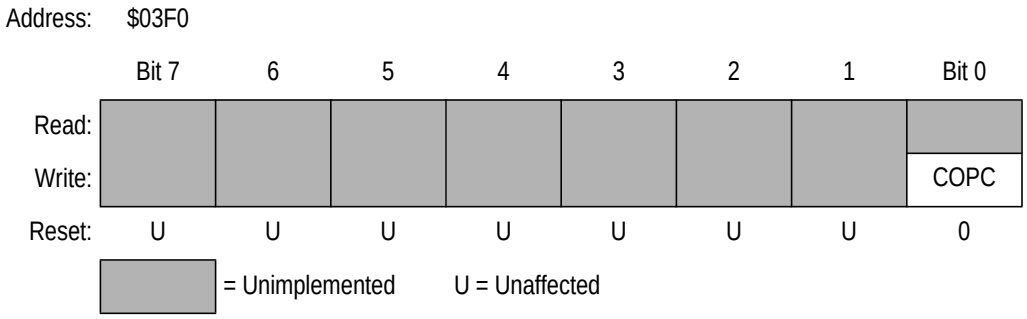


Figure 5-2. COP Register (COPR)

COPC — COP Clear Bit

COPC is a write-only bit. Periodically writing a logic 0 to COPC prevents the COP watchdog from resetting the MCU. Writing a logic 1 has no effect. Reset clears the COPC bit.

5.3.4 Illegal Address Reset

An opcode fetch from an address that is not in the erasable, programmable read-only memory (EPROM) (locations $0200–$03FF) or the random-access memory (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.

5.3.5 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 1 enables the low-voltage reset function. When erased, the LVRE bit in the mask option register disables the low-voltage reset circuit. See **9.6 Mask Option Register**.

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.

5.4 Reset States

This subsection describes how resets initialize the MCU.

5.4.1 CPU

A reset has these effects on the central processor unit (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

5.4.2 I/O Port Registers

A reset has these effects on input/output (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 0)
- 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 0)
- 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

5.4.3 Multifunction Timer

A reset has these effects on the multifunction timer:

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

5.4.4 COP Watchdog

A reset clears the COP watchdog timeout counter.

Section 6. Low-Power Modes

6.1 Contents

6.2 Introduction

This section describes the four low-power modes:

- Stop mode
- Wait mode
- Halt mode
- Data-retention mode

6.3 Stop Mode

If the SWAIT bit in the mask option register (MOR) is programmed to a logic 0, the STOP instruction puts the microcontroller unit (MCU) in its lowest power-consumption mode and has these 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 central processor unit (CPU) clock and the timer clock, including the computer operating properly (COP) watchdog

The STOP instruction does not affect any other registers or any input/output (I/O) lines.

These 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 1, 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 0 on the $\overline{RESET}$ 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.

6.4 Wait Mode

The WAIT instruction puts the MCU in an intermediate power-consumption mode and has these 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.

These conditions restart the CPU clock and bring the MCU out of wait 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 1, 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 0 on the $\overline{RESET}$ pin resets the MCU and loads the program counter with the contents of locations $03FE and $03FF.

6.5 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 1. 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.

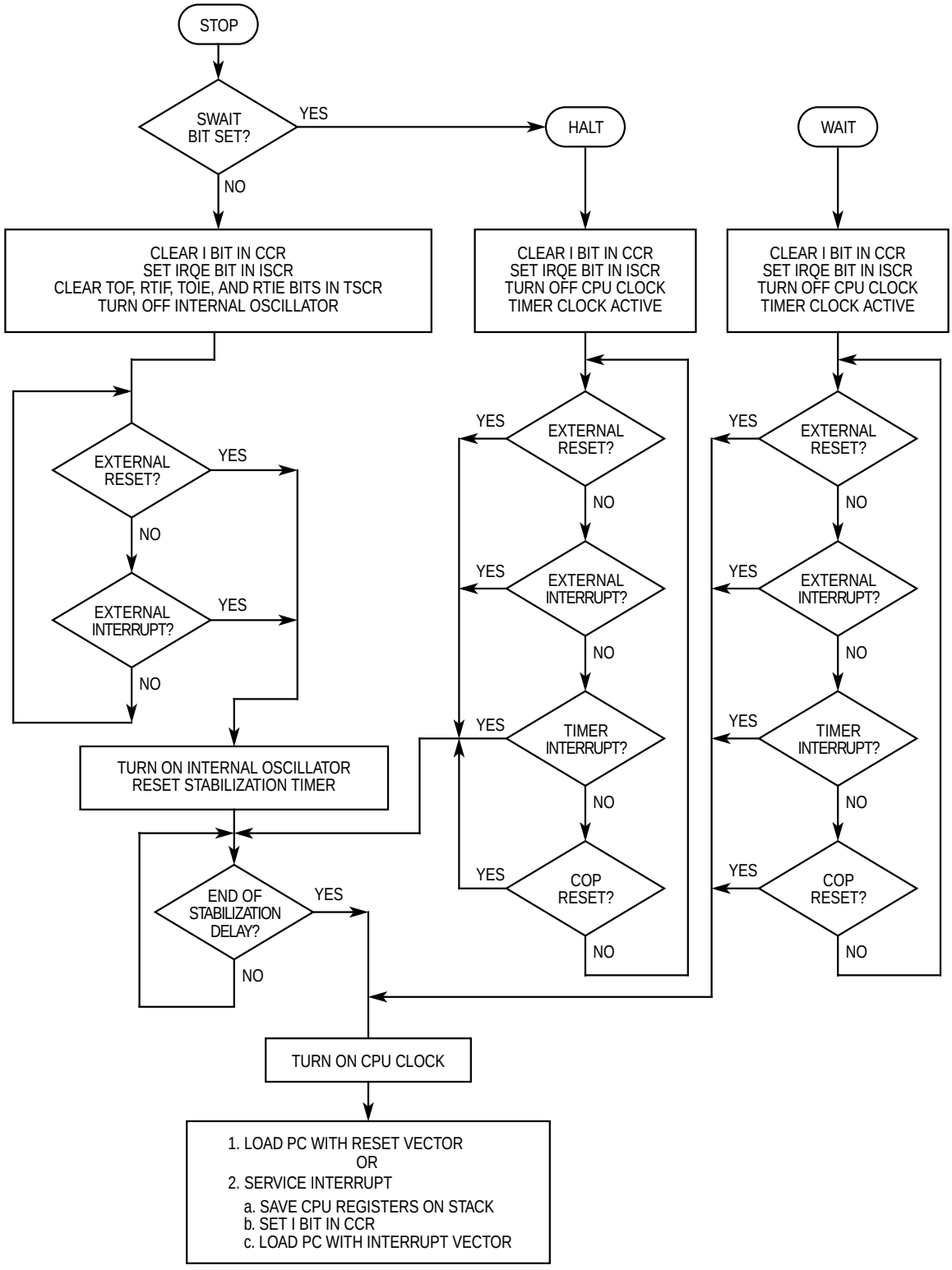


Figure 6-1. Stop/Wait/Halt Flowchart

6.6 Data-Retention Mode

In data-retention mode, the MCU retains random-access memory (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 0.
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 1.

Low-Power Modes

Section 7. Parallel Input/Output (I/O)

7.1 Contents

7.2 Introduction

This section describes the two bidirectional input/output (I/O) ports:

- Port A
- Port B

7.3 I/O Port Function

The 10 bidirectional input/output (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 10 I/O pins have software-programmable pulldown devices.

7.4 Port A

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

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

7.4.1 Port A Data Register

The port A data register (PORTA) 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.

Address: $0000

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	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.

7.4.2 Data Direction Register A

The contents of data direction register A (DDRA) determine whether each port A pin is an input or an output. Writing a logic 1 to a DDRA bit enables the output buffer for the associated port A pin; a logic 0 disables the output buffer. A reset initializes all DDRA bits to logic 0s, configuring all port A pins as inputs. If the pulldown devices are enabled, setting a DDRA bit to a logic 1 turns off the pulldown device for that pin.

Address: $0004

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	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 0 to logic 1.*

7.4.3 Pulldown Register A

Programming the SWPDI bit in the mask option register to a logic 0 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 (PDRA).

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 0, reset initializes all port A pins as inputs with pulldown devices turned on.

Programming the SWPDI bit to a logic 1 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 1.

Address: $0010

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

= Unimplemented

Figure 7-3. Pulldown Register A (PDRA)

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

Writing logic 0s 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 1 to logic 0.*

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

7.4.4 Port A External Interrupts

Programming the PIRQ bit in the mask option register to a logic 1 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 1 or a rising edge. The active interrupt state for the $\overline{IRQ}/V_{PP}$ pin is a logic 0 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 branch if interrupt pin is high (BIH) and branch if interrupt pin is low (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.*

7.4.5 Port A Logic

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

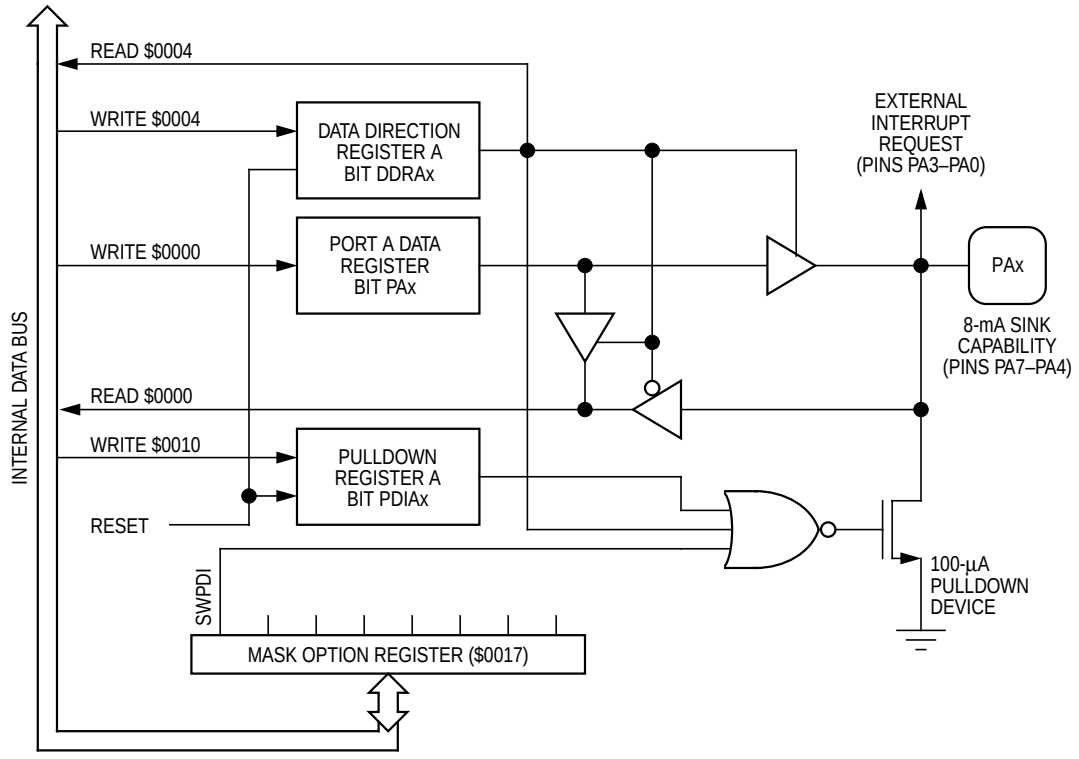


Figure 7-4. Port A I/O Circuit

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 data direction register 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–PDIA0	DDRA7–DDRA0	Pin	PA7–PA0
No	X	1	Output	U	PDIA7–PDIA0	DDRA7–DDRA0	PA7-PA0	PA7–PA0
Yes	0	0	Input, pulldown on	U	PDIA7–PDIA0	DDRA7–DDRA0	Pin	PA7–PA0
Yes	0	1	Output	U	PDIA7–PDIA0	DDRA7–DDRA0	PA7-PA0	PA7–PA0
Yes	1	0	Input, hi-z	U	PDIA7–PDIA0	DDRA7–DDRA0	Pin	PA7–PA0
Yes	1	1	Output	U	PDIA7–PDIA0	DDRA7–DDRA0	PA7-PA0	PA7–PA0

1. X = Don’t care
2. U = Undefined

7.5 Port B

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

- Programmable pulldown devices
- Oscillator output for 3-pin resistance-capacitance (RC) oscillator configuration

7.5.1 Port B Data Register

The port B data register (PORTB) 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.

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

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

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 3-pin RC oscillator configuration. Using PB1/OSC3 as an oscillator output affects port B in these ways:

a. Bit PB1 can be used as a read/write storage location without affecting the oscillator. Reset has no effect on bit PB1.

b. Bit DDRB1 in data direction register B can be used as a read/write storage location without affecting the oscillator. Reset clears DDRB1.

c. 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 0s. Writes to these bits have no effect.

7.5.2 Data Direction Register B

The contents of data direction register B (DDRB) determine whether each port B pin is an input or an output. Writing a logic 1 to a DDRB bit enables the output buffer for the associated port B pin; a logic 0 disables the output buffer. A reset initializes all DDRB bits to logic 0, configuring all port B pins as inputs. Setting a DDRB bit to a logic 1 turns off the pulldown device for that pin.

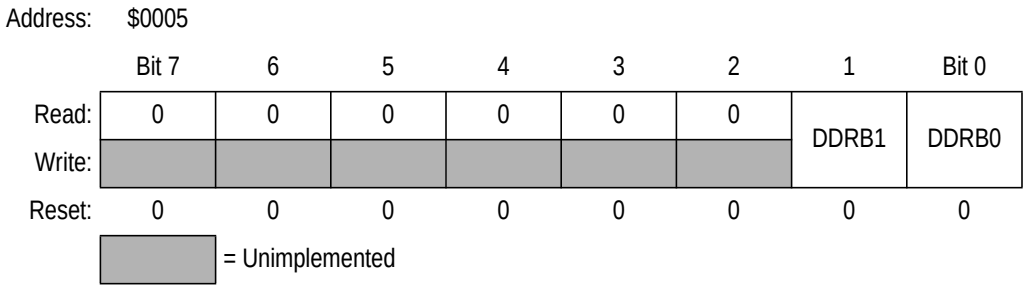


Figure 7-6. Data Direction Register B (DDRB)

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 0s. 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 0 to logic 1.*

7.5.3 Pulldown Register B

Programming the SWPDI bit in the mask option register to a logic 0 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 (PDRB). 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 0, reset initializes both port B pins as inputs with pulldown devices turned on.

Programming the SWPDI bit to a logic 1 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 1.

Address: $0011

	Bit 7	6	5	4	3	2	1	Bit 0
Read:								
Write:	0	0	0	0	0	0	PDIB1	PDIB0
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 7-7. Pulldown Register B (PDRB)

PDIB1 and PDIB0 — Port B Pulldown Inhibit Bits 1 and 0

Writing logic 0s 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 0s.

Programming the SWPDI bit in the mask option register to logic 1 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 1.

NOTE: *Avoid a floating port B input by clearing its pulldown register bit before changing its DDRB bit from logic 1 to logic 0.*

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

7.5.4 Port B Logic

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

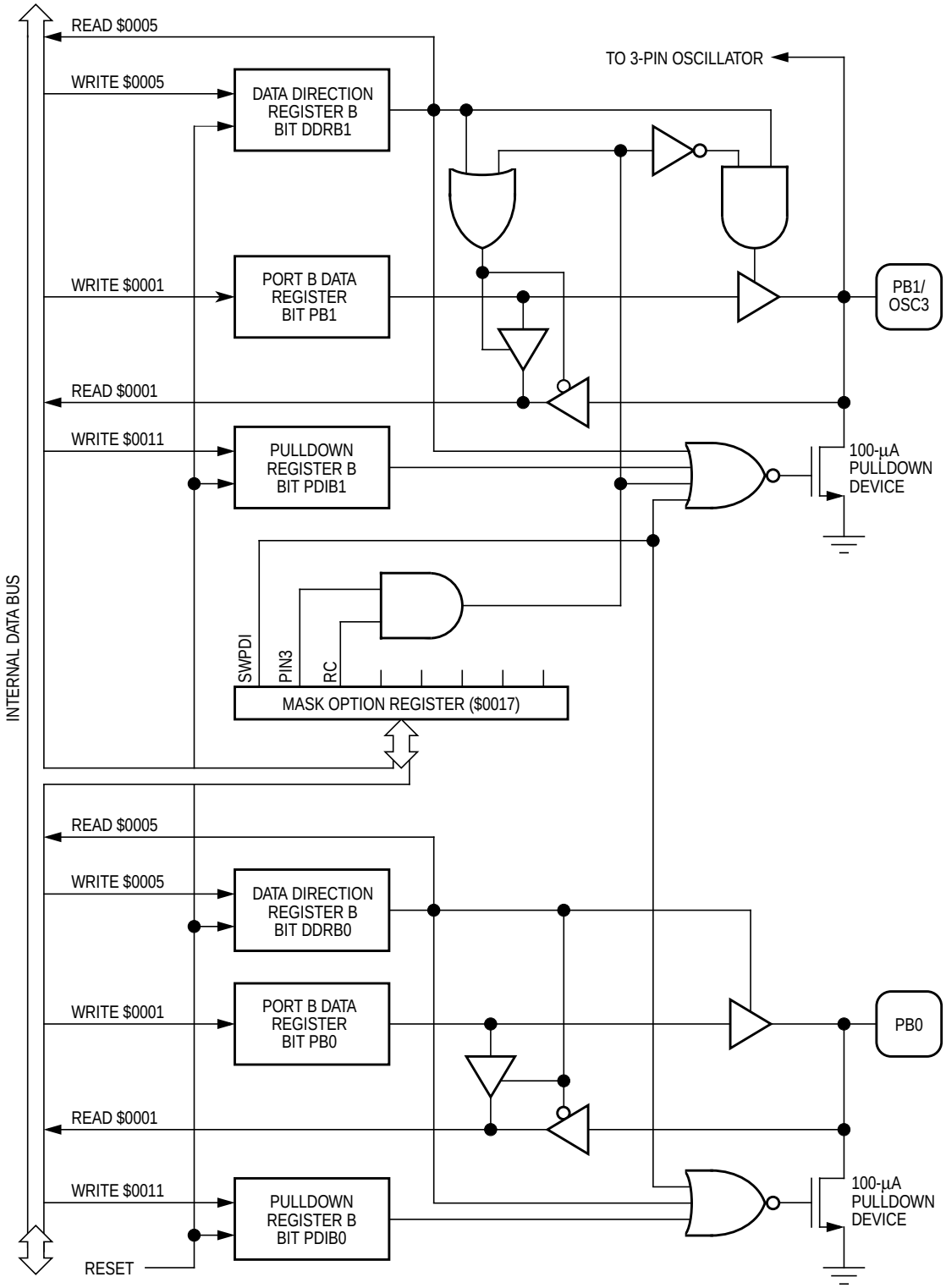


Figure 7-8. Port B I/O Circuit

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.

Programming the RC and PIN3 bits to logic 1 disables the PB1/OSC3 output buffer and pulldown device. The PB1/OSC3 bit becomes an output from the 3-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 7-2. 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

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

8.1 Contents

8.2 Introduction

This section describes the operation of the multifunction timer and the computer operating properly (COP) watchdog. **Figure 8-1** shows the organization of the timer subsystem.

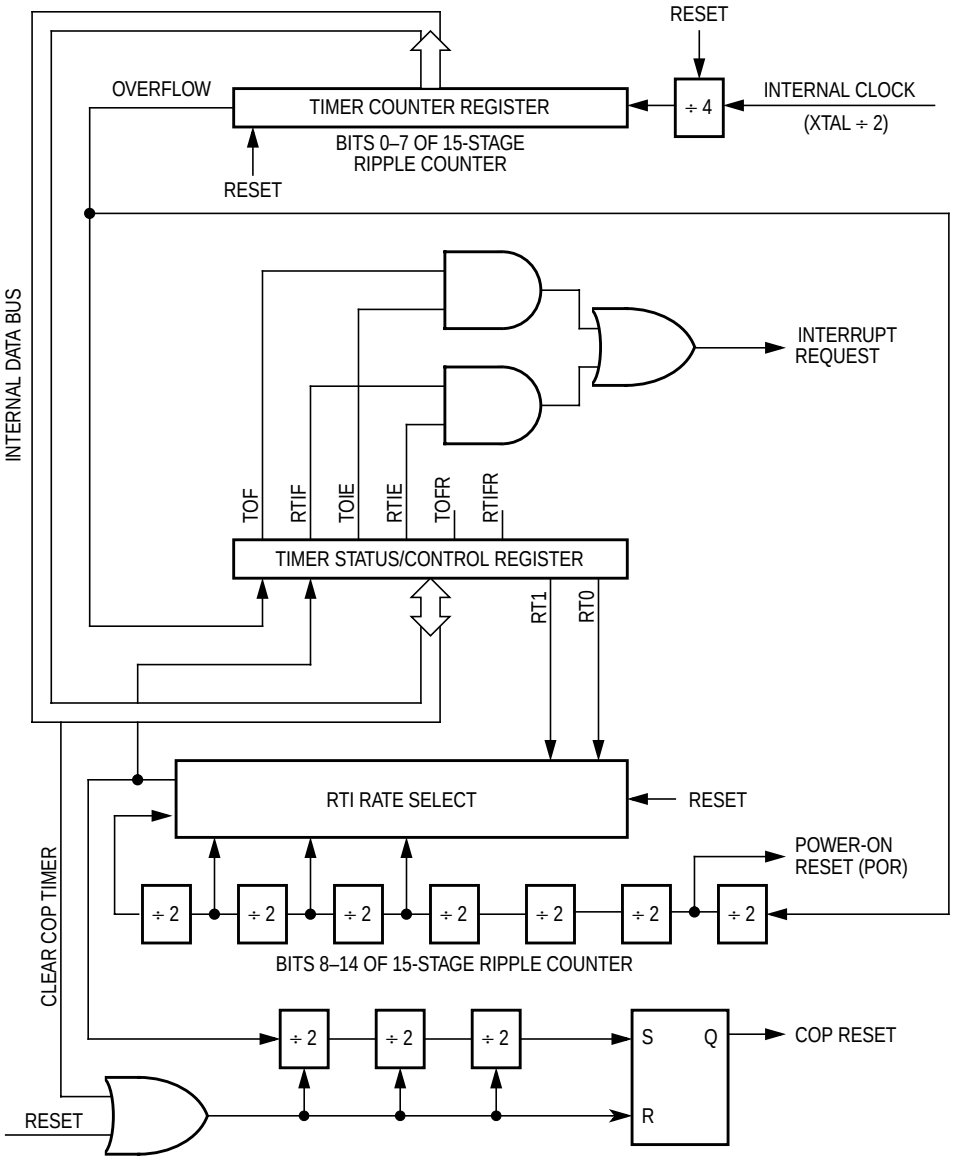


Figure 8-1. Multifunction Timer Block Diagram

8.3 Timer Status and Control Register

The read/write timer status and control register (TSCR) contains these bits:

- Timer interrupt enable bits
- Timer interrupt flags
- Timer interrupt flag reset bits
- Timer interrupt rate select bits

Address: $0008

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TOF	RTIF	TOIE	RTIE			RT1	RT0
Write:					TOFR	RTIFR		
Reset:	0	0	0	0	U	U	1	1

= Unimplemented U = Unaffected

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 1 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 real-time interrupt (RTI) output becomes active. RTIF generates a real-time interrupt request if RTIE is also set. Clear RTIF by writing a logic 1 to the RTIFR bit. Writing to RTIF has no effect. Reset clears RTIF.

TOIE — Timer Overflow Interrupt Enable Bit

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 Bit

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 1 to this write-only bit clears the TOF bit. TOFR always reads as a logic 0. Reset does not affect TOFR.

RTIFR — Real-Time Interrupt Flag Reset

Writing a logic 1 to this write-only bit clears the RTIF bit. RTIFR always reads as a logic 0. Reset does not affect RTIFR.

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

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

NOTE: *Changing RT1 and RT0 when a COP timeout is imminent or uncertain may cause an RTI request to be missed or an additional RTI request to be generated. Clear the COP timer just before changing RT1 and RT0.*

Table 8-1. Real-Time Interrupt Rate Selection

RT1:RT0	Number of Cycles to RTI	RTI Period(1)	Number of Cycles to COP Reset	COP Timeout Period(1)
0 0	2^{14} = 16,384	8.2 ms	2^{17} = 131,072	65.5 ms
0 1	2^{15} = 32,768	16.4 ms	2^{18} = 262,144	131.1 ms
1 0	2^{16} = 65,536	32.8 ms	2^{19} = 524,288	262.1 ms
1 1	2^{17} = 131,072	65.5 ms	2^{20} = 1,048,576	524.3 ms

1. At 2-MHz bus, 4-MHz XTAL, 0.5 µs per cycle

8.4 Timer Counter Register

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).

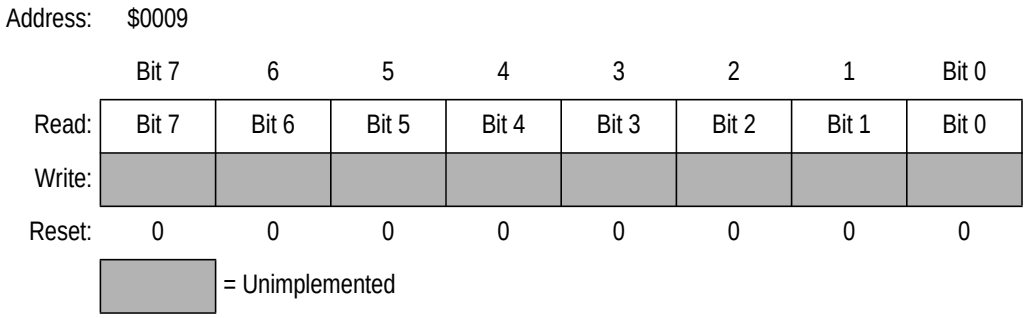


Figure 8-3. Timer Counter Register (TCNTR)

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.

8.5 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 0 to bit 0 of the COP register clears the COP watchdog and prevents a COP reset.

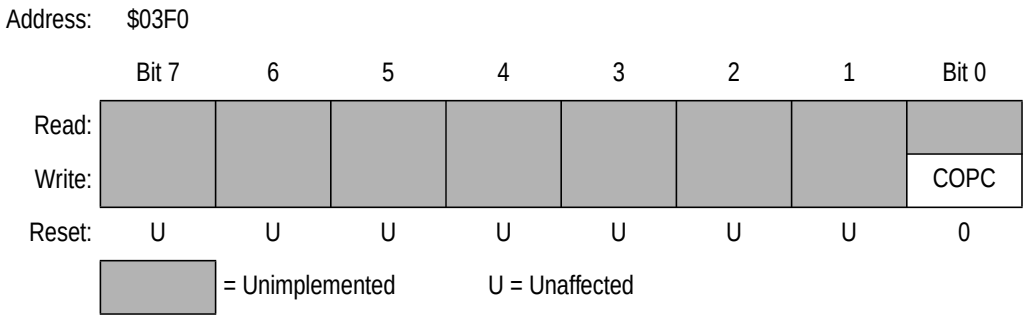


Figure 8-4. COP Register (COPR)

COPC — COP Clear Bit

This write-only bit resets the COP watchdog. Reading address $03F0 returns the read-only memory (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 1 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 1 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

9.1 Contents

9.2 Introduction

This section describes how to program the 504-byte erasable, programmable read-only memory (EPROM)/one-time programmable read-only memory (OTPROM).

NOTE: *In packages with no quartz window, the 504 bytes of EPROM function as an OTPROM.*

9.3 EPROM Programming Register

The EPROM programming register (EPROG) 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 0s.

Address: $0018

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	R	R	R	R	R	ELAT	MPGM	EPGM
Reset:	U	U	U	U	U	0	0	0

R = Reserved U = Unaffected

Figure 9-1. EPROM Programming Register (EPROG)

ELAT — EPROM Bus Latch Bit

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 Bit

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 Bit

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 1s 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 0s.

9.4 EPROM/OTPROM Programming

The MC68HC705K1 does not contain built-in bootloader ROM code. To program this device, use an external programming system such as the M68HC705KICS evaluation module (EVM) or an M68HC705K1GANG programmer.

Factory-provided software for programming the EPROM/OTPROM is available through the Freescale web site at:
htt://freescale.com

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.

This 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.*

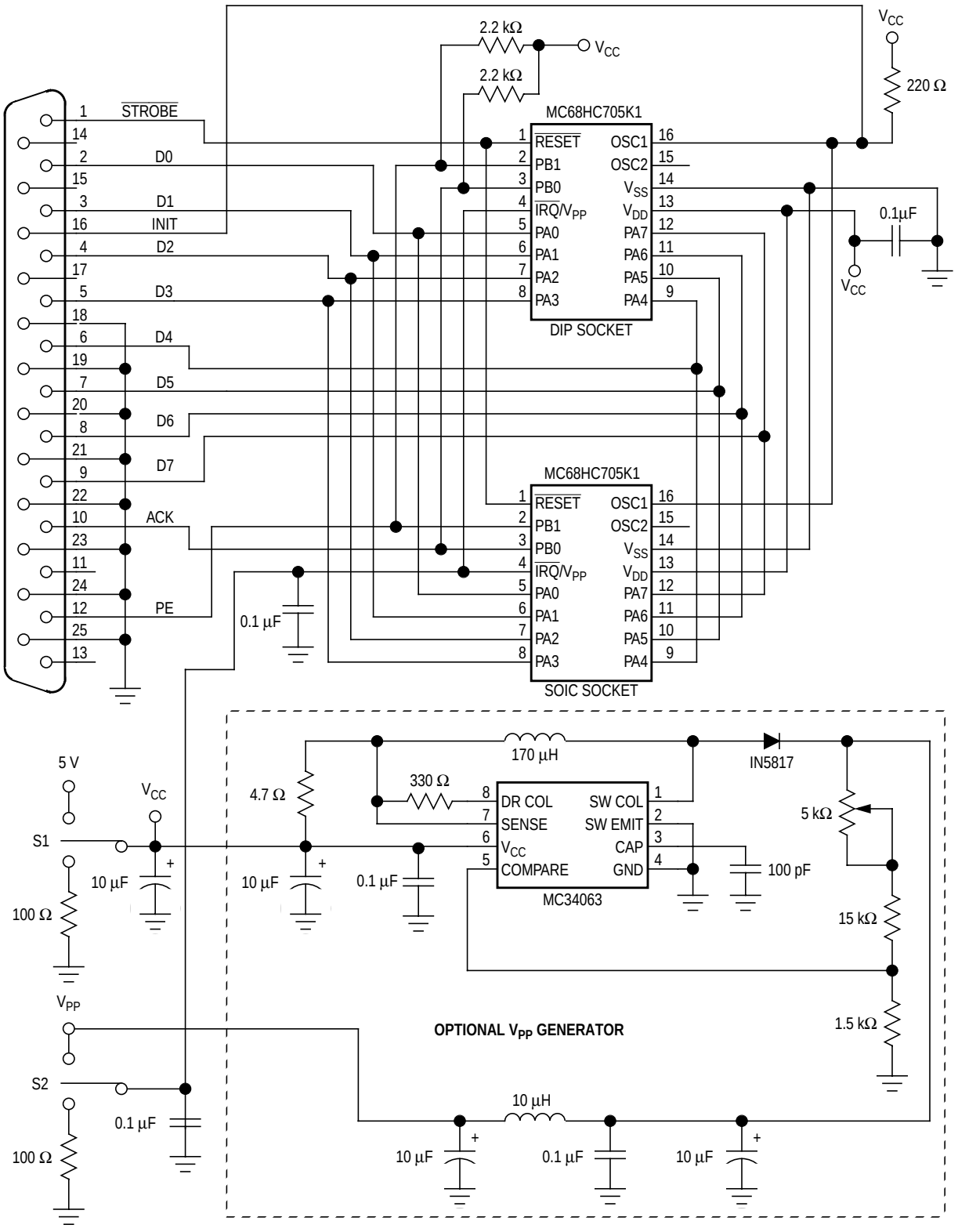


Figure 9-2. Programming Circuit

9.5 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 wave length 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 0.

9.6 Mask Option Register

The mask option register (MOR) is an EPROM/OTPROM byte that controls these options:

- Port A and port B programmable pulldown devices (enable or disable)
- Oscillator connections (2-pin or 3-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.

Address: $0017

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	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 Bit

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

1 = Software pulldown inhibited
0 = Software pulldown enabled

PIN3 — 3-Pin RC Oscillator Bit

This EPROM bit configures the on-chip oscillator as either a 3-pin oscillator or as a 2-pin oscillator. The PIN3 bit should be cleared when the RC bit is clear.

1 = 3-pin oscillator configured
0 = 2-pin oscillator configured

RC — RC Oscillator Bit

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 Bit

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 Bit

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

1 = LVR circuit enabled
0 = LVR circuit disabled

PIRQ — Port A IRQ Enable Bit

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 Bit

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 Bit

This EPROM bit enables the COP watchdog.

1 = COP watchdog enabled
0 = COP watchdog disabled

NOTE: *In 3-pin RC oscillator configurations, the personality EPROM (PEPROM) cannot be programmed by user software. If the voltage on $\overline{IRQ}/V_{PP}$ is raised above V_{DD}, the oscillator will revert to a 2-pin oscillator configuration and device operation will be disrupted.*

EPROM/OTPROM

Section 10. Personality EPROM (PEPROM)

10.1 Contents

10.2 Introduction

This section describes how to program the 64-bit personality erasable, programmable read-only memory (PEPROM). **Figure 10-1** shows the structure of the PEPROM subsystem.

Personality EPROM (PEPROM)

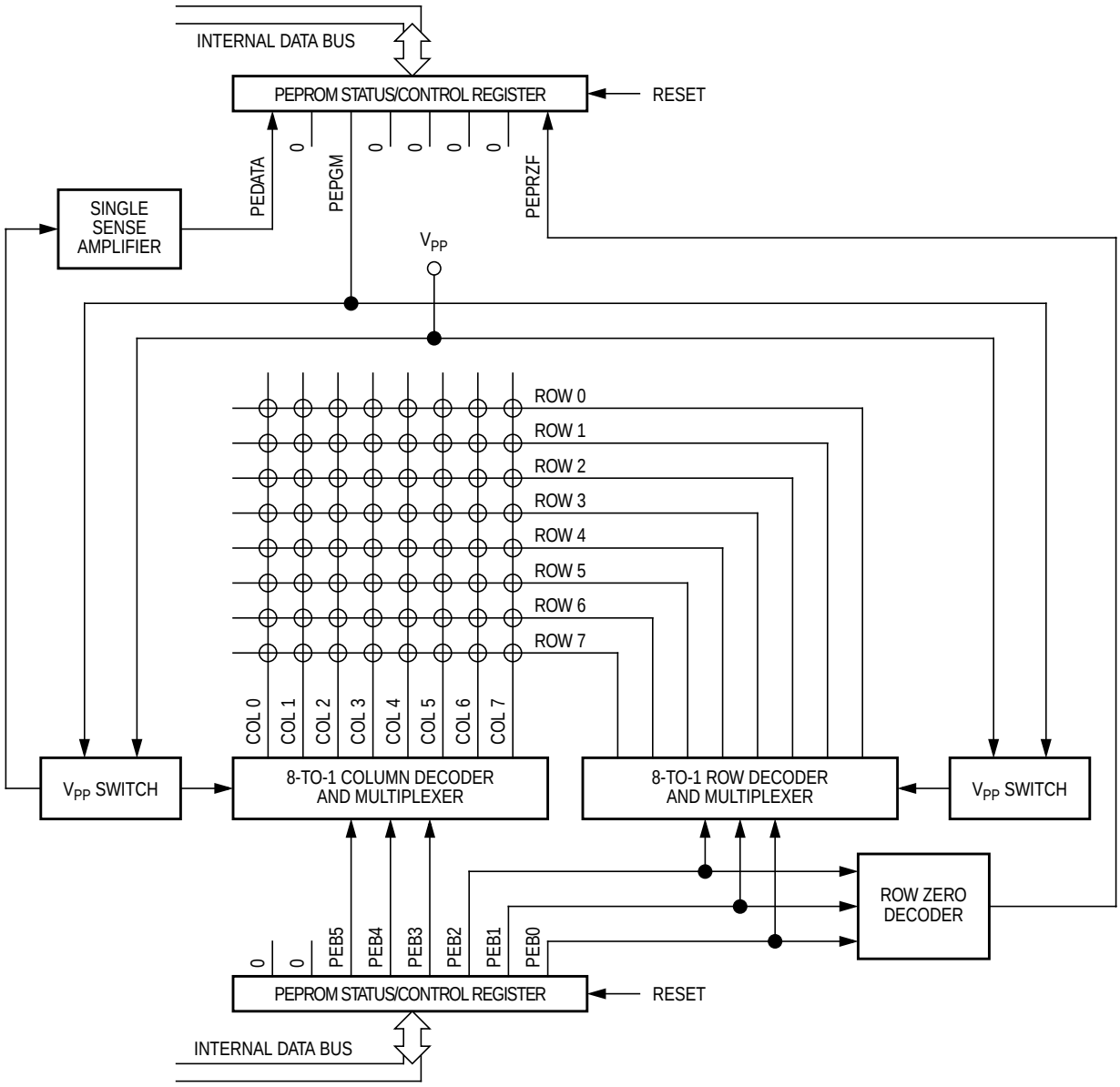


Figure 10-1. Personality EPROM

10.3 PEPROM Registers

Two input/output (I/O) registers control programming and reading of the PEPROM:

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

10.3.1 PEPROM Bit Select Register

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

Address: $000E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	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–PEB0 select the PEPROM row, and bits PEB5–PEB3 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

10.3.2 PEPROM Status and Control Register

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

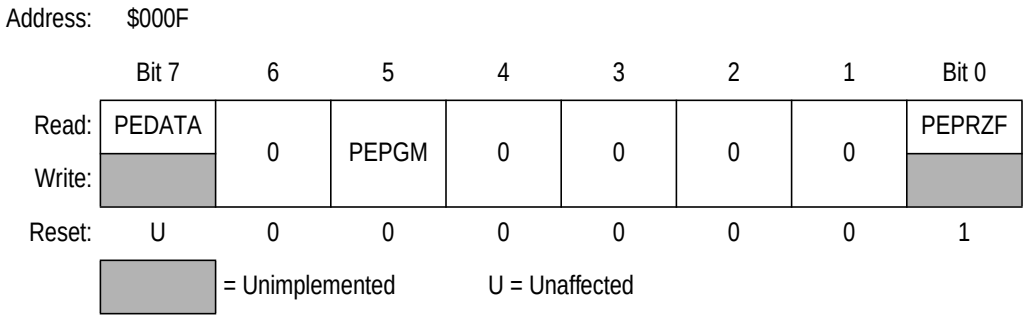


Figure 10-3. PEPROM Status and Control Register (PESCR)

PEDATA — PEPROM Data Bit

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 1
0 = PEPROM data logic 0

PEPGM — PEPROM Program Control Bit

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

10.4 PEPROM Programming

Factory-provided software for programming the PEPROM is available through the Freescale web site at:

htt://freescale.com

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. This 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).*

In 3-pin RC oscillator configurations, the PEPROM cannot be programmed by user software. If the voltage on $\overline{IRQ}/V_{PP}$ is raised above V_{DD}, the oscillator will revert to a 2-pin oscillator configuration and device operation will be disrupted.

10.5 PEPROM Reading

This 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.

10.6 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 wave length 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 0.

Personality EPROM (PEPROM)

Section 11. Instruction Set

11.1 Contents

11.2 Introduction

The microcontroller unit (MCU) instruction set has 62 instructions and uses eight addressing modes. The instructions include all those of the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. The MUL instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is stored in the index register, and the low-order product is stored in the accumulator.

11.3 Addressing Modes

The central processor unit (CPU) uses eight addressing modes for flexibility in accessing data. The addressing modes provide eight different ways for the CPU to find the data required to execute an instruction.

The eight addressing modes are:

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

11.3.1 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 operand address and are one byte long.

11.3.2 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 operand address and are two bytes long. The opcode is the first byte, and the immediate data value is the second byte.

11.3.3 Direct

Direct instructions can access any of the first 256 memory locations 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.

11.3.4 Extended

Extended instructions use three bytes and can 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 Freescale 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.

11.3.5 Indexed, No Offset

Indexed instructions with no offset are 1-byte instructions that can access data with variable addresses within the first 256 memory locations. The index register contains the low byte of the effective address of the operand. The CPU automatically uses $00 as the high byte, so these instructions can address locations $0000–$00FF.

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

11.3.6 Indexed, 8-Bit Offset

Indexed, 8-bit offset instructions are 2-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 effective 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.

11.3.7 Indexed, 16-Bit Offset

Indexed, 16-bit offset instructions are 3-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 effective 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.

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 Freescale assembler determines the shortest form of indexed addressing.

11.3.8 Relative

Relative addressing is only for branch instructions. If the branch condition is true, the CPU finds the effective 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 Freescale 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.

11.4 Instruction Types

The MCU instructions fall into five categories:

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

11.4.1 Register/Memory Instructions

These instructions operate on CPU registers and memory locations. Most of them 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. 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

11.4.2 Read-Modify-Write Instructions

These instructions read a memory location or a register, modify its contents, and write the modified value back to the memory location or to the register.

NOTE: *Do not use read-modify-write operations on write-only registers.*

Table 11-2. Read-Modify-Write Instructions

Instruction	Mnemonic
Arithmetic shift left (same as LSL)	ASL
Arithmetic shift right	ASR
Bit clear	BCLR(1)
Bit set	BSET(1)
Clear register	CLR
Complement (one's complement)	COM
Decrement	DEC
Increment	INC
Logical shift left (same as ASL)	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(2)

1. Unlike other read-modify-write instructions, BCLR and BSET use only direct addressing.
2. TST is an exception to the read-modify-write sequence because it does not write a replacement value.

11.4.3 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.

The BRCLR and BRSET instructions cause a branch based on the state of any readable bit in the first 256 memory locations. These 3-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 effective 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. 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

11.4.4 Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory, which includes I/O registers and on-chip RAM locations. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations.

Table 11-4. Bit Manipulation Instructions

Instruction	Mnemonic
Bit clear	BCLR
Branch if bit clear	BRCLR
Branch if bit set	BRSET
Bit set	BSET

11.4.5 Control Instructions

These instructions act on CPU registers and control CPU operation during program execution.

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

11.5 Instruction Set Summary

Table 11-6. Instruction Set Summary (Sheet 1 of 6)

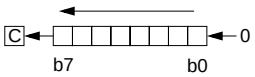
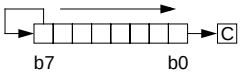
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	↕	—	↕	↕	↕	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr*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

Table 11-6. Instruction Set Summary (Sheet 2 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BIH *rel*	Branch if IRQ Pin High	PC ← (PC) + 2 + *rel* ? IRQ = 1	—	—	—	—	—	REL	2F	rr	3
BIL *rel*	Branch if IRQ Pin Low	PC ← (PC) + 2 + *rel* ? 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	2 3 4 5 4 3
BLO *rel*	Branch if Lower (Same as BCS)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	PC ← (PC) + 2 + *rel* ? C ∨ Z = 1	—	—	—	—	—	REL	23	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	PC ← (PC) + 2 + *rel* ? I = 0	—	—	—	—	—	REL	2C	rr	3
BMI *rel*	Branch if Minus	PC ← (PC) + 2 + *rel* ? N = 1	—	—	—	—	—	REL	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	PC ← (PC) + 2 + *rel* ? I = 1	—	—	—	—	—	REL	2D	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + 2 + *rel* ? Z = 0	—	—	—	—	—	REL	26	rr	3
BPL *rel*	Branch if Plus	PC ← (PC) + 2 + *rel* ? N = 0	—	—	—	—	—	REL	2A	rr	3
BRA *rel*	Branch Always	PC ← (PC) + 2 + *rel* ? 1 = 1	—	—	—	—	—	REL	20	rr	3
BRCLR *n opr rel*	Branch if Bit n Clear	PC ← (PC) + 2 + *rel* ? Mn = 0	—	—	—	—	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN *rel*	Branch Never	PC ← (PC) + 2 + *rel* ? 1 = 0	—	—	—	—	—	REL	21	rr	3
BRSET *n opr rel*	Branch if Bit n Set	PC ← (PC) + 2 + *rel* ? Mn = 1	—	—	—	—	↕	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
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

Table 11-6. Instruction Set Summary (Sheet 3 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
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}$) = $FF – (A) X ← ($\overline{X}$) = $FF – (X) M ← ($\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)	—	—	↕	↕	↕	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
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

Table 11-6. Instruction Set Summary (Sheet 4 of 6)

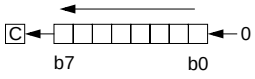
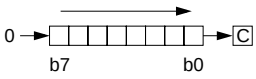
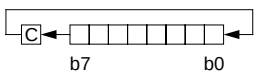
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Effective Address	—	—	—	—	—	DIR EXT IX2 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		1 1
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	dd ff 	5 3 3 6 5
NOP	No Operation		—	—	—	—	—	INH	9D		2
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X	Logical OR Accumulator with Memory	A ← (A) ∨ (M)	—	—	↕	↕	—	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff	2 3 4 5 4 3
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X	Rotate Byte Left through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	39 49 59 69 79	dd ff 	5 3 3 6 5

Table 11-6. Instruction Set Summary (Sheet 5 of 6)

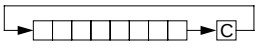
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ROR *opr* RORA RORX ROR *opr*,X ROR ,X	Rotate Byte Right through Carry Bit		—	—	↕	↕	↕	DIR INH INH IX1 IX	36 46 56 66 76	dd ff	5 3 3 6 5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2
RTI	Return 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		9
RTS	Return from Subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	—	—	—	—	—	INH	81		6
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 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
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		1 0
TAX	Transfer Accumulator to Index Register	X ← (A)	—	—	—	—	—	INH	97		2

Table 11-6. Instruction Set Summary (Sheet 6 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
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		—	0	—	—	—	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

11.6 Opcode Map

See **Table 11-7**.

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

INH = Inherent
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	5 BRSET0 3 DIR

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

Section 12. Electrical Specifications

12.1 Contents

12.2 Introduction

This section contains electrical and timing specifications.

12.3 Maximum Ratings

Maximum ratings are the extreme limits to which the MCU can be exposed without permanently damaging it.

The MCU contains circuitry to protect the inputs against damage from high static voltages; however, do not apply voltages higher than those shown in the table here. Keep V_{In} and V_{Out} within the range $V_{SS} \le (V_{In}$ or $V_{Out}) \le V_{DD}$. Connect unused inputs to the appropriate voltage level, either V_{SS} or V_{DD}.

Rating[1]	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}	V_{DD} –0.3 to 16.0	
Current drain per pin excluding V_{DD} and V_{SS}	I	25	mA
Storage temperature range	T_{STG}	–65 to +150	°C

1. Voltages referenced to V_{SS}

NOTE: *This device is not guaranteed to operate properly at the maximum ratings. Refer to* ***12.8 5.0-Volt DC Electrical Characteristics*** *and* ***12.9 3.3-Volt DC Electrical Characteristics*** *for guaranteed operating conditions.*

12.4 Operating Temperature Range

Rating[1]	Symbol	Value	Unit
Operating temperature range[2] MC68HC705K1P, DW, S MC68HC705K1CP, CDW, CS	T_A	T_L to T_H –40 to +85	°C

1. Voltages referenced to V_{SS}
2. P = Plastic dual in-line package (PDIP)
   DW = Small outline integrated circuit (SOIC)
   S = Ceramic dual in-line package (cerdip)
   C = Extended temperature range (–40°C to +85°C)

12.5 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Maximum junction temperature	T_J	150	°C
Thermal resistance MC68HC705K1P(1) MC68HC705K1DW(2)	θ_{JA}	 100 140	°C/W

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

12.6 Power Considerations

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

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

Where:

T_A = ambient temperature in °C
θ_{JA} = package thermal resistance, junction to ambient in °C/W
$P_D = P_{INT} + P_{I/O}$
$P_{INT} = I_{CC} \times V_{CC}$ = chip internal power dissipation
$P_{I/O}$ = power dissipation on input and output pins (user-determined)

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

Ignoring $P_{I/O}$, the relationship between P_D and T_J is approximately:

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

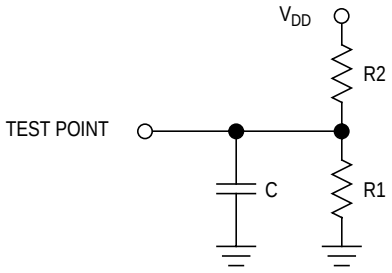
Solving equations (1) and (2) for K gives:

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

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.

12.7 Equivalent Pin Loading

Figure 12-1 shows the equivalent input/output (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

12.8 5.0-Volt DC Electrical Characteristics

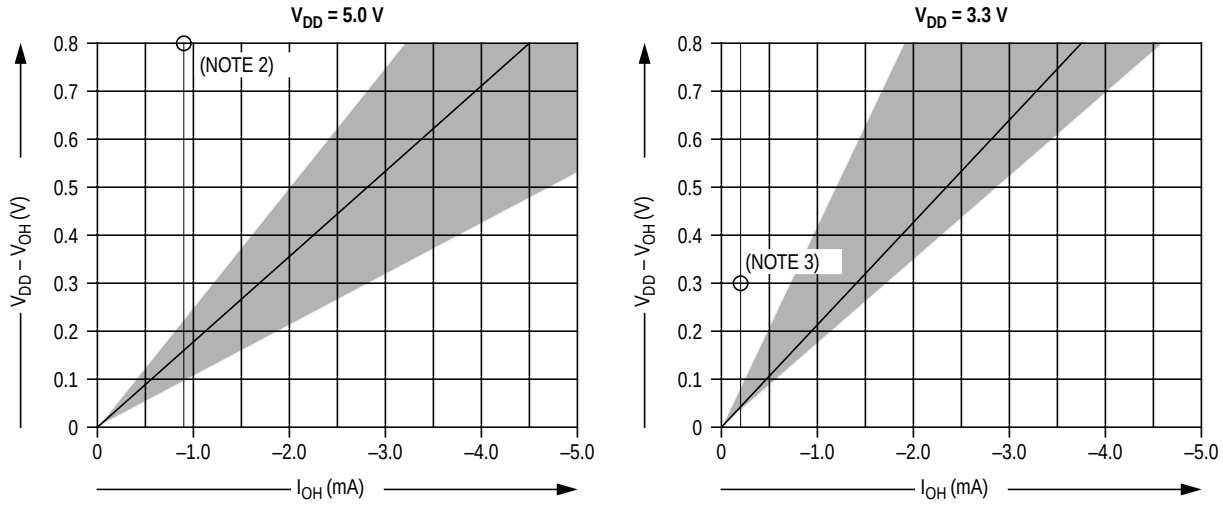
Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output voltage I_{Load} = 10.0 µA I_{Load} = –10.0 µA	 V_{OL} V_{OH}	 — V_{DD} – 0.1	 — —	 0.1 —	 V
Output high voltage, I_{Load} = –0.8 mA PA7–PA0, PB1/OSC3, PB0	V_{OH}	V_{DD} – 0.8	—	—	V
Output low voltage I_{Load} = 1.6 mA, PA3–PA0, PB1/OSC3, PB0 I_{Load} = 8.0 mA, PC7–PC4	V_{OL}	 — —	 — —	 0.4 0.4	 V
Input high voltage PA7–PA0, PB1/OSC3, PB0, $\overline{IRQ}/V_{PP}$, $\overline{RESET}$, OSC1	V_{IH}	0.7 x 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 x 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 devices off) $\overline{RESET}$ (pulldown devices 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
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 V ±10%; V_{SS} = 0 Vdc, T_A = T_L to T_H, unless otherwise noted
2. Typical 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). 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). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. 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.

12.9 3.3-Volt DC Electrical Characteristics

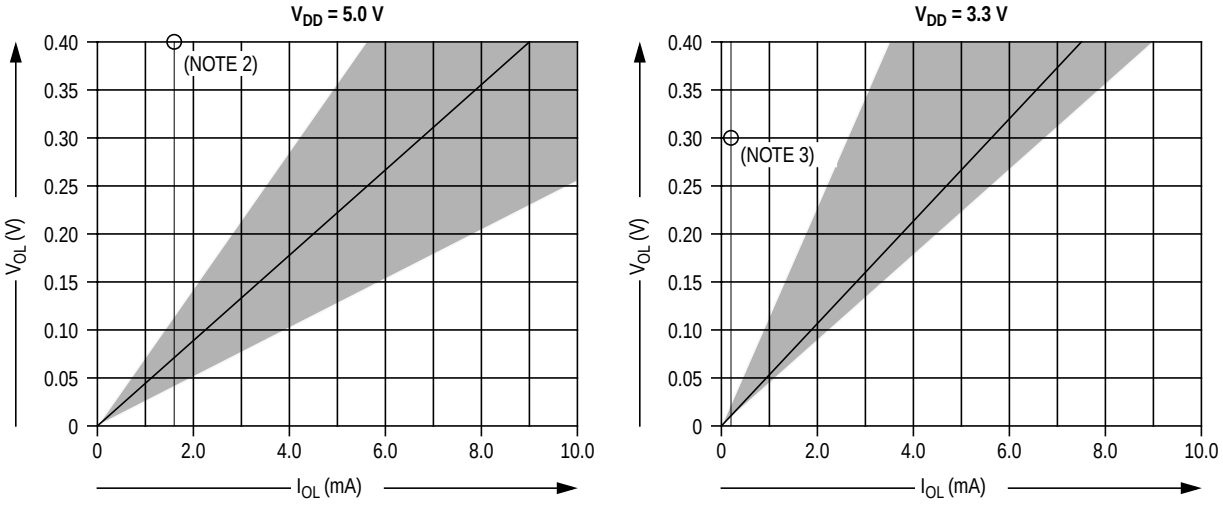
Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output voltage I_{Load} = 10.0 µA I_{Load} = –10.0 µA	 V_{OL} V_{OH}	 — V_{DD} – 0.1	 — —	 0.1 —	 V
Output high voltage, I_{Load} = –0.4 mA PA7–PA0, PB1/OSC3, PB0	V_{OH}	V_{DD} – 0.3	—	—	V
Output low voltage I_{Load} = 0.4 mA, PA3–PA0, PB1/OSC3, PB0 I_{Load} = 3.0 mA, PC7–PC4	V_{OL}	 — —	 — —	 0.3 0.3	 V
Input high voltage PA7–PA0, PB1/OSC3, PB0, $\overline{IRQ}$/V_{PP}, $\overline{RESET}$, OSC1	V_{IH}	0.7 x 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 x 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
Oscillator internal resistor (OSC1 to OSC2)	R_{OSC}	1.0	2.0	3.0	MΩ

1. V_{DD} = 3.3 V ±10%; V_{SS} = 0 Vdc, T_A = T_L to T_H, unless otherwise noted
2. Typical 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). 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). All inputs 0.2 V from rail. No dc loads. Less than 50 pF on all outputs. C_L = 20 pF on OSC2. 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.



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 versus 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 versus 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

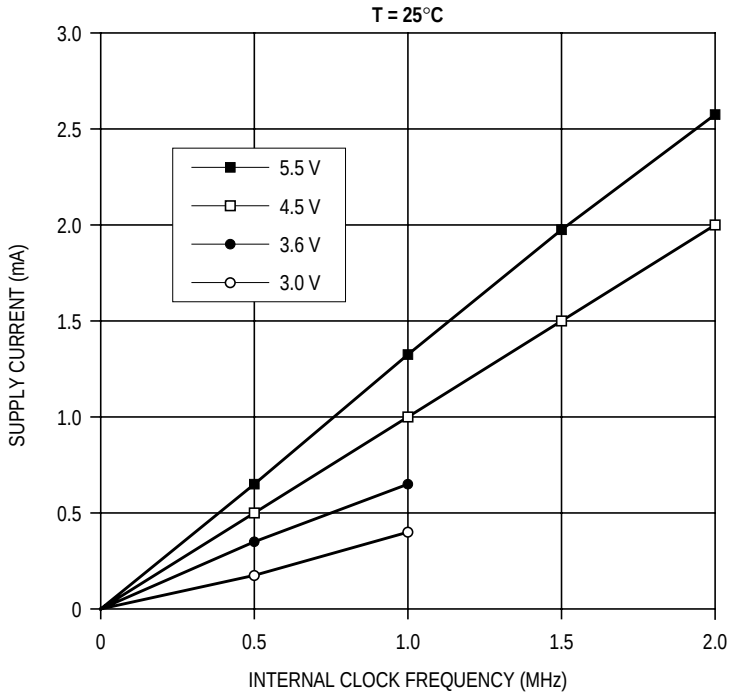


Figure 12-4. Run I_{DD} versus Internal Clock Frequency

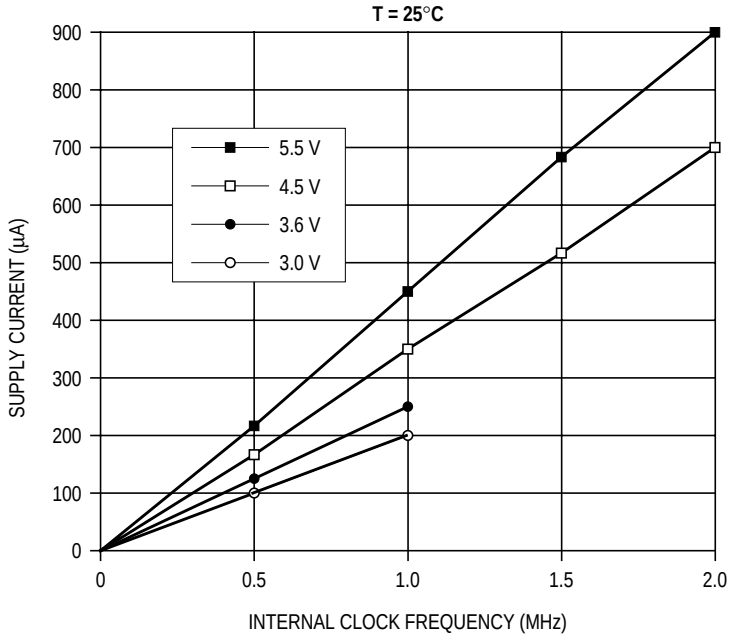


Figure 12-5. Wait I_{DD} versus Internal Clock Frequency

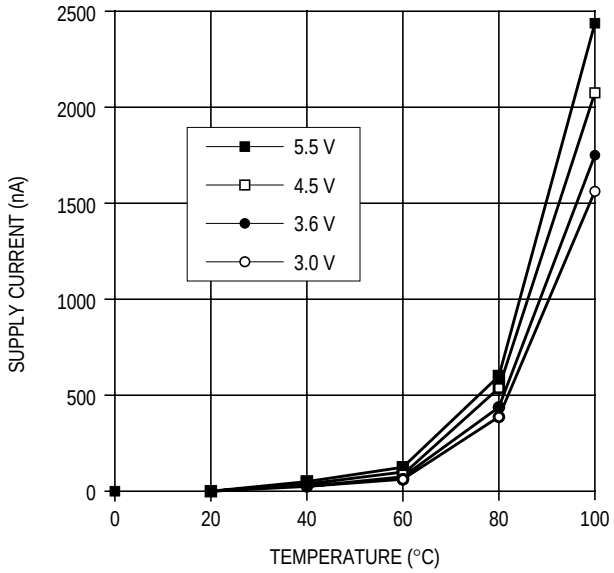


Figure 12-6. Stop I_{DD} versus Temperature

12.10 5.0-Volt Control Timing

Characteristic[1]	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 startup time	t_{OXON}	—	100	ms
Stop recovery startup 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.

12.11 3.3-Volt Control Timing

Characteristic[1]	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 startup time	t_{OXON}	—	100	ms
Stop recovery startup 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}.

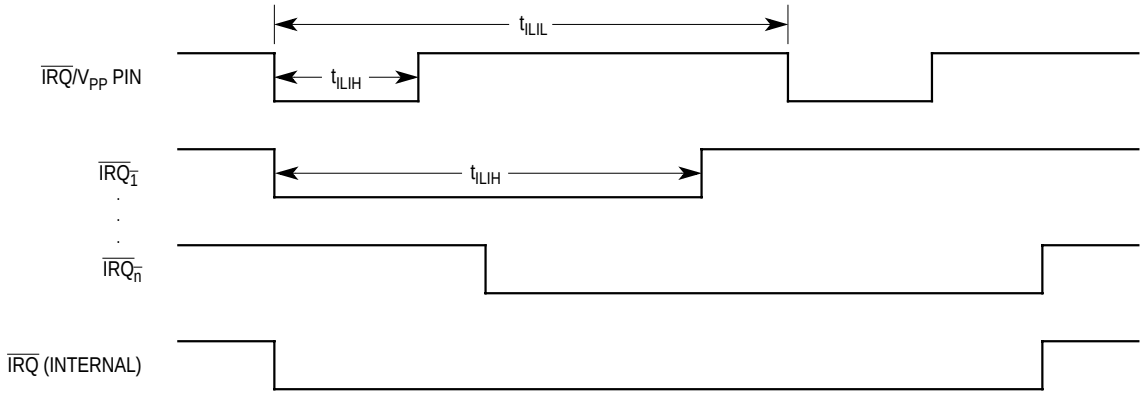
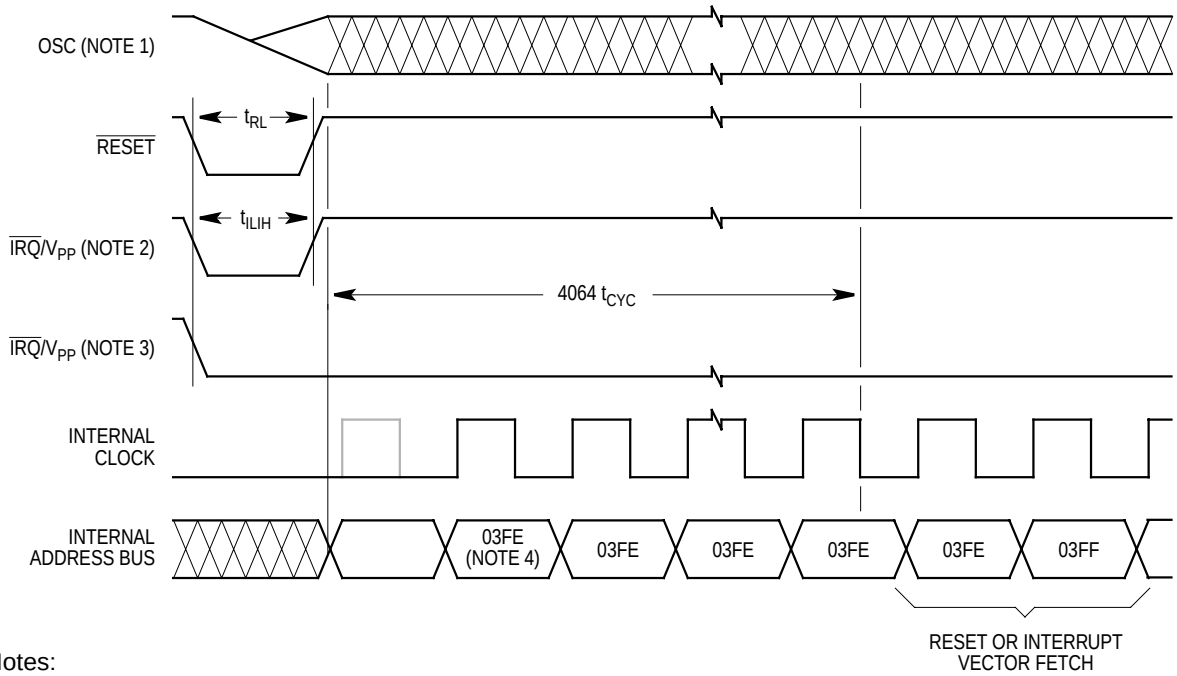
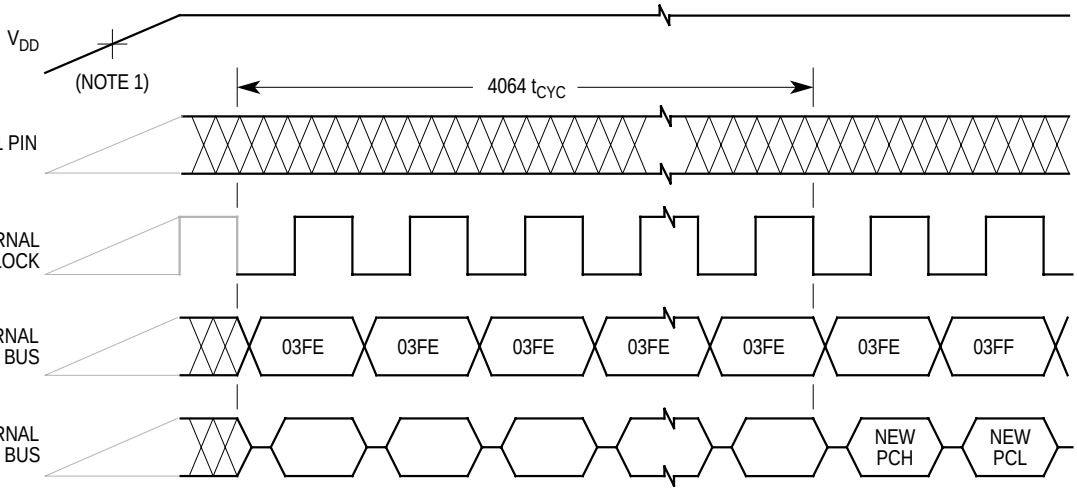


Figure 12-7. External Interrupt Timing



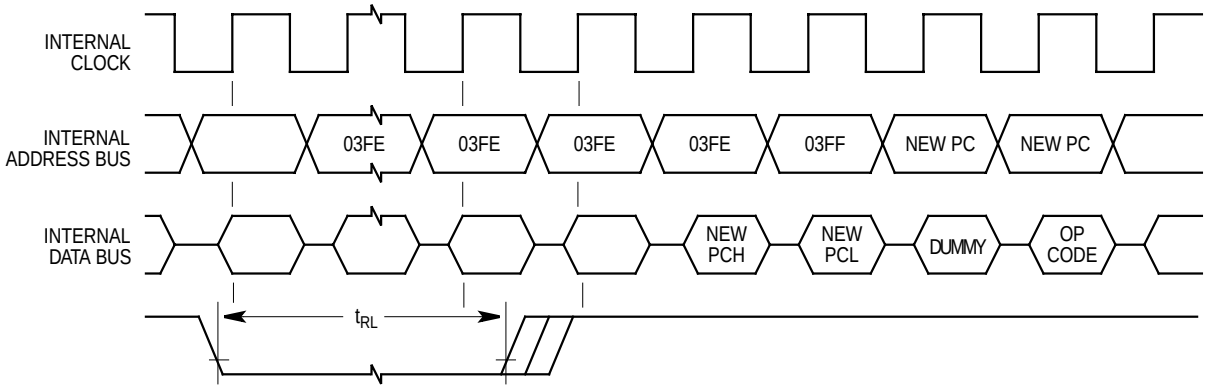
Notes:
1. Internal clocking from OSC1 pin.
2. Edge-triggered external interrupt mask option.
3. Edge- and level-triggered external interrupt mask option.
4. Reset vector shown as example.

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

12.12 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.

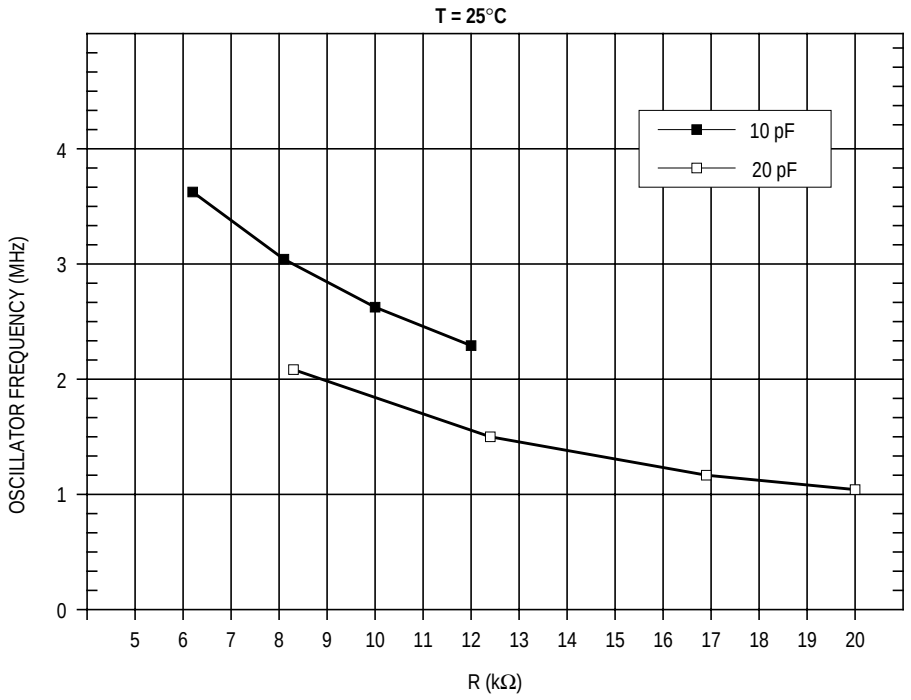


Figure 12-11. 2-Pin RC Oscillator R versus Frequency (V_{DD} = 5.0 V)

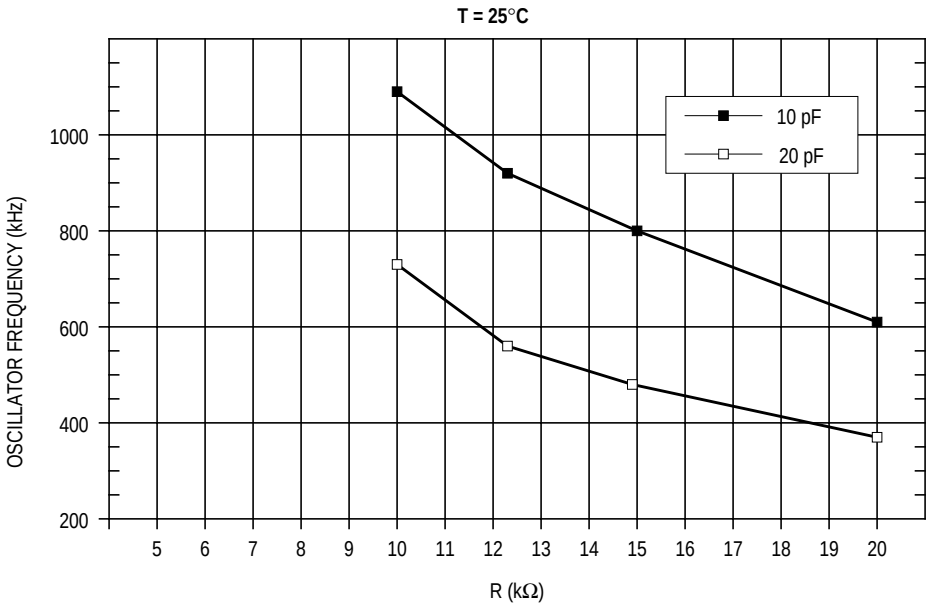


Figure 12-12. 3-Pin RC Oscillator R versus Frequency (V_{DD} = 5.0 V)

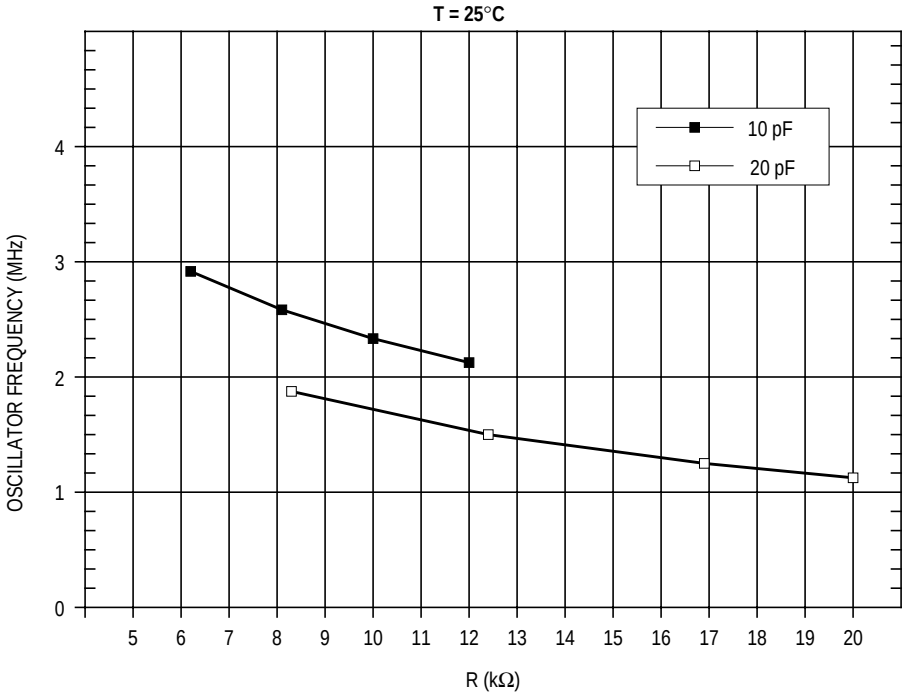


Figure 12-13. 2-Pin Oscillator R versus Frequency (V_{DD} = 3.0 V)

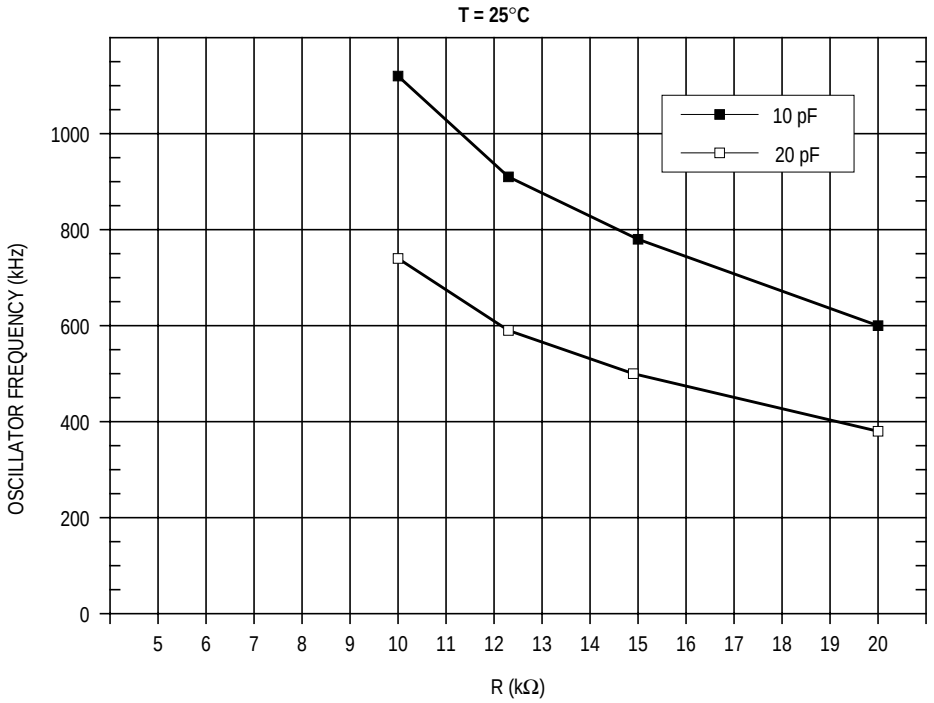


Figure 12-14. 3-Pin Oscillator R versus Frequency (V_{DD} = 3.0 V)

Section 13. Mechanical Specifications

13.1 Contents

13.2 Introduction

Package dimensions available at the time of this publication for the MC68HC705K1 are provided in this section. The packages are:

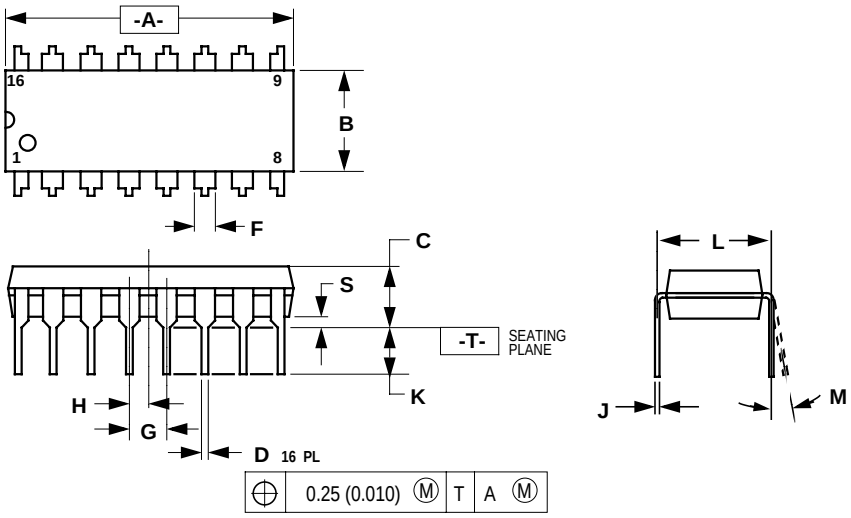
- 16-pin plastic dual in-line package (PDIP)
- 16-pin small outline integrated circuit package (SOIC)
- 16-pin ceramic DIP (cerdip)

To make sure that you have the latest case outline specifications, contact one of the following:

- Local Freescale Sales Office
- Worldwide Web (wwweb) at http://www.freescale.com

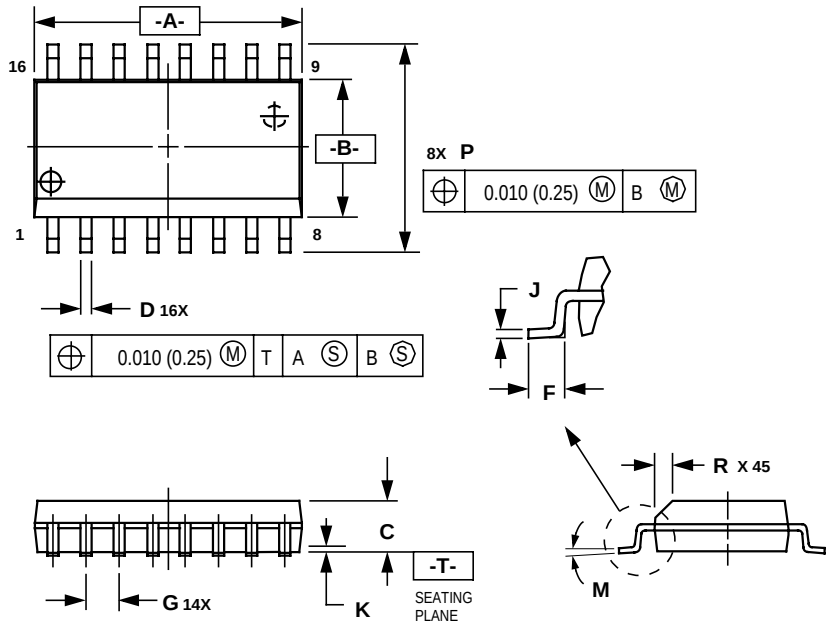
Follow wweb on-line instructions to retrieve the current mechanical specifications.

13.3 Plastic Dual In-Line Package (Case 648)



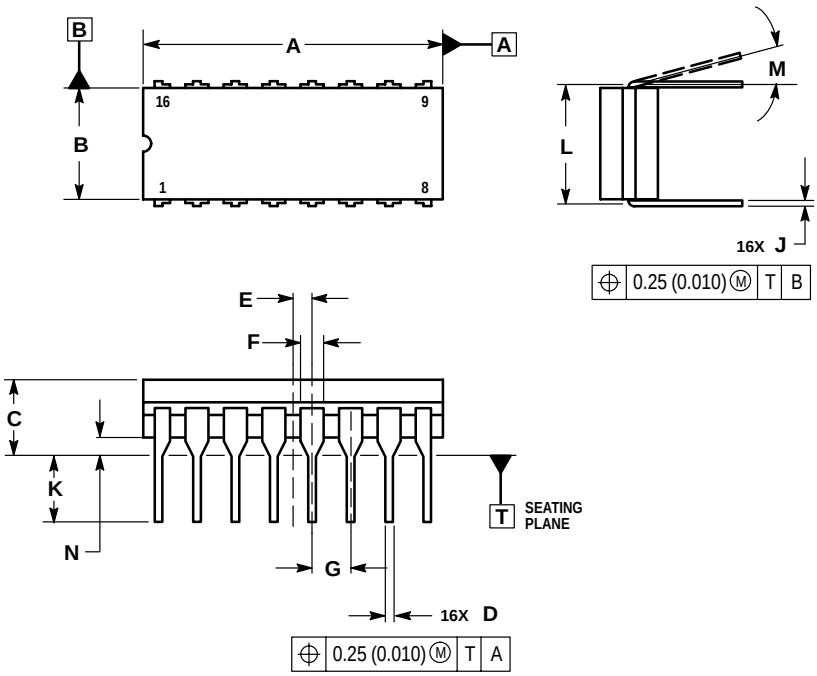
DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.740	0.770	18.80	19.55
B	0.250	0.270	6.35	6.85
C	0.145	0.175	3.69	4.44
D	0.015	0.021	0.39	0.53
F	0.040	0.70	1.02	1.77
G	0.100 BSC		2.54 BSC	
H	0.050 BSC		1.27 BSC	
J	0.008	0.015	0.21	0.38
K	0.110	0.130	2.80	3.30
L	0.295	0.305	7.50	7.74
M	0°	10°	0°	10°
S	0.020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

13.4 Small Outline Integrated Circuit (Case 751)



DIM	MILLIMETERS		INCHES	
	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

13.5 Ceramic Dual In-Line Package (Case 620)



NOTES:
1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M, 1994.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.
4. DIMENSION 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	0.785	19.05	19.93
B	0.240	0.295	6.10	7.49
C	—	0.200	—	5.08
D	0.015	0.020	0.39	0.50
E	0.050 BSC		1.27 BSC	
F	0.055	0.065	1.40	1.65
G	0.100 BSC		2.54 BSC	
H	0.008	0.015	0.21	0.38
K	0.125	0.170	3.18	4.31
L	0.300 BSC		7.62 BSC	
M	0 °	15 °	0 °	15 °
N	0.020	0.040	0.51	1.01

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

Mechanical Specifications

Section 13. Ordering Information

13.1 Contents

13.2 Introduction

This section contains ordering information for the available package types.

13.3 MCU Order Numbers

Table 13-1 lists the MC order numbers.

Table 13-1. MC68HC705K1 Order Numbers

Package Type	Temperature Range	Order Number
16-pin plastic dual in-line package (PDIP)	0°C to +70°C	MC68HC705K1P(1)
16-pin small outline integrated circuit (SOIC)	0°C to +70°C	MC68HC705K1DW(2)
16-pin ceramic dual in-line package (cerdip)	0°C to +70°C	MC68HC705K1S(3)
16-pin plastic dual in-line package (PDIP)	–40°C to +85°C	MC68HC705K1CP(4)
16-pin small outline integrated circuit (SOIC)	–40°C to +85°C	MC68HC705K1CDW
16-pin ceramic dual in-line package (cerdip)	–40°C to +85°C	MC68HC705K1CS

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°C to +85°C)

Ordering Information

Home Page:
www.freescale.com
email:
support@freescale.com
USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130
support@freescale.com
Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com
Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 2666 8080
support.japan@freescale.com
Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com
For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

RoHS-compliant and/or Pb- free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb- free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale.s Environmental Products program, go to http://www.freescale.com/epp.

freescale™
semiconductor