# THE DATASHEET OF

# MC68HC705KJ1CDW

www.win-source.net 0086-755-83957316

# MC68HC705KJ1
# MC68HRC705KJ1
# MC68HLC705KJ1

Data Sheet

**M68HC05**
**Microcontrollers**

MC68HC705KJ1
Rev. 4.1
07/2005

freescale.com

freescale™
semiconductor

# MC68HC705JK1
# MC68HRC705KJ1
# MC68HLC705KJ1

**Data Sheet**

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

http://freescale.com

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc.

© Freescale Semiconductor, Inc., 2005. All rights reserved.

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

# Revision History

| Date | Revision Level | Description | Page Number(s) |
|---|---|---|---|
| April, 2002 | 3.0 | Figure 1-4. Crystal Connections with Oscillator Internal Resistor Mask Option — changed PA7 designator to OSC1 in two places | 17 |
| | | Figure 1-5. Crystal Connections without Oscillator Internal Resistor Mask Option — changed PA7 designator to OSC1 in two places | 17 |
| | | Figure 1-6. Ceramic Resonator Connections with Oscillator Internal Resistor Mask Option — changed PA7 designator to OSC1 in two places | 18 |
| | | Figure 1-7. Ceramic Resonator Connections without Oscillator Internal Resistor Mask Option — changed PA7 designator to OSC1 in two places | 18 |
| | | Figure 1-8. External Clock Connections — changed PA7 designator to OSC1 in two places | 19 |
| | | Figure B-1. Crystal Connections — added OSC2 designation | 105 |
| | | Table B-3. MC68HLC705KJ1 (Low Frequency) Order Numbers — Corrected table title | 106 |
| May, 2003 | 4.0 | Reformatted to new publications standards. | Throughout |
| | | Figure A-2. Typical Internal Operating Frequency for Various VDD at 25°C — RC Oscillator Option Only — replaced graph | 102 |
| July, 2005 | 4.1 | Updated to meet Freescale identity guidelines. | Throughout |

# List of Chapters


Chapter 1 Introduction . . . . . 13

Chapter 2 Memory . . . . . 21

Chapter 3 Computer Operating Properly Module (COP) . . . . . 29

Chapter 4 Central Processor Unit (CPU) . . . . . 33

Chapter 5 External Interrupt Module (IRQ) . . . . . 51

Chapter 6 Low-Power Modes . . . . . 57

Chapter 7 Parallel I/O Ports (PORTS) . . . . . 63

Chapter 8 Resets and Interrupts . . . . . 71

Chapter 9 Multifunction Timer Module . . . . . 79

Chapter 10 Electrical Specifications . . . . . 85

Chapter 11 Ordering Information and Mechanical Specifications . . . . . 97

Appendix A MC68HRC705KJ1 . . . . . 101

Appendix B MC68HLC705KJ1 . . . . . 105

# Table of Contents

## Chapter 1
## Introduction


1.1 Features . . . 13
1.2 Structure . . . 14
1.3 Programmable Options . . . 15
1.4 Pin Functions . . . 15
1.4.1 $V_{DD}$ and $V_{SS}$ . . . 16
1.4.2 OSC1 and OSC2 . . . 16
1.4.2.1 Crystal Oscillator . . . 16
1.4.2.2 Ceramic Resonator Oscillator . . . 17
1.4.2.3 RC Oscillator . . . 18
1.4.2.4 External Clock . . . 19
1.4.3 $\overline{\text{RESET}}$ . . . 19
1.4.4 $\overline{\text{IRQ}}/V_{PP}$ . . . 19
1.4.5 PA0–PA7 . . . 19
1.4.6 PB2 and PB3 . . . 19


## Chapter 2
## Memory


2.1 Introduction . . . 21
2.2 Unimplemented Memory Locations . . . 21
2.3 Reserved Memory Locations . . . 21
2.4 Memory Map . . . 21
2.5 Input/Output Register Summary . . . 23
2.6 RAM . . . 25
2.7 EPROM/OTPROM . . . 25
2.7.1 EPROM/OTPROM Programming . . . 25
2.7.2 EPROM Programming Register . . . 26
2.7.3 EPROM Erasing . . . 26
2.8 Mask Option Register . . . 27
2.9 EPROM Programming Characteristics . . . 28


## Chapter 3
## Computer Operating Properly Module (COP)


3.1 Introduction . . . 29
3.2 Features . . . 29
3.3 Operation . . . 29
3.3.1 COP Watchdog Timeout . . . 29
3.3.2 COP Watchdog Timeout Period . . . 29
3.3.3 Clearing the COP Watchdog . . . 30

3.4 Interrupts . . . 30
3.5 COP Register . . . 30
3.6 Low-Power Modes . . . 30
3.6.1 Stop Mode . . . 30
3.6.2 Wait Mode . . . 31

## Chapter 4
## Central Processor Unit (CPU)

4.1 Introduction . . . 33
4.2 Features . . . 33
4.3 CPU Control Unit . . . 33
4.4 Arithmetic/Logic Unit . . . 33
4.5 CPU Registers . . . 35
4.5.1 Accumulator . . . 35
4.5.2 Index Register . . . 35
4.5.3 Stack Pointer . . . 35
4.5.4 Program Counter . . . 36
4.5.5 Condition Code Register . . . 36
4.6 Instruction Set . . . 37
4.6.1 Addressing Modes . . . 37
4.6.1.1 Inherent . . . 37
4.6.1.2 Immediate . . . 37
4.6.1.3 Direct . . . 37
4.6.1.4 Extended . . . 37
4.6.1.5 Indexed, No Offset . . . 38
4.6.1.6 Indexed, 8-Bit Offset . . . 38
4.6.1.7 Indexed, 16-Bit Offset . . . 38
4.6.1.8 Relative . . . 38
4.6.2 Instruction Types . . . 39
4.6.2.1 Register/Memory Instructions . . . 39
4.6.2.2 Read-Modify-Write Instructions . . . 40
4.6.2.3 Jump/Branch Instructions . . . 40
4.6.2.4 Bit Manipulation Instructions . . . 42
4.6.2.5 Control Instructions . . . 42
4.6.3 Instruction Set Summary . . . 43
4.7 Opcode Map . . . 48

## Chapter 5
## External Interrupt Module (IRQ)

5.1 Introduction . . . 51
5.2 Features . . . 51
5.3 Operation . . . 51
5.3.1 $\overline{\text{IRQ}}/V_{PP}$ Pin . . . 51
5.3.2 Optional External Interrupts . . . 52
5.4 IRQ Status and Control Register . . . 54
5.5 Timing . . . 55

## Chapter 6
## Low-Power Modes


6.1 Introduction . . . 57
6.2 Exiting Stop and Wait Modes . . . 57
6.3 Effects of Stop and Wait Modes . . . 58
6.3.1 Clock Generation . . . 58
6.3.1.1 STOP . . . 58
6.3.1.2 WAIT . . . 58
6.3.2 CPU . . . 58
6.3.2.1 STOP . . . 58
6.3.2.2 WAIT . . . 58
6.3.3 COP Watchdog . . . 58
6.3.3.1 STOP . . . 59
6.3.3.2 WAIT . . . 59
6.3.4 Timer . . . 59
6.3.4.1 STOP . . . 59
6.3.4.2 WAIT . . . 59
6.3.5 EPROM/OTPROM . . . 59
6.3.5.1 STOP . . . 60
6.3.5.2 WAIT . . . 60
6.4 Data-Retention Mode . . . 60
6.5 Timing . . . 60


## Chapter 7
## Parallel I/O Ports (PORTS)


7.1 Introduction . . . 63
7.2 Port A . . . 64
7.2.1 Port A Data Register . . . 64
7.2.2 Data Direction Register A . . . 64
7.2.3 Pulldown Register A . . . 65
7.2.4 Port LED Drive Capability . . . 66
7.2.5 Port A I/O Pin Interrupts . . . 66
7.3 Port B . . . 66
7.3.1 Port B Data Register . . . 66
7.3.2 Data Direction Register B . . . 67
7.3.3 Pulldown Register B . . . 68
7.4 I/O Port Electrical Characteristics . . . 69


## Chapter 8
## Resets and Interrupts


8.1 Introduction . . . 71
8.2 Resets . . . 71
8.2.1 Power-On Reset . . . 72
8.2.2 External Reset . . . 72
8.2.3 COP Watchdog Reset . . . 73
8.2.4 Illegal Address Reset . . . 73

8.3 Interrupts . . . 73
8.3.1 Software Interrupt . . . 73
8.3.2 External Interrupt . . . 73
8.3.3 Timer Interrupts . . . 75
8.3.3.1 Real-Time Interrupt . . . 75
8.3.3.2 Timer Overflow Interrupt . . . 75
8.3.4 Interrupt Processing . . . 75

## Chapter 9
## Multifunction Timer Module

9.1 Introduction . . . 79
9.2 Features . . . 79
9.3 Operation . . . 79
9.4 Interrupts . . . 80
9.5 I/O Registers . . . 81
9.5.1 Timer Status and Control Register . . . 81
9.5.2 Timer Counter Register . . . 82
9.6 Low-Power Modes . . . 83
9.6.1 Stop Mode . . . 83
9.6.2 Wait Mode . . . 83

## Chapter 10
## Electrical Specifications

10.1 Maximum Ratings . . . 85
10.2 Operating Temperature Range . . . 85
10.3 Thermal Characteristics . . . 85
10.4 Power Considerations . . . 86
10.5 5.0-V DC Electrical Characteristics . . . 87
10.6 3.3-V DC Electrical Characteristics . . . 88
10.7 Driver Characteristics . . . 89
10.8 Typical Supply Currents . . . 91
10.9 EPROM Programming Characteristics . . . 92
10.10 Control Timing . . . 92

## Chapter 11
## Ordering Information and Mechanical Specifications

11.1 Introduction . . . 97
11.2 MCU Order Numbers . . . 97
11.3 16-Pin PDIP — Case #648 . . . 98
11.4 16-Pin SOIC — Case #751G . . . 98
11.5 16-Pin Cerdip — Case #620A . . . 99

## Appendix A
## MC68HRC705KJ1


A.1 Introduction . . . . . 101
A.2 RC Oscillator Connections . . . . . 101
A.3 Typical Internal Operating Frequency for RC Oscillator Option . . . . . 102
A.4 RC Oscillator Connections (No External Resistor) . . . . . 103
A.5 Typical Internal Operating Frequency Versus Temperature (No External Resistor) . . . . . 104
A.6 Package Types and Order Numbers . . . . . 104


## Appendix B
## MC68HLC705KJ1


B.1 Introduction . . . . . 105
B.2 DC Electrical Characteristics . . . . . 105
B.3 Package Types and Order Numbers . . . . . 106

# Chapter 1
# Introduction

## 1.1 Features

Features on the MC68HC705KJ1 include:

- Robust noise immunity
- 4.0-MHz internal operating frequency at 5.0 V
- 1240 Bytes of EPROM/OTPROM (electrically programmable read-only memory/one-time programmable read-only memory), including eight bytes for user vectors
- 64 bytes of user RAM
- Peripheral modules:
  - 15-stage multifunction timer
  - Computer operating properly (COP) watchdog
- 10 bidirectional input/output (I/O) lines, including:
  - 10-mA sink capability on all I/O pins
  - Software programmable pulldowns on all I/O pins
  - Keyboard scan with selectable interrupt on four I/O pins
  - 5.5-mA source capability on six I/O pins
- Selectable sensitivity on external interrupt (edge- and level-sensitive or edge-sensitive only)
- On-chip oscillator with connections for:
  - Crystal
  - Ceramic resonator
  - Resistor-capacitor (RC) oscillator (MC68HRC705KJ1) with or without external resistor
  - External clock
  - Low-speed (32-kHz) crystal (MC68HLC705KJ1)
- Memory-mapped I/O registers
- Fully static operation with no minimum clock speed
- Power-saving stop, halt, wait, and data-retention modes
- External interrupt mask bit and acknowledge bit
- Illegal address reset
- Internal steering diode and pullup resistor from $\overline{\text{RESET}}$ pin to $V_{DD}$
- Selectable EPROM security[1]
- Selectable oscillator bias resistor

1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the EPROM/OTPROM difficult for unauthorized users.

## 1.2 Structure

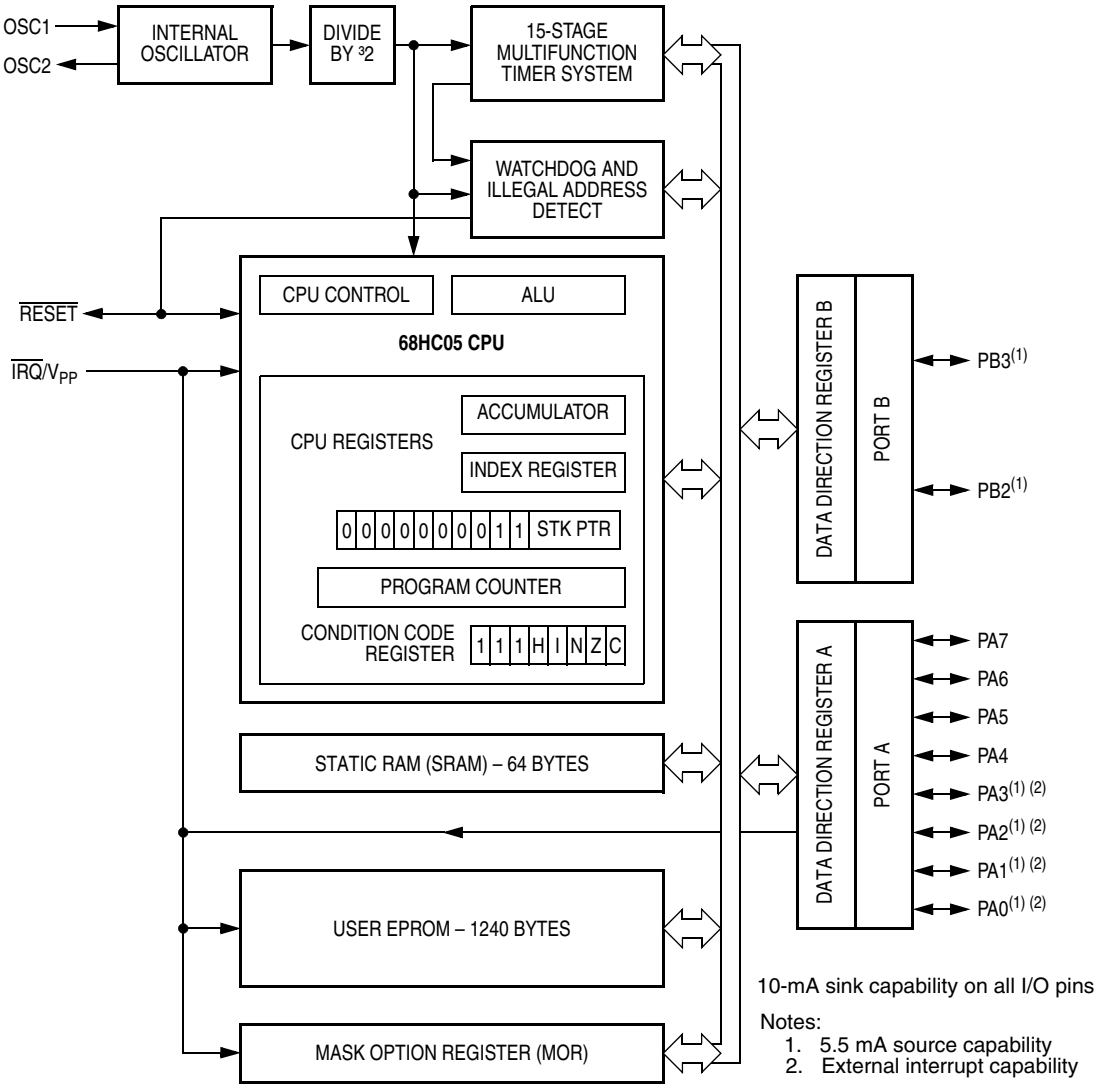



**Figure 1-1. Block Diagram**

## 1.3 Programmable Options

The options in Table 1-1 are programmable in the mask option register.

**Table 1-1. Programmable Options**

| Feature | Option |
|---|---|
| COP watchdog timer | Enabled or disabled |
| External interrupt triggering | Edge-sensitive only or edge- and level-sensitive |
| Port A $\overline{IRQ}$ pin interrupts | Enabled or disabled |
| Port pulldown resistors | Enabled or disabled |
| STOP instruction mode | Stop mode or halt mode |
| Crystal oscillator internal resistor | Enabled or disabled |
| EPROM security | Enabled or disabled |
| Short oscillator delay counter | Enabled or disabled |

## 1.4 Pin Functions

Pin assignments are shown in Figure 1-2 with the functions described in the following subsections.

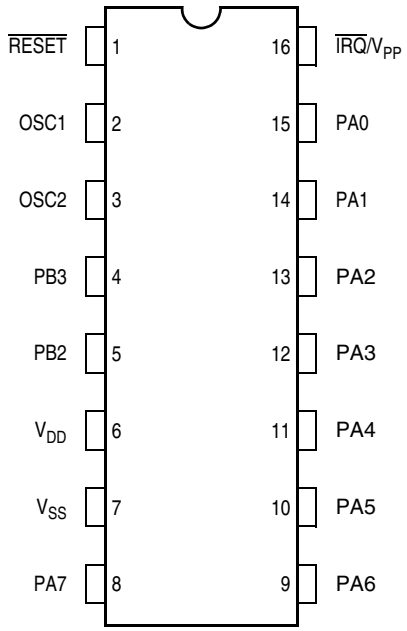



**Figure 1-2. Pin Assignments**

### 1.4.1 $V_{DD}$ and $V_{SS}$

$V_{DD}$ and $V_{SS}$ are the power supply and ground pins. The MCU operates from a single 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, as Figure 1-3 shows, by placing the bypass capacitors as close as possible to the MCU. C2 is an optional bulk current bypass capacitor for use in applications that require the port pins to source high current levels.

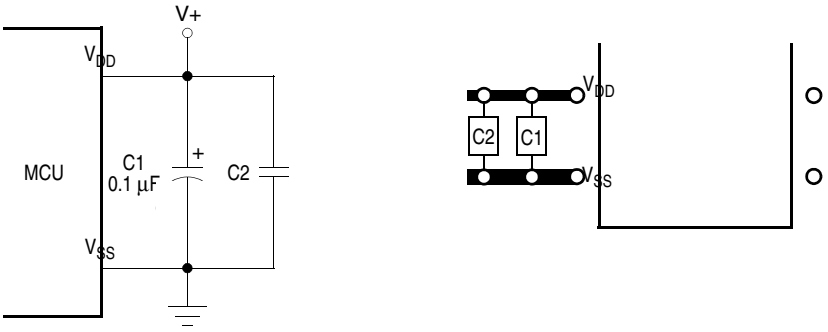



**Figure 1-3. Bypassing Layout Recommendation**

### 1.4.2 OSC1 and OSC2

The OSC1 and OSC2 pins are the connections for the on-chip oscillator. The oscillator can be driven by any of the following:

1. Standard crystal (See Figure 1-4 and Figure 1-5.)
2. Ceramic resonator (See Figure 1-6 and Figure 1-7.)
3. Resistor/capacitor (RC) oscillator (Refer to Appendix A MC68HRC705KJ1.)
4. External clock signal as shown in (See Figure 1-8.)
5. Low speed (32 kHz) crystal connections (Refer to Appendix B MC68HLC705KJ1.)

The frequency, $f_{OSC}$, of the oscillator or external clock source is divided by two to produce the internal operating frequency, $f_{OP}$.

#### *1.4.2.1 Crystal Oscillator*

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

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

***NOTE***

*Use an AT-cut crystal and not an AT-strip crystal because the MCU can overdrive an AT-strip crystal.*

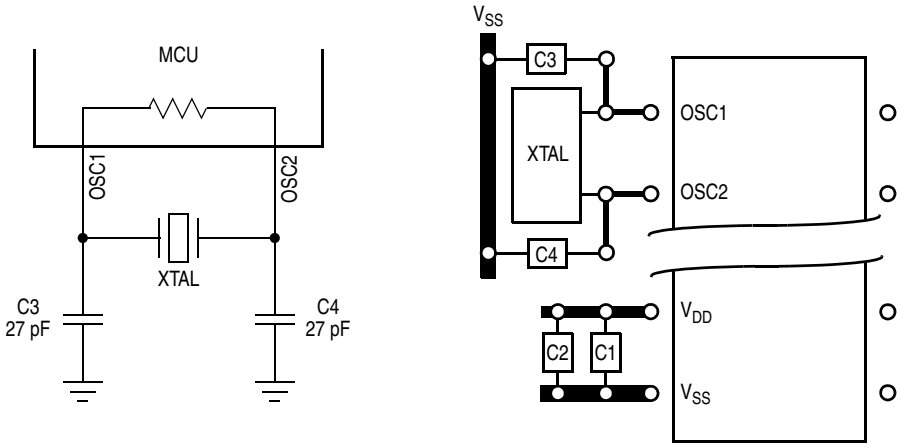



**Figure 1-4. Crystal Connections with Oscillator Internal Resistor Mask Option**

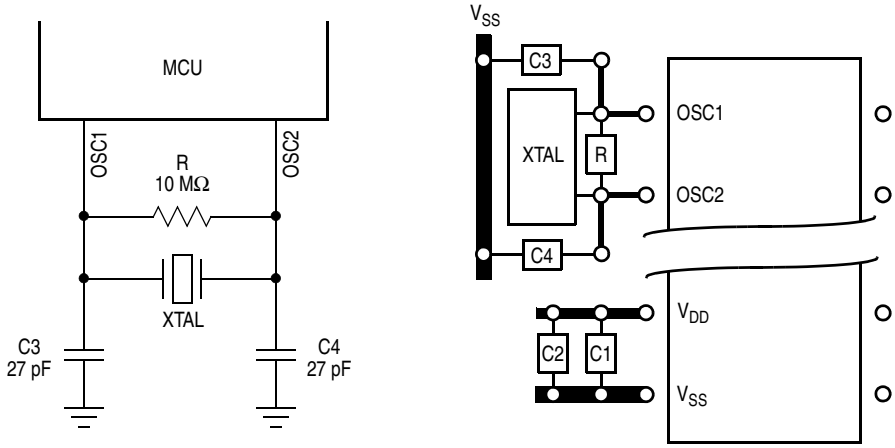



**Figure 1-5. Crystal Connections without Oscillator Internal Resistor Mask Option**

### *1.4.2.2 Ceramic Resonator Oscillator*

To reduce cost, use a ceramic resonator instead of the crystal. The circuits shown in Figure 1-6 and Figure 1-7 show ceramic resonator circuits. Follow the resonator manufacturer's recommendations, as the resonator parameters determine the external component values required for maximum stability and reliable starting. The load capacitance values used in the oscillator circuit design should include all stray capacitances.

Mount the resonator and components as close as possible to the pins for startup stabilization and to minimize output distortion. An internal startup resistor of approximately 2 MΩ is provided between OSC1 and OSC2 as a programmable mask option.

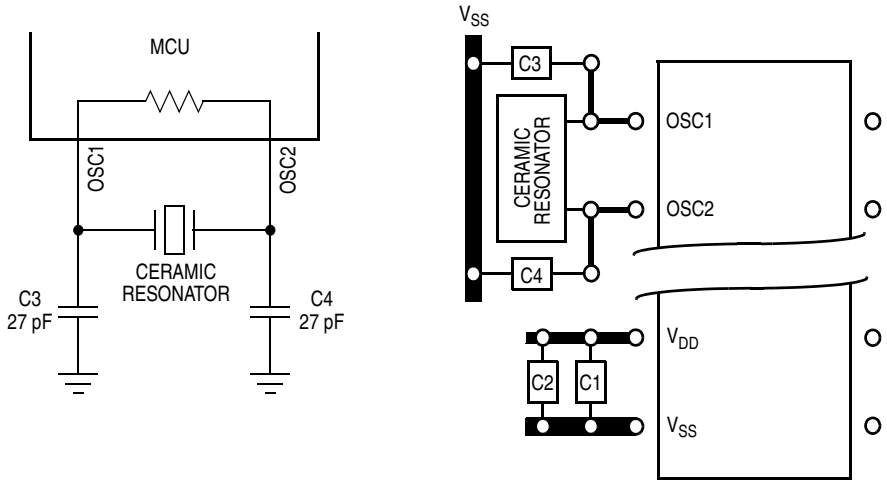



**Figure 1-6. Ceramic Resonator Connections with Oscillator Internal Resistor Mask Option**

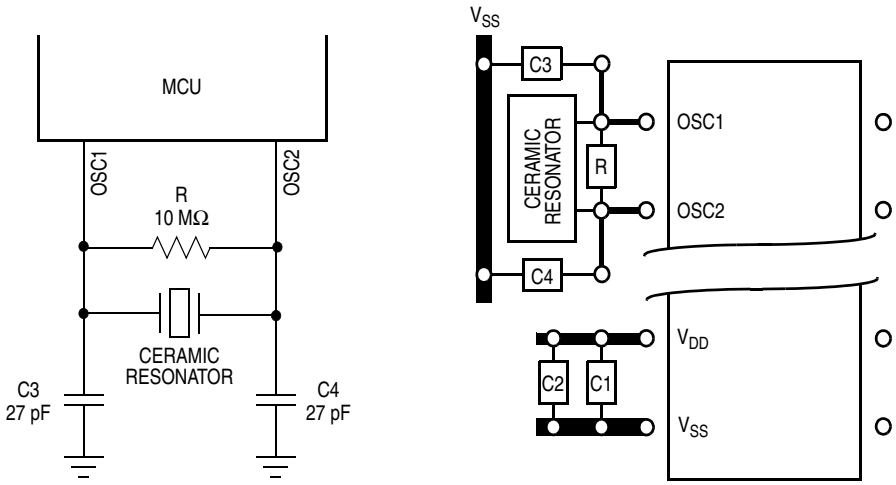



**Figure 1-7. Ceramic Resonator Connections without Oscillator Internal Resistor Mask Option**

### *1.4.2.3 RC Oscillator*

Refer to Appendix A MC68HRC705KJ1.

#### *1.4.2.4 External Clock*

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

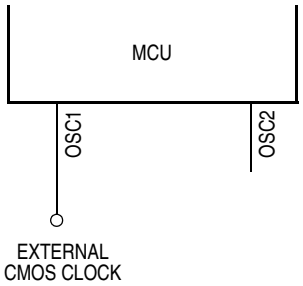



**Figure 1-8. External Clock Connections**

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

Applying a logic 0 to the $\overline{\text{RESET}}$ pin forces the MCU to a known startup state. An internal reset also pulls the $\overline{\text{RESET}}$ pin low. An internal resistor to $V_{DD}$ pulls the $\overline{\text{RESET}}$ pin high. A steering diode between the $\overline{\text{RESET}}$ and $V_{DD}$ pins discharges any $\overline{\text{RESET}}$ pin voltage when power is removed from the MCU. The $\overline{\text{RESET}}$ pin contains an internal Schmitt trigger to improve its noise immunity as an input. Refer to Chapter 8 Resets and Interrupts for more information.

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

The external interrupt/programming voltage pin ($\overline{\text{IRQ}}/V_{PP}$) drives the asynchronous IRQ interrupt function of the CPU. Additionally, it is used to program the user EPROM and mask option register. (See Chapter 2 Memory and Chapter 5 External Interrupt Module (IRQ).)

The LEVEL bit in the mask option register provides negative edge-sensitive triggering or both negative edge-sensitive and low level-sensitive triggering for the interrupt function.

If level-sensitive triggering is selected, the $\overline{\text{IRQ}}/V_{PP}$ input requires an external resistor to $V_{DD}$ for wired-OR operation. If the $\overline{\text{IRQ}}/V_{PP}$ pin is not used, it must be tied to the $V_{DD}$ supply.

The $\overline{\text{IRQ}}/V_{PP}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. The voltage on this pin should not exceed $V_{DD}$ except when the pin is being used for programming the EPROM.

***NOTE***

*The mask option register can enable the PA0–PA3 pins to function as external interrupt pins.*

### 1.4.5 PA0–PA7

These eight input/output (I/O) lines comprise port A, a general-purpose bidirectional I/O port. (See Chapter 5 External Interrupt Module (IRQ) for information on PA0–PA3 external interrupts.)

### 1.4.6 PB2 and PB3

These two I/O lines comprise port B, a general-purpose bidirectional I/O port.

# Chapter 2
# Memory

## 2.1 Introduction

This section provides:

- Memory map (Figure 2-1)
- Summary of the input/output registers (Figure 2-2)
- Description of:
  - Random-access memory (RAM)
  - EPROM/OTPROM (electrically programmable read-only memory/one-time programmable read-only memory)
  - Mask option register

Memory features include:

- 1232 Bytes of User EPROM, Plus Eight Bytes for User Vectors
- 64 Bytes of User RAM

## 2.2 Unimplemented Memory Locations

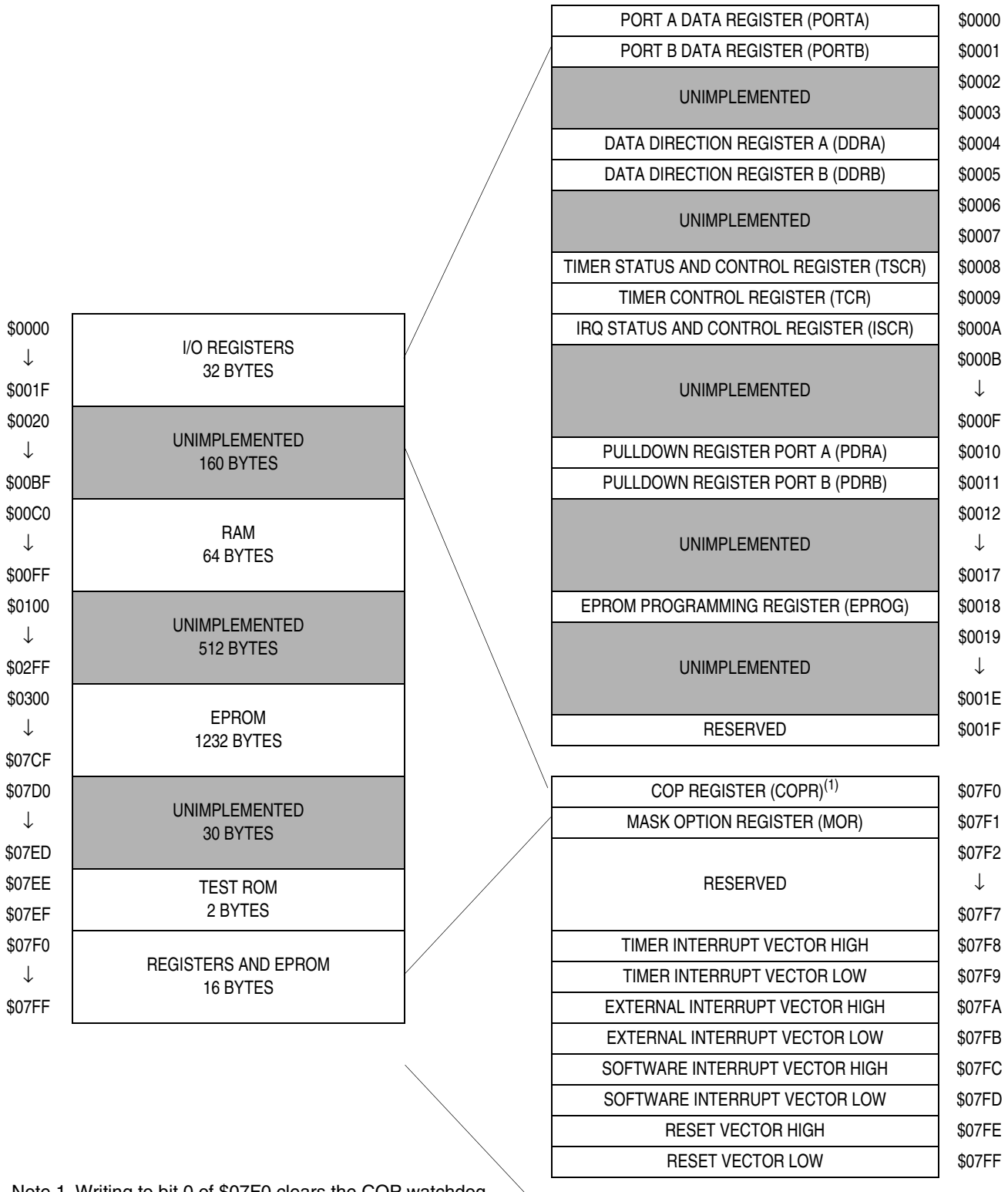
Accessing an unimplemented location can have unpredictable effects on MCU operation. In Figure 2-2 and in register figures in this document, unimplemented locations are shaded.

## 2.3 Reserved Memory Locations

Accessing a reserved location can have unpredictable effects on MCU operation. In Figure 2-2 and in register figures in this document, reserved locations are marked with the word Reserved or with the letter R.

## 2.4 Memory Map

See Figure 2-1.




Note 1. Writing to bit 0 of $07F0 clears the COP watchdog.

**Figure 2-1. Memory Map**

## 2.5 Input/Output Register Summary

| Addr. | Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $0000 | Port A Data Register (PORTA) See page 64. | Read: Write: | PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| | | Reset: | Unaffected by reset | | | | | | | |
| $0001 | Port B Data Register (PORTB) See page 66. | Read: | 0 | 0 | Refer to Chapter 7 Parallel I/O Ports (PORTS) | | PB3 | PB2 | Refer to Chapter 7 Parallel I/O Ports (PORTS) | |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| $0002 | Unimplemented | | | | | | | | | |
| $0003 | Unimplemented | | | | | | | | | |
| $0004 | Data Direction Register A (DDRA) See page 64. | Read: Write: | DDRA7 | DDRA6 | DDRA5 | DDRA4 | DDRA3 | DDRA2 | DDRA1 | DDRA0 |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $0005 | Data Direction Register B (DDRB) See page 67. | Read: | 0 | 0 | Refer to Chapter 7 Parallel I/O Ports (PORTS) | | DDRB3 | DDRB2 | Refer to Chapter 7 Parallel I/O Ports (PORTS) | |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $0006 | Unimplemented | | | | | | | | | |
| $0007 | Unimplemented | | | | | | | | | |
| $0008 | Timer Status and Control Register (TSCR) See page 81. | Read: | TOF | RTIF | TOIE | RTIE | 0 | 0 | RT1 | RT0 |
| | | Write: | | | | | TOFR | RTIFR | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| $0009 | Timer Counter Register (TCR) See page 82. | Read: | TCR7 | TCR6 | TCR5 | TCR4 | TCR3 | TCR2 | TCR1 | TCR0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $000A | IRQ Status and Control Register (ISCR) See page 54. | Read: | IRQE | 0 | 0 | 0 | IRQF | 0 | 0 | 0 |
| | | Write: | | | | R | | | IRQR | |
| | | Reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented R = Reserved U = Unaffected

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

| Addr. | Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $000B | Unimplemented | | | | | | | | | |
| ↓ | | | | | | | | | | |
| $000F | Unimplemented | | | | | | | | | |
| $0010 | Pulldown Register Port A (PDRA) See page 65. | Read: | | | | | | | | |
| | | Write: | PDIA7 | PDIA6 | PDIA5 | PDIA4 | PDIA3 | PDIA2 | PDIA1 | PDIA0 |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $0011 | Pulldown Register Port B (PDRB) See page 68. | Read: | | | | | | | | |
| | | Write: | | | Refer to Chapter 7 Parallel I/O Ports (PORTS) | | PDIB3 | PDIB2 | Refer to Chapter 7 Parallel I/O Ports (PORTS) | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $0012 | Unimplemented | | | | | | | | | |
| ↓ | | | | | | | | | | |
| $0017 | Unimplemented | | | | | | | | | |
| $0018 | EPROM Programming Register (EPROG) See page 26. | Read: | 0 | 0 | 0 | 0 | 0 | ELAT | MPGM | EPGM |
| | | Write: | | R | R | R | R | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $0019 | Unimplemented | | | | | | | | | |
| ↓ | | | | | | | | | | |
| $001E | Unimplemented | | | | | | | | | |
| $001F | Reserved | | R | R | R | R | R | R | R | R |
| $07F0 | COP Register (COPR) See page 30. | Read: | | | | | | | | |
| | | Write: | | | | | | | | COPC |
| | | Reset: | U | U | U | U | U | U | U | 0 |
| $07F1 | Mask Option Register (MOR) See page 27. | Read: | SOSCD | EPMSEC | OSCRES | SWAIT | PDI | PIRQ | LEVEL | COPEN |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |

= Unimplemented R = Reserved U = Unaffected

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

## 2.6 RAM

The 64 addresses from $00C0 to $00FF serve as both the user RAM and the stack RAM. Before processing an interrupt, the CPU uses five bytes of the stack to save the contents of the CPU registers. During a subroutine call, the CPU uses two bytes of the stack to store the return address. The stack pointer decrements when the CPU stores a byte on the stack and increments when the CPU retrieves a byte from the stack.

***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.*

## 2.7 EPROM/OTPROM

An MCU with a quartz window has 1240 bytes of erasable, programmable ROM (EPROM). The quartz window allows EPROM erasure with ultraviolet light.

***NOTE***

*Keep the quartz window covered with an opaque material except when erasing the MCU. Ambient light can affect MCU operation.*

In an MCU without the quartz window, the EPROM cannot be erased and serves as 1240 bytes of one-time programmable ROM (OTPROM).

The following addresses are user EPROM/OTPROM locations:

- $0300–$07CF
- $07F8–$07FF, used for user-defined interrupt and reset vectors

The COP register (COPR) is an EPROM/OTPROM location at address $07F0.

The mask option register (MOR) is an EPROM/OTPROM location at address $07F1.

### 2.7.1 EPROM/OTPROM Programming

The two ways to program the EPROM/OTPROM are:

- Manipulating the control bits in the EPROM programming register to program the EPROM/OTPROM on a byte-by-byte basis
- Programming the EPROM/OTPROM with the M68HC705J In-Circuit Simulator (M68HC705JICS) available from Freescale

## 2.7.2 EPROM Programming Register

The EPROM programming register (EPROG) contains the control bits for programming the EPROM/OTPROM.

Address: $0018

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

= Unimplemented  R = Reserved

**Figure 2-3. EPROM Programming Register (EPROG)**

**ELAT — EPROM Bus Latch Bit**

This read/write bit latches the address and data buses for EPROM/OTPROM programming. Clearing the ELAT bit automatically clears the EPGM bit. EPROM/OTPROM data cannot be read while the ELAT bit is set. Reset clears the ELAT bit.

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

**MPGM — MOR Programming Bit**

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

1 = Programming voltage applied to MOR
0 = Programming voltage not applied to MOR

**EPGM — EPROM Programming Bit**

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

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

***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**

Take the following steps to program a byte of EPROM/OTPROM:

1. Apply the programming voltage, $V_{PP}$, to the $\overline{IRQ}/V_{PP}$ pin.
2. Set the ELAT bit.
3. Write to any EPROM/OTPROM address.
4. Set the EPGM bit and wait for a time, $t_{EPGM}$.
5. Clear the ELAT bit.

## 2.7.3 EPROM Erasing

The erased state of an EPROM bit is logic 0. Erase the EPROM by exposing it to 15 $Ws/cm^2$ of ultraviolet light with a wavelength of 2537 angstroms. Position the ultraviolet light source one inch from the EPROM. Do not use a shortwave filter.

## 2.8 Mask Option Register

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

- COP watchdog (enable or disable)
- External interrupt pin triggering (edge-sensitive only or edge- and level-sensitive)
- Port A external interrupts (enable or disable)
- Port pulldown resistors (enable or disable)
- STOP instruction (stop mode or halt mode)
- Crystal oscillator internal resistor (enable or disable)
- EPROM security (enable or disable)
- Short oscillator delay (enable or disable)

Take the following steps to program the mask option register (MOR):

1. Apply the programming voltage, $V_{PP}$, to the $\overline{IRQ}/V_{PP}$ pin.
2. Write to the MOR.
3. Set the MPGM bit and wait for a time, $t_{MPGM}$.
4. Clear the MPGM bit.
5. Reset the MCU.

Address: $07F1

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Read:<br>Write: | SOSCD | EPMSEC | OSCRES | SWAIT | SWPDI | PIRQ | LEVEL | COPEN |
| Reset: | Unaffected by reset | | | | | | | |

**Figure 2-4. Mask Option Register (MOR)**

**SOSCD — Short Oscillator Delay Bit**

The SOSCD bit controls the oscillator stabilization counter. The normal stabilization delay following reset or exit from stop mode is 4064 $t_{cyc}$. Setting SOSCD enables a 128 $t_{cyc}$ stabilization delay.

1 = Short oscillator delay enabled
0 = Short oscillator delay disabled

**EPMSEC — EPROM Security Bit**

The EPMSEC bit controls access to the EPROM/OTPROM.

1 = External access to EPROM/OTPROM denied
0 = External access to EPROM/OTPROM not denied

**OSCRES — Oscillator Internal Resistor Bit**

The OSCRES bit enables a 2-MΩ internal resistor in the oscillator circuit.

1 = Oscillator internal resistor enabled
0 = Oscillator internal resistor disabled

***NOTE***

*Program the OSCRES bit to logic 0 in devices using low-speed crystal or RC oscillators with external resistor.*

**SWAIT — Stop-to-Wait Conversion Bit**

The SWAIT bit enables halt mode. When the SWAIT bit is set, the CPU interprets the STOP instruction as a WAIT instruction, and the MCU enters halt mode. Halt mode is the same as wait mode, except that an oscillator stabilization delay of 1 to 4064 $t_{cyc}$ occurs after exiting halt mode.

1 = Halt mode enabled
0 = Halt mode not enabled

**SWPDI — Software Pulldown Inhibit Bit**

The SWPDI bit inhibits software control of the I/O port pulldown devices. The SWPDI bit overrides the pulldown inhibit bits in the port pulldown inhibit registers.

1 = Software pulldown control inhibited
0 = Software pulldown control not inhibited

**PIRQ — Port A External Interrupt Bit**

The PIRQ bit enables the PA0–PA3 pins to function as external interrupt pins.

1 = PA0–PA3 enabled as external interrupt pins
0 = PA0–PA3 not enabled as external interrupt pins

**LEVEL —External Interrupt Sensitivity Bit**

The LEVEL bit controls external interrupt triggering sensitivity.

1 = External interrupts triggered by active edges and active levels
0 = External interrupts triggered only by active edges

**COPEN — COP Enable Bit**

The COPEN bit enables the COP watchdog.

1 = COP watchdog enabled
0 = COP watchdog disabled

## 2.9 EPROM Programming Characteristics

**Table 2-1. EPROM Programming Characteristics(1)**

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

1. $V_{DD}$ = 5.0 Vdc ± 10%, $V_{SS}$ = 0 Vdc, $T_A$ = –40°C to +85°C

# Chapter 3
# Computer Operating Properly Module (COP)

## 3.1 Introduction

The computer operating properly (COP) watchdog resets the MCU in case of software failure. Software that is operating properly periodically services the COP watchdog and prevents COP reset. The COP watchdog function is programmable by the COPEN bit in the mask option register.

## 3.2 Features

The computer operating properly module (COP) includes these features:

- Protection from runaway software
- Wait mode and halt mode operations

## 3.3 Operation

Operation of the COP module is discussed here.

### 3.3.1 COP Watchdog Timeout

Four counter stages at the end of the timer make up the COP watchdog. The COP resets the MCU if the timeout period occurs before the COP watchdog timer is cleared by application software and the $\overline{IRQ}/V_{PP}$ pin voltage is between $V_{SS}$ and $V_{DD}$. Periodically clearing the counter starts a new timeout period and prevents COP reset. A COP watchdog timeout indicates that the software is not executing instructions in the correct sequence.

***NOTE***

*The internal clock drives the COP watchdog. Therefore, the COP watchdog cannot generate a reset for errors that cause the internal clock to stop. The COP watchdog depends on a power supply voltage at or above a minimum specification and is not guaranteed to protect against brownout.*

### 3.3.2 COP Watchdog Timeout Period

The COP watchdog timer function is implemented by dividing the output of the real-time interrupt circuit (RTI) by eight. The RTI select bits in the timer status and control register control RTI output, and the selected output drives the COP watchdog. (See timer status and control register in Chapter 9 Multifunction Timer Module.)

***NOTE***

*The minimum COP timeout period is seven times the RTI period. The COP is cleared asynchronously with the value in the RTI divider; hence, the COP timeout period will vary between 7x and 8x the RTI period.*

### 3.3.3 Clearing the 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 $07F0 (see Figure 3-1). Clearing the COP bit disables the COP watchdog timer regardless of the $\overline{IRQ}/V_{PP}$ pin voltage.

If the main program executes within the COP timeout period, the clearing routine should be executed only once. If the main program takes longer than the COP timeout period, the clearing routine must be executed more than once.

***NOTE***

*Place the clearing routine in the main program and not in an interrupt routine. Clearing the COP watchdog in an interrupt routine might prevent COP watchdog timeouts even though the main program is not operating properly.*

## 3.4 Interrupts

The COP watchdog does not generate interrupts.

## 3.5 COP Register

The COP register (COPR) is a write-only register that returns the contents of EPROM location $07F0 when read.

Address: $07F0

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Read: | | | | | | | | |
| Write: | | | | | | | | COPC |
| Reset: | U | U | U | U | U | U | U | 0 |

= Unimplemented U = Unaffected

**Figure 3-1. COP Register (COPR)**

**COPC — COP Clear Bit**

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

## 3.6 Low-Power Modes

The STOP and WAIT instructions have the following effects on the COP watchdog.

### 3.6.1 Stop Mode

The STOP instruction clears the COP watchdog counter and disables the clock to the COP watchdog.

***NOTE***

*To prevent the STOP instruction from disabling the COP watchdog, program the stop-to-wait conversion bit (SWAIT) in the mask option register to logic 1.*

Upon exit from stop mode by external reset:

- The counter begins counting from $0000.
- The counter is cleared again after the oscillator stabilization delay and begins counting from $0000 again.

Upon exit from stop mode by external interrupt:

- The counter begins counting from $0000.
- The counter is not cleared again after the oscillator stabilization delay and continues counting throughout the oscillator stabilization delay.

***NOTE***

*Immediately after exiting stop mode by external interrupt, service the COP to ensure a full COP timeout period.*

### 3.6.2 Wait Mode

The WAIT instruction has no effect on the COP watchdog.

***NOTE***

*To prevent a COP timeout during wait mode, exit wait mode periodically to service the COP.*

# Chapter 4
# Central Processor Unit (CPU)

## 4.1 Introduction

The central processor unit (CPU) consists of a CPU control unit, an arithmetic/logic unit (ALU), and five CPU registers. The CPU control unit fetches and decodes instructions. The ALU executes the instructions. The CPU registers contain data, addresses, and status bits that reflect the results of CPU operations.

## 4.2 Features

Features of the CPU include:

- 4.0-MHz bus frequency on standard part
- 8-bit accumulator
- 8-bit index register
- 11-bit program counter
- 6-bit stack pointer
- Condition code register with five status flags
- 62 instructions
- 8 addressing modes
- Power-saving stop, wait, halt, and data-retention modes

The programming model is shown in Figure 4-1.

## 4.3 CPU Control Unit

The CPU control unit fetches and decodes instructions during program operation. The control unit selects the memory locations to read and write and coordinates the timing of all CPU operations.

## 4.4 Arithmetic/Logic Unit

The arithmetic/logic unit (ALU) performs the arithmetic, logic, and manipulation operations decoded from the instruction set by the CPU control unit. The ALU produces the results called for by the program and sets or clears status and control bits in the condition code register (CCR).

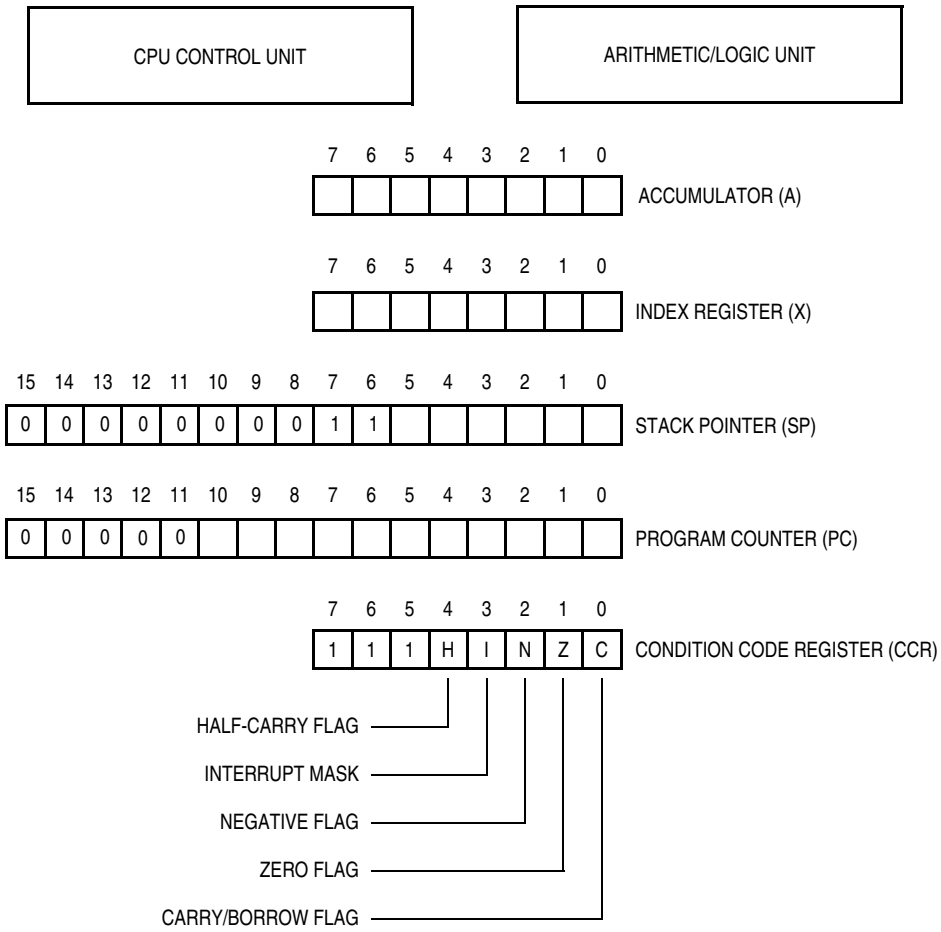



**Figure 4-1. Programming Model**

## 4.5 CPU Registers

The M68HC05 CPU contains five registers that control and monitor MCU operation:

- Accumulator
- Index register
- Stack pointer
- Program counter
- Condition code register

CPU registers are not memory mapped.

### 4.5.1 Accumulator

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

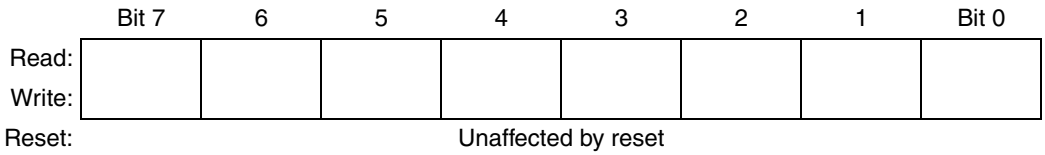



**Figure 4-2. Accumulator (A)**

### 4.5.2 Index Register

In the indexed addressing modes, the CPU uses the byte in the index register to determine the conditional address of the operand. The index register also can serve as a temporary storage location or a counter.

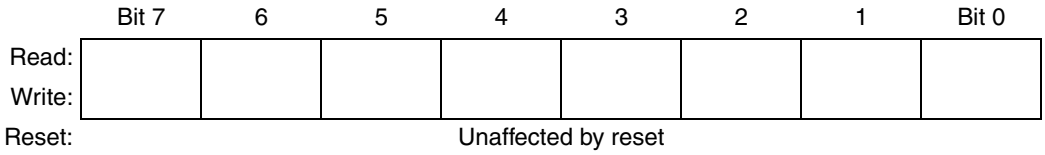



**Figure 4-3. Index Register (X)**

### 4.5.3 Stack Pointer

The stack pointer is a 16-bit register that contains the address of the next location on the stack. During a reset or after the reset stack pointer instruction (RSP), the stack pointer is preset to $00FF. The address in the stack pointer decrements after a byte is stacked and increments before a byte is unstacked.

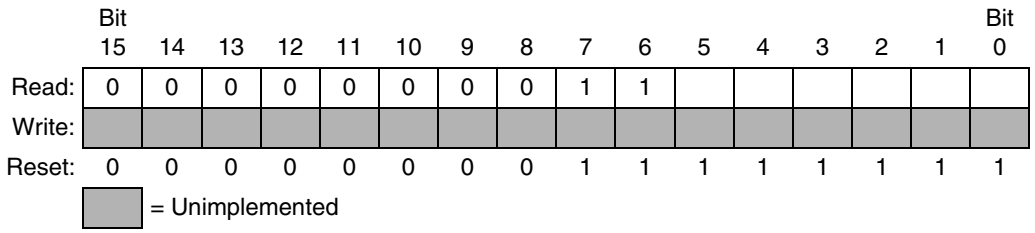



**Figure 4-4. Stack Pointer (SP)**

The 10 most significant bits of the stack pointer are permanently fixed at 0000000011, so the stack pointer produces addresses from $00C0 to $00FF. If subroutines and interrupts use more than 64 stack locations, the stack pointer wraps around to address $00FF and begins writing over the previously stored data. A subroutine uses two stack locations; an interrupt uses five locations.

### 4.5.4 Program Counter

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

Normally, the address in the program counter automatically increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, and interrupt operations load the program counter with an address other than that of the next sequential location.

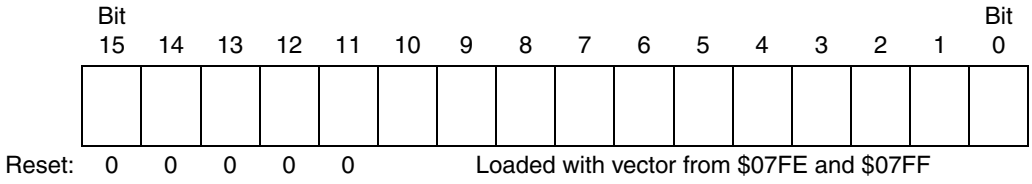



**Figure 4-5. Program Counter (PC)**

### 4.5.5 Condition Code Register

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

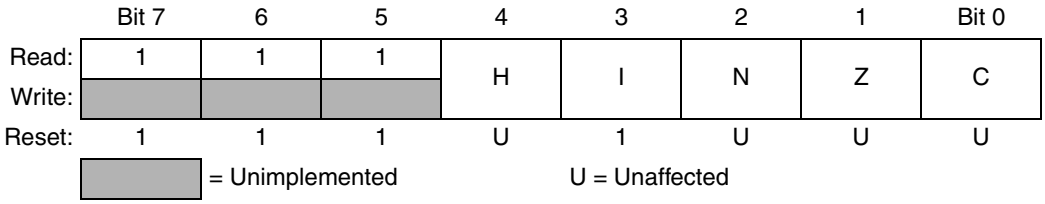



**Figure 4-6. Condition Code Register (CCR)**

**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 or ADC operation. The half-carry flag is required for binary-coded decimal (BCD) arithmetic operations.

**I — Interrupt Mask**

Setting the interrupt mask disables interrupts. If an interrupt request occurs while the interrupt mask is 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 logic 1, the interrupt request is latched. Normally, the CPU processes the latched interrupt request as soon as the interrupt mask is cleared again.

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

**N — Negative Flag**

The CPU sets the negative flag when an ALU operation produces a negative result.

**Z — Zero Flag**

The CPU sets the zero flag when an ALU operation produces a result of $00.

**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 flag.

## 4.6 Instruction Set

The MCU instruction set has 62 instructions and uses eight addressing modes.

### 4.6.1 Addressing Modes

The 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

#### *4.6.1.1 Inherent*

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

#### *4.6.1.2 Immediate*

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

#### *4.6.1.3 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.

#### *4.6.1.4 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.

#### *4.6.1.5 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 input/output (I/O) location.

#### *4.6.1.6 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.

#### *4.6.1.7 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.

#### *4.6.1.8 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.

### 4.6.2 Instruction Types

The MCU instructions fall into the following five categories:

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

#### *4.6.2.1 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 4-1. 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 |

#### 4.6.2.2 Read-Modify-Write Instructions

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

**NOTE**

*Do not use read-modify-write instructions on registers with write-only bits.*

**Table 4-2. 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.

#### 4.6.2.3 Jump/Branch Instructions

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

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.

***NOTE***

*Do not use BRCLR or BRSET instructions on registers with write-only bits.*

**Table 4-3. 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 |

#### 4.6.2.4 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 4-4. Bit Manipulation Instructions**

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

***NOTE***

*Do not use bit manipulation instructions on registers with write-only bits.*

#### 4.6.2.5 Control Instructions

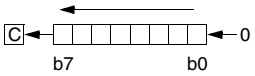
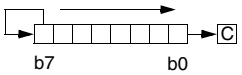
These instructions act on CPU registers and control CPU operation during program execution.

**Table 4-5. 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 |

### 4.6.3 Instruction Set Summary

**Table 4-6. 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*<br>ADC *opr*<br>ADC *opr*<br>ADC *opr*,X<br>ADC *opr*,X<br>ADC ,X | Add with Carry | A ← (A) + (M) + (C) | ↕ | — | ↕ | ↕ | ↕ | IMM<br>DIR<br>EXT<br>IX2<br>IX1<br>IX | A9<br>B9<br>C9<br>D9<br>E9<br>F9 | ii<br>dd<br>hh ll<br>ee ff<br>ff | 2<br>3<br>4<br>5<br>4<br>3 |
| ADD #*opr*<br>ADD *opr*<br>ADD *opr*<br>ADD *opr*,X<br>ADD *opr*,X<br>ADD ,X | Add without Carry | A ← (A) + (M) | ↕ | — | ↕ | ↕ | ↕ | IMM<br>DIR<br>EXT<br>IX2<br>IX1<br>IX | AB<br>BB<br>CB<br>DB<br>EB<br>FB | ii<br>dd<br>hh ll<br>ee ff<br>ff | 2<br>3<br>4<br>5<br>4<br>3 |
| AND #*opr*<br>AND *opr*<br>AN*D opr*<br>AND *opr*,X<br>AND *opr*,X<br>AND ,X | Logical AND | A ← (A) ∧ (M) | — | — | ↕ | ↕ | — | IMM<br>DIR<br>EXT<br>IX2<br>IX1<br>IX | A4<br>B4<br>C4<br>D4<br>E4<br>F4 | ii<br>dd<br>hh ll<br>ee ff<br>ff | 2<br>3<br>4<br>5<br>4<br>3 |
| ASL *opr*<br>ASLA<br>ASLX<br>ASL *opr*,X<br>ASL ,X | Arithmetic Shift Left (Same as LSL) |  | — | — | ↕ | ↕ | ↕ | DIR<br>INH<br>INH<br>IX1<br>IX | 38<br>48<br>58<br>68<br>78 | dd<br><br><br>ff<br> | 5<br>3<br>3<br>6<br>5 |
| ASR *opr*<br>ASRA<br>ASRX<br>ASR *opr*,X<br>ASR ,X | Arithmetic Shift Right |  | — | — | ↕ | ↕ | ↕ | DIR<br>INH<br>INH<br>IX1<br>IX | 37<br>47<br>57<br>67<br>77 | dd<br><br><br>ff<br> | 5<br>3<br>3<br>6<br>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)<br>DIR (b1)<br>DIR (b2)<br>DIR (b3)<br>DIR (b4)<br>DIR (b5)<br>DIR (b6)<br>DIR (b7) | 11<br>13<br>15<br>17<br>19<br>1B<br>1D<br>1F | dd<br>dd<br>dd<br>dd<br>dd<br>dd<br>dd<br>dd | 5<br>5<br>5<br>5<br>5<br>5<br>5<br>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 |
| 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 |

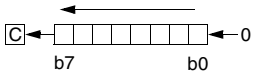
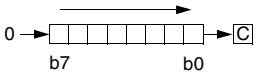
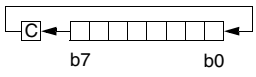
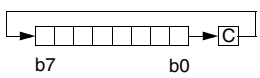
## Table 4-6. Instruction Set Summary (Sheet 2 of 6)

| Source Form | Operation | Description | Effect on CCR | | | | | Address Mode | Opcode | Operand | Cycles |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | H | I | N | Z | C | | | | |
| BIT #*opr*<br>BIT *opr*<br>BIT *opr*<br>BIT *opr*,X<br>BIT *opr*,X<br>BIT ,X | Bit Test Accumulator with Memory Byte | (A) ∧ (M) | — | — | ↕ | ↕ | — | IMM<br>DIR<br>EXT<br>IX2<br>IX1<br>IX | A5<br>B5<br>C5<br>D5<br>E5<br>F5 | ii<br>dd<br>hh ll<br>ee ff<br>ff | 2<br>3<br>4<br>5<br>4<br>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)<br>DIR (b1)<br>DIR (b2)<br>DIR (b3)<br>DIR (b4)<br>DIR (b5)<br>DIR (b6)<br>DIR (b7) | 01<br>03<br>05<br>07<br>09<br>0B<br>0D<br>0F | dd rr<br>dd rr<br>dd rr<br>dd rr<br>dd rr<br>dd rr<br>dd rr<br>dd rr | 5<br>5<br>5<br>5<br>5<br>5<br>5<br>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)<br>DIR (b1)<br>DIR (b2)<br>DIR (b3)<br>DIR (b4)<br>DIR (b5)<br>DIR (b6)<br>DIR (b7) | 00<br>02<br>04<br>06<br>08<br>0A<br>0C<br>0E | dd rr<br>dd rr<br>dd rr<br>dd rr<br>dd rr<br>dd rr<br>dd rr<br>dd rr | 5<br>5<br>5<br>5<br>5<br>5<br>5<br>5 |
| BSET *n opr* | Set Bit n | Mn ← 1 | — | — | — | — | — | DIR (b0)<br>DIR (b1)<br>DIR (b2)<br>DIR (b3)<br>DIR (b4)<br>DIR (b5)<br>DIR (b6)<br>DIR (b7) | 10<br>12<br>14<br>16<br>18<br>1A<br>1C<br>1E | dd<br>dd<br>dd<br>dd<br>dd<br>dd<br>dd<br>dd | 5<br>5<br>5<br>5<br>5<br>5<br>5<br>5 |
| BSR *rel* | Branch to Subroutine | PC ← (PC) + 2; push (PCL)<br>SP ← (SP) – 1; push (PCH)<br>SP ← (SP) – 1<br>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 4-6. 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*<br>CLRA<br>CLRX<br>CLR *opr*,X<br>CLR ,X | Clear Byte | M ← $00<br>A ← $00<br>X ← $00<br>M ← $00<br>M ← $00 | — | — | 0 | 1 | — | DIR<br>INH<br>INH<br>IX1<br>IX | 3F<br>4F<br>5F<br>6F<br>7F | dd<br><br><br>ff<br> | 5<br>3<br>3<br>6<br>5 |
| CMP #*opr*<br>CMP *opr*<br>CMP *opr*<br>CMP *opr*,X<br>CMP *opr*,X<br>CMP ,X | Compare Accumulator with Memory Byte | (A) – (M) | — | — | ↕ | ↕ | ↕ | IMM<br>DIR<br>EXT<br>IX2<br>IX1<br>IX | A1<br>B1<br>C1<br>D1<br>E1<br>F1 | ii<br>dd<br>hh ll<br>ee ff<br>ff<br> | 2<br>3<br>4<br>5<br>4<br>3 |
| COM *opr*<br>COMA<br>COMX<br>COM *opr*,X<br>COM ,X | Complement Byte (One's Complement) | M ← ($\overline{M}$) = $FF – (M)<br>A ← ($\overline{A}$) = $FF – (A)<br>X ← ($\overline{X}$) = $FF – (X)<br>M ← ($\overline{M}$) = $FF – (M)<br>M ← ($\overline{M}$) = $FF – (M) | — | — | ↕ | ↕ | 1 | DIR<br>INH<br>INH<br>IX1<br>IX | 33<br>43<br>53<br>63<br>73 | dd<br><br><br>ff<br> | 5<br>3<br>3<br>6<br>5 |
| CPX #*opr*<br>CPX *opr*<br>CPX *opr*<br>CPX *opr*,X<br>CPX *opr*,X<br>CPX ,X | Compare Index Register with Memory Byte | (X) – (M) | — | — | ↕ | ↕ | ↕ | IMM<br>DIR<br>EXT<br>IX2<br>IX1<br>IX | A3<br>B3<br>C3<br>D3<br>E3<br>F3 | ii<br>dd<br>hh ll<br>ee ff<br>ff<br> | 2<br>3<br>4<br>5<br>4<br>3 |
| DEC *opr*<br>DECA<br>DECX<br>DEC *opr*,X<br>DEC ,X | Decrement Byte | M ← (M) – 1<br>A ← (A) – 1<br>X ← (X) – 1<br>M ← (M) – 1<br>M ← (M) – 1 | — | — | ↕ | ↕ | — | DIR<br>INH<br>INH<br>IX1<br>IX | 3A<br>4A<br>5A<br>6A<br>7A | dd<br><br><br>ff<br> | 5<br>3<br>3<br>6<br>5 |
| EOR #*opr*<br>EOR *opr*<br>EOR *opr*<br>EOR *opr*,X<br>EOR *opr*,X<br>EOR ,X | EXCLUSIVE OR Accumulator with Memory Byte | A ← (A) ⊕ (M) | — | — | ↕ | ↕ | — | IMM<br>DIR<br>EXT<br>IX2<br>IX1<br>IX | A8<br>B8<br>C8<br>D8<br>E8<br>F8 | ii<br>dd<br>hh ll<br>ee ff<br>ff<br> | 2<br>3<br>4<br>5<br>4<br>3 |
| INC *opr*<br>INCA<br>INCX<br>INC *opr*,X<br>INC ,X | Increment Byte | M ← (M) + 1<br>A ← (A) + 1<br>X ← (X) + 1<br>M ← (M) + 1<br>M ← (M) + 1 | — | — | ↕ | ↕ | — | DIR<br>INH<br>INH<br>IX1<br>IX | 3C<br>4C<br>5C<br>6C<br>7C | dd<br><br><br>ff<br> | 5<br>3<br>3<br>6<br>5 |
| JMP *opr*<br>JMP *opr*<br>JMP *opr*,X<br>JMP *opr*,X<br>JMP ,X | Unconditional Jump | PC ← Jump Address | — | — | — | — | — | DIR<br>EXT<br>IX2<br>IX1<br>IX | BC<br>CC<br>DC<br>EC<br>FC | dd<br>hh ll<br>ee ff<br>ff<br> | 2<br>3<br>4<br>3<br>2 |
| JSR *opr*<br>JSR *opr*<br>JSR *opr*,X<br>JSR *opr*,X<br>JSR ,X | Jump to Subroutine | PC ← (PC) + n (n = 1, 2, or 3)<br>Push (PCL); SP ← (SP) – 1<br>Push (PCH); SP ← (SP) – 1<br>PC ← Effective Address | — | — | — | — | — | DIR<br>EXT<br>IX2<br>IX1<br>IX | BD<br>CD<br>DD<br>ED<br>FD | dd<br>hh ll<br>ee ff<br>ff<br> | 5<br>6<br>7<br>6<br>5 |

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

| Source Form | Operation | Description | Effect on CCR | | | | | Address Mode | Opcode | Operand | Cycles |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | H | I | N | Z | C | | | | |
| LDA #*opr*<br>LDA *opr*<br>LDA *opr*<br>LDA *opr*,X<br>LDA *opr*,X<br>LDA ,X | Load Accumulator with Memory Byte | A ← (M) | — | — | ↕ | ↕ | — | IMM<br>DIR<br>EXT<br>IX2<br>IX1<br>IX | A6<br>B6<br>C6<br>D6<br>E6<br>F6 | ii<br>dd<br>hh ll<br>ee ff<br>ff | 2<br>3<br>4<br>5<br>4<br>3 |
| LDX #*opr*<br>LDX *opr*<br>LDX *opr*<br>LDX *opr*,X<br>LDX *opr*,X<br>LDX ,X | Load Index Register with Memory Byte | X ← (M) | — | — | ↕ | ↕ | — | IMM<br>DIR<br>EXT<br>IX2<br>IX1<br>IX | AE<br>BE<br>CE<br>DE<br>EE<br>FE | ii<br>dd<br>hh ll<br>ee ff<br>ff | 2<br>3<br>4<br>5<br>4<br>3 |
| LSL *opr*<br>LSLA<br>LSLX<br>LSL *opr*,X<br>LSL ,X | Logical Shift Left (Same as ASL) |  | — | — | ↕ | ↕ | ↕ | DIR<br>INH<br>INH<br>IX1<br>IX | 38<br>48<br>58<br>68<br>78 | dd<br><br><br>ff<br> | 5<br>3<br>3<br>6<br>5 |
| LSR *opr*<br>LSRA<br>LSRX<br>LSR *opr*,X<br>LSR ,X | Logical Shift Right |  | — | — | 0 | ↕ | ↕ | DIR<br>INH<br>INH<br>IX1<br>IX | 34<br>44<br>54<br>64<br>74 | dd<br><br><br>ff<br> | 5<br>3<br>3<br>6<br>5 |
| MUL | Unsigned Multiply | X : A ← (X) × (A) | 0 | — | — | — | 0 | INH | 42 | | 1<br>1 |
| NEG *opr*<br>NEGA<br>NEGX<br>NEG *opr*,X<br>NEG ,X | Negate Byte (Two's Complement) | M ← –(M) = $00 – (M)<br>A ← –(A) = $00 – (A)<br>X ← –(X) = $00 – (X)<br>M ← –(M) = $00 – (M)<br>M ← –(M) = $00 – (M) | — | — | ↕ | ↕ | ↕ | DIR<br>INH<br>INH<br>IX1<br>IX | 30<br>40<br>50<br>60<br>70 | dd<br><br><br>ff<br> | 5<br>3<br>3<br>6<br>5 |
| NOP | No Operation | | — | — | — | — | — | INH | 9D | | 2 |
| ORA #*opr*<br>ORA *opr*<br>ORA *opr*<br>ORA *opr*,X<br>ORA *opr*,X<br>ORA ,X | Logical OR Accumulator with Memory | A ← (A) ∨ (M) | — | — | ↕ | ↕ | — | IMM<br>DIR<br>EXT<br>IX2<br>IX1<br>IX | AA<br>BA<br>CA<br>DA<br>EA<br>FA | ii<br>dd<br>hh ll<br>ee ff<br>ff | 2<br>3<br>4<br>5<br>4<br>3 |
| ROL *opr*<br>ROLA<br>ROLX<br>ROL *opr*,X<br>ROL ,X | Rotate Byte Left through Carry Bit |  | — | — | ↕ | ↕ | ↕ | DIR<br>INH<br>INH<br>IX1<br>IX | 39<br>49<br>59<br>69<br>79 | dd<br><br><br>ff<br> | 5<br>3<br>3<br>6<br>5 |
| ROR *opr*<br>RORA<br>RORX<br>ROR *opr*,X<br>ROR ,X | Rotate Byte Right through Carry Bit |  | — | — | ↕ | ↕ | ↕ | DIR<br>INH<br>INH<br>IX1<br>IX | 36<br>46<br>56<br>66<br>76 | dd<br><br><br>ff<br> | 5<br>3<br>3<br>6<br>5 |
| RSP | Reset Stack Pointer | SP ← $00FF | — | — | — | — | — | INH | 9C | | 2 |

## Table 4-6. Instruction Set Summary (Sheet 5 of 6)

| Source Form | Operation | Description | Effect on CCR | | | | | Address Mode | Opcode | Operand | Cycles |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | H | I | N | Z | C | | | | |
| RTI | Return from Interrupt | SP ← (SP) + 1; Pull (CCR)<br>SP ← (SP) + 1; Pull (A)<br>SP ← (SP) + 1; Pull (X)<br>SP ← (SP) + 1; Pull (PCH)<br>SP ← (SP) + 1; Pull (PCL) | ↕ | ↕ | ↕ | ↕ | ↕ | INH | 80 | | 9 |
| RTS | Return from Subroutine | SP ← (SP) + 1; Pull (PCH)<br>SP ← (SP) + 1; Pull (PCL) | — | — | — | — | — | INH | 81 | | 6 |
| SBC #*opr*<br>SBC *opr*<br>SBC *opr*<br>SBC *opr*,X<br>SBC *opr*,X<br>SBC ,X | Subtract Memory Byte and Carry Bit from Accumulator | A ← (A) – (M) – (C) | — | — | ↕ | ↕ | ↕ | IMM<br>DIR<br>EXT<br>IX2<br>IX1<br>IX | A2<br>B2<br>C2<br>D2<br>E2<br>F2 | ii<br>dd<br>hh ll<br>ee ff<br>ff | 2<br>3<br>4<br>5<br>4<br>3 |
| SEC | Set Carry Bit | C ← 1 | — | — | — | — | 1 | INH | 99 | | 2 |
| SEI | Set Interrupt Mask | I ← 1 | — | 1 | — | — | — | INH | 9B | | 2 |
| STA *opr*<br>STA *opr*<br>STA *opr*,X<br>STA *opr*,X<br>STA ,X | Store Accumulator in Memory | M ← (A) | — | — | ↕ | ↕ | — | DIR<br>EXT<br>IX2<br>IX1<br>IX | B7<br>C7<br>D7<br>E7<br>F7 | dd<br>hh ll<br>ee ff<br>ff | 4<br>5<br>6<br>5<br>4 |
| STOP | Stop Oscillator and Enable IRQ Pin | | — | 0 | — | — | — | INH | 8E | | 2 |
| STX *opr*<br>STX *opr*<br>STX *opr*,X<br>STX *opr*,X<br>STX ,X | Store Index Register In Memory | M ← (X) | — | — | ↕ | ↕ | — | DIR<br>EXT<br>IX2<br>IX1<br>IX | BF<br>CF<br>DF<br>EF<br>FF | dd<br>hh ll<br>ee ff<br>ff | 4<br>5<br>6<br>5<br>4 |
| SUB #*opr*<br>SUB *opr*<br>SUB *opr*<br>SUB *opr*,X<br>SUB *opr*,X<br>SUB ,X | Subtract Memory Byte from Accumulator | A ← (A) – (M) | — | — | ↕ | ↕ | ↕ | IMM<br>DIR<br>EXT<br>IX2<br>IX1<br>IX | A0<br>B0<br>C0<br>D0<br>E0<br>F0 | ii<br>dd<br>hh ll<br>ee ff<br>ff | 2<br>3<br>4<br>5<br>4<br>3 |
| SWI | Software Interrupt | PC ← (PC) + 1; Push (PCL)<br>SP ← (SP) – 1; Push (PCH)<br>SP ← (SP) – 1; Push (X)<br>SP ← (SP) – 1; Push (A)<br>SP ← (SP) – 1; Push (CCR)<br>SP ← (SP) – 1; I ← 1<br>PCH ← Interrupt Vector High Byte<br>PCL ← Interrupt Vector Low Byte | — | 1 | — | — | — | INH | 83 | | 1<br>0 |
| TAX | Transfer Accumulator to Index Register | X ← (A) | — | — | — | — | — | INH | 97 | | 2 |
| TST *opr*<br>TSTA<br>TSTX<br>TST *opr*,X<br>TST ,X | Test Memory Byte for Negative or Zero | (M) – $00 | — | — | ↕ | ↕ | — | DIR<br>INH<br>INH<br>IX1<br>IX | 3D<br>4D<br>5D<br>6D<br>7D | dd<br><br><br>ff | 4<br>3<br>3<br>5<br>4 |

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

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

## 4.7 Opcode Map

See Table 4-7.

## Table 4-7. Opcode Map

| | Bit Manipulation | | Branch | Read-Modify-Write | | | | | Control | | Register/Mem | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DIR | DIR | REL | DIR | INH | INH | IX1 | IX | INH | INH | IMM | DIR | EXT | IX |
| MSB / LSB | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D |
| **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 | SU |
| **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 | CM |
| **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 | SB |
| **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 | CP |
| **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 | AN |
| **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 | BI |
| **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 | LD |
| **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 | ST |
| **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 | EO |
| **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 | AD |
| **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 | OR |
| **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 | AD |
| **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 | JM |
| **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 | JS |
| **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 | LD |
| **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 | ST |

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 Op
Number of Cycle
Opcode Mnemo
Number of Bytes

# Chapter 5
# External Interrupt Module (IRQ)

## 5.1 Introduction

The external interrupt (IRQ) module provides asynchronous external interrupts to the CPU. The following sources can generate external interrupts:

- $\overline{IRQ}/V_{PP}$ pin
- PA0–PA3 pins

## 5.2 Features

The external interrupt module (IRQ) includes these features:

- Dedicated external interrupt pin ($\overline{IRQ}/V_{PP}$)
- Selectable interrupt on four input/output (I/O) pins (PA0–PA3)
- Programmable edge-only or edge- and level-interrupt sensitivity

## 5.3 Operation

The interrupt request/programming voltage pin ($\overline{IRQ}/V_{PP}$) and port A pins 0–3 (PA0–PA3) provide external interrupts. The PIRQ bit in the mask option register (MOR) enables PA0–PA3 as IRQ interrupt sources, which are combined into a single OR'ing function to be latched by the IRQ latch. Figure 5-1 shows the structure of the IRQ module.

After completing its current instruction, the CPU tests the IRQ latch. If the IRQ latch is set, the CPU then tests the I bit in the condition code register and the IRQE bit in the IRQ status and control register. If the I bit is clear and the IRQE bit is set, the CPU then begins the interrupt sequence. This interrupt is serviced by the interrupt service routine located at $07FA and $07FB.

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 5-3 shows the sequence of events caused by an interrupt.

### 5.3.1 $\overline{IRQ}/V_{PP}$ Pin

An interrupt signal on the $\overline{IRQ}/V_{PP}$ pin latches an external interrupt request. The LEVEL bit in the mask option register provides negative edge-sensitive triggering or both negative edge-sensitive and low level-sensitive triggering for the interrupt function.

If edge- and level-sensitive triggering is selected, a falling edge or a low level on the $\overline{IRQ}/V_{PP}$ pin latches an external interrupt request. Edge- and level-sensitive triggering allows the use of multiple wired-OR external interrupt sources. An external interrupt request is latched as long as any source is holding the $\overline{IRQ}/V_{PP}$ pin low.

If level-sensitive triggering is selected, the $\overline{IRQ}/V_{PP}$ input requires an external resistor to $V_{DD}$ for wired-OR operation. If the $\overline{IRQ}/V_{PP}$ pin is not used, it must be tied to the $V_{DD}$ supply.

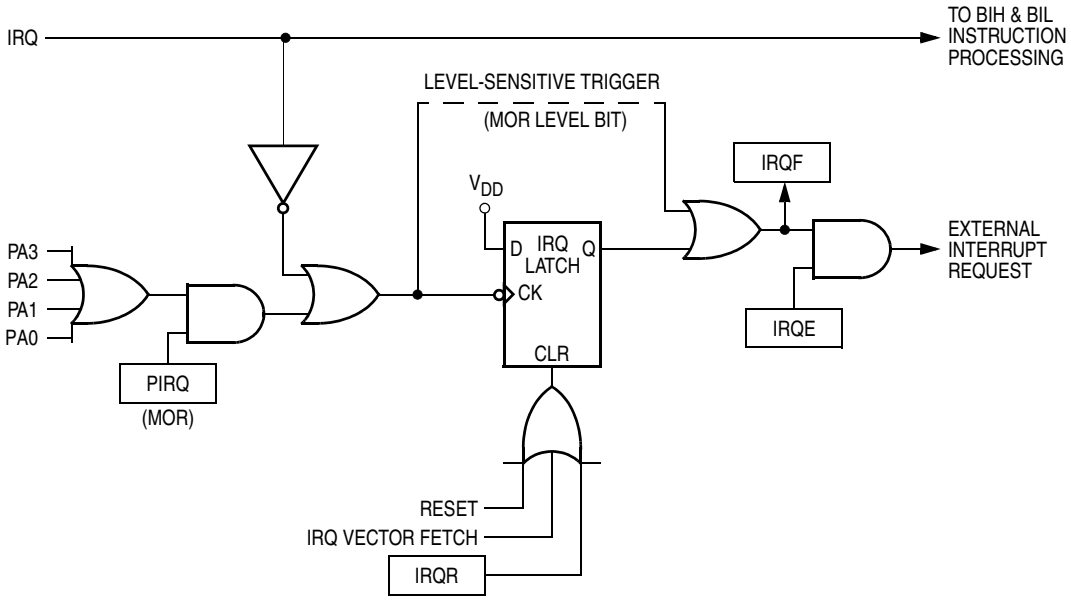



**Figure 5-1. IRQ Module Block Diagram**

| Addr. | Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $000A | IRQ Status and Control Register (ISCR) See page 54. | Read: | IRQE | 0 | 0 | 0 | IRQF | 0 | 0 | 0 |
| | | Write: | | | | R | | | IRQR | |
| | | Reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented  R = Reserved

**Figure 5-2. IRQ Module I/O Register Summary**

If edge-sensitive-only triggering is selected, a falling edge on the $\overline{IRQ}/V_{PP}$ pin latches an external interrupt request. A subsequent external 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.

The $\overline{IRQ}/V_{PP}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. The voltage on this pin can affect the mode of operation and should not exceed $V_{DD}$.

### 5.3.2 Optional External Interrupts

The inputs for the lower four bits of port A (PA0–PA3) can be connected to the $\overline{IRQ}$ pin input of the CPU if enabled by the PIRQ bit in the mask option register. This capability allows keyboard scan applications where the transitions or levels on the I/O pins will behave the same as the $\overline{IRQ}/V_{PP}$ pin except for the inverted phase (logic 1, rising edge). The active state of the $\overline{IRQ}/V_{PP}$ pin is a logic 0 (falling edge).

The PA0–PA3 pins are selected as a group to function as IRQ interrupts and are enabled by the IRQE bit in the IRQ status and control register. The PA0–PA3 pins can be positive-edge triggered only or positive-edge and high-level triggered.

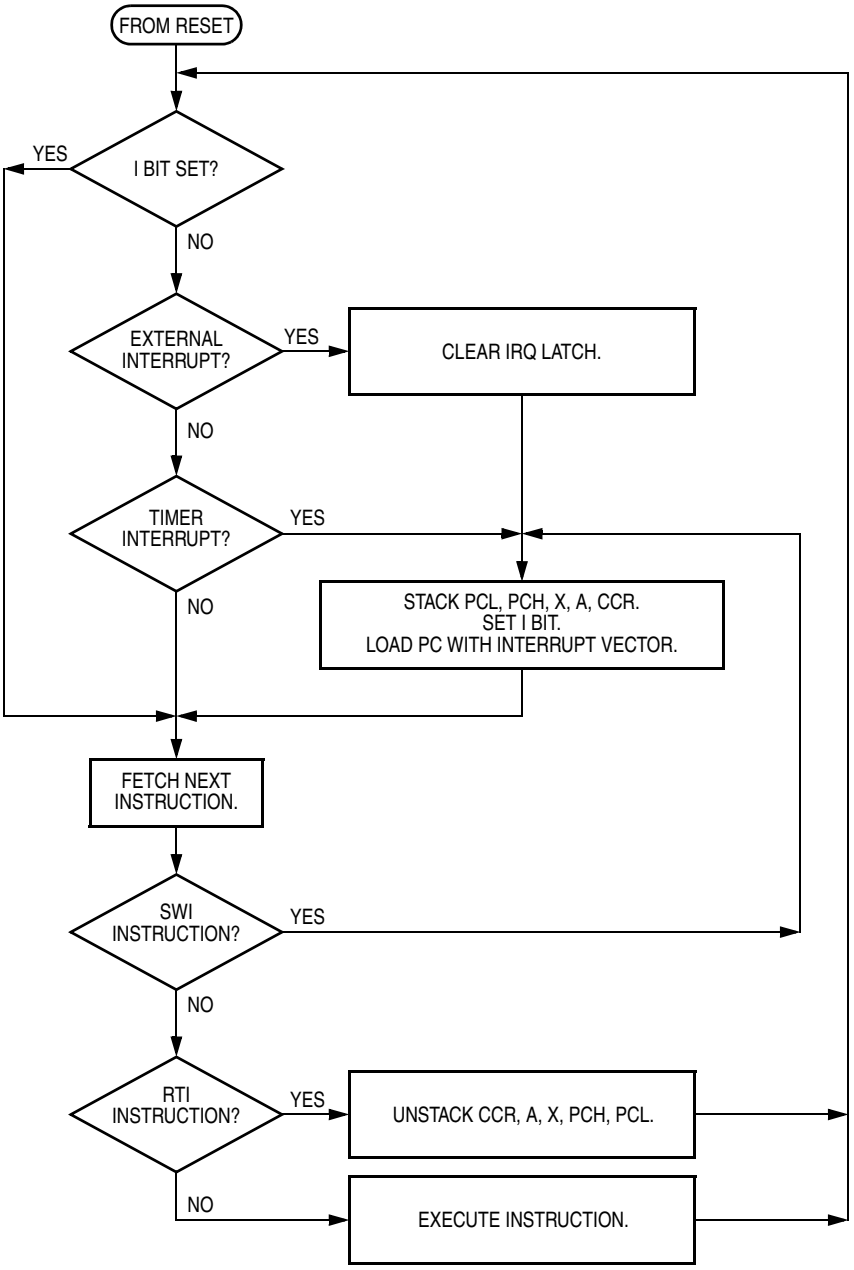



**Figure 5-3. Interrupt Flowchart**

If edge- and level-sensitive triggering is selected, a rising edge or a high level on a PA0–PA3 pin latches an external interrupt request. Edge- and level-sensitive triggering allows the use of multiple wired-OR external interrupt sources. As long as any source is holding a PA0–PA3 pin high, an external interrupt request is latched, and the CPU continues to execute the interrupt service routine.

If edge-sensitive only triggering is selected, a rising edge on a PA0–PA3 pin latches an external interrupt request. A subsequent external interrupt request can be latched only after the voltage level of the previous interrupt signal returns to logic 0 and then rises again to logic 1.

***NOTE***

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

*Enabled PA0–PA3 pins cause an IRQ interrupt regardless of whether these pins are configured as inputs or outputs.*

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

*The interrupt mask bit (I) in the condition code register (CCR) disables all maskable interrupt requests, including external interrupt requests.*

## 5.4 IRQ Status and Control Register

The IRQ status and control register (ISCR) controls and monitors operation of the IRQ module. All unused bits in the ISCR read as logic 0s. The IRQF bit is cleared and the IRQE bit is set by reset.

Address: $000A

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Read: | IRQE | 0 | 0 | 0 | IRQF | 0 | 0 | 0 |
| Write: | | | | R | | | IRQR | |
| Reset: | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented  R = Reserved

**Figure 5-4. IRQ Status and Control Register (ISCR)**

**IRQR — Interrupt Request Reset Bit**

This write-only bit clears the external interrupt request flag.

1 = Clears external interrupt and IRQF bit
0 = No effect on external interrupt and IRQF bit

**IRQF — External Interrupt Request Flag**

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

1 = External interrupt request pending
0 = No external interrupt request pending

**IRQE — External Interrupt Request Enable Bit**

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

1 = External interrupt requests enabled
0 = External interrupt requests disabled

The STOP and WAIT instructions set the IRQE bit so that an external interrupt can bring the MCU out of these low-power modes. In addition, reset sets the I bit which masks all interrupt sources.

## 5.5 Timing

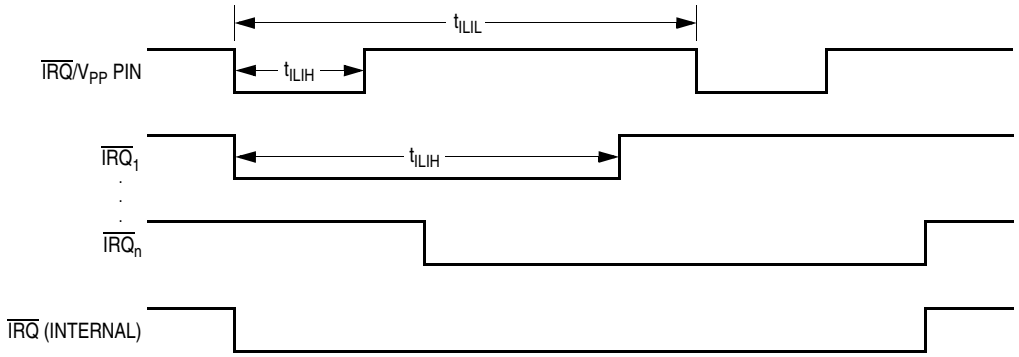



**Figure 5-5. External Interrupt Timing**

**Table 5-1. External Interrupt Timing ($V_{DD}$ = 5.0 Vdc)(1)**

| Characteristic | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| IRQ Interrupt Pulse Width Low (Edge-Triggered) | $t_{ILIH}$ | 1.5 | — | $t_{cyc}$(2) |
| IRQ Interrupt Pulse Width (Edge- and Level-Triggered) | $t_{ILIH}$ | 1.5 | Note(3) | $t_{cyc}$ |
| PA0–PA3 Interrupt Pulse Width High (Edge-Triggered) | $t_{ILIL}$ | 1.5 | — | $t_{cyc}$ |
| PA0–PA3 Interrupt Pulse Width High (Edge- and Level-Triggered) | $t_{ILIH}$ | 1.5 | Note(3) | $t_{cyc}$ |

1. $V_{DD}$ = 5.0 Vdc ± 10%, $V_{SS}$ = 0 Vdc, $T_A$ = –40°C to + 85°C, unless otherwise noted.
2. $t_{cyc} = 1/f_{OP}$; $f_{OP} = f_{OSC}/2$.
3. The minimum $t_{ILIL}$ should not be less than the number of interrupt service routine cycles plus 19 $t_{cyc}$.

**Table 5-2. External Interrupt Timing ($V_{DD}$ = 3.3 Vdc)(1)**

| Characteristic | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| IRQ Interrupt Pulse Width Low (Edge-Triggered) | $t_{ILIH}$ | 1.5 | — | $t_{cyc}$(2) |
| IRQ Interrupt Pulse Width (Edge- and Level-Triggered) | $t_{ILIH}$ | 1.5 | Note(3) | $t_{cyc}$ |
| PA0–PA3 Interrupt Pulse Width High (Edge-Triggered) | $t_{ILIL}$ | 1.5 | — | $t_{cyc}$ |
| PA0–PA3 Interrupt Pulse Width High (Edge- and Level-Triggered) | $t_{ILIH}$ | 1.5 | Note(3) | $t_{cyc}$ |

1. $V_{DD}$ = 3.3 Vdc ± 10%, $V_{SS}$ = 0 Vdc, $T_A$ = –40°C to + 85°C, unless otherwise noted.
2. $t_{cyc} = 1/f_{OP}$; $f_{OP} = f_{OSC}/2$.
3. The minimum $t_{ILIL}$ should not be less than the number of interrupt service routine cycles plus 19 $t_{cyc}$.

# Chapter 6
# Low-Power Modes

## 6.1 Introduction

The MCU can enter the following low-power standby modes:

- Stop mode — The STOP instruction puts the MCU in its lowest power-consumption mode.
- Wait mode — The WAIT instruction puts the MCU in an intermediate power-consumption mode.
- Halt mode — Halt mode is identical to wait mode, except that an oscillator stabilization delay of 1 to 4064 internal clock cycles occurs when the MCU exits halt mode. The stop-to-wait conversion bit, SWAIT, in the mask option register, enables halt mode.

  Enabling halt mode prevents the computer operating properly (COP) watchdog from being inadvertently turned off by a STOP instruction.
- Data-retention mode — In data-retention mode, the MCU retains RAM contents and CPU register contents at $V_{DD}$ voltages as low as 2.0 Vdc. The data-retention feature allows the MCU to remain in a low power-consumption state during which it retains data, but the CPU cannot execute instructions.

## 6.2 Exiting Stop and Wait Modes

The following events bring the MCU out of stop mode and load the program counter with the reset vector or with an interrupt vector:

Exiting Stop Mode

- External reset — A logic 0 on the $\overline{\text{RESET}}$ pin resets the MCU, starts the CPU clock, and loads the program counter with the contents of locations $07FE and $07FF.
- External interrupt — A high-to-low transition on the $\overline{\text{IRQ}}/V_{PP}$ pin or a low-to-high transition on an enabled port A external interrupt pin starts the CPU clock and loads the program counter with the contents of locations $07FA and $07FB.

Exiting Wait Mode

- External reset — A logic 0 on the $\overline{\text{RESET}}$ pin resets the MCU, starts the CPU clock, and loads the program counter with the contents of locations $07FE and $07FF.
- External interrupt — A high-to-low transition on the $\overline{\text{IRQ}}/V_{PP}$ pin or a low-to-high transition on an enabled port A external interrupt pin starts the CPU clock and loads the program counter with the contents of locations $07FA and $07FB.
- COP watchdog reset — A timeout of the COP watchdog resets the MCU, starts the CPU clock, and loads the program counter with the contents of locations $07FE and $07FF. Software can enable timer interrupts so that the MCU periodically can exit wait mode to reset the COP watchdog.
- Timer interrupt — Real-time interrupt requests and timer overflow interrupt requests start the MCU clock and load the program counter with the contents of locations $07F8 and $07F9.

## 6.3 Effects of Stop and Wait Modes

The STOP and WAIT instructions have the following effects on MCU modules.

### 6.3.1 Clock Generation

Effects of STOP and WAIT on clock generation are discussed here.

#### *6.3.1.1 STOP*

The STOP instruction disables the internal oscillator, stopping the CPU clock and all peripheral clocks.

After exiting stop mode, the CPU clock and all enabled peripheral clocks begin running after the oscillator stabilization delay.

***NOTE***

*The oscillator stabilization delay holds the MCU in reset for the first 4064 internal clock cycles.*

#### *6.3.1.2 WAIT*

The WAIT instruction disables the CPU clock.

After exiting wait mode, the CPU clock and all enabled peripheral clocks immediately begin running.

### 6.3.2 CPU

Effects of STOP and WAIT on the CPU are discussed here.

#### *6.3.2.1 STOP*

The STOP instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling external interrupts
- Disables the CPU clock

After exiting stop mode, the CPU clock begins running after the oscillator stabilization delay.

After exit from stop mode by external interrupt, the I bit remains clear.

After exit from stop mode by reset, the I bit is set.

#### *6.3.2.2 WAIT*

The WAIT instruction:

- Clears the interrupt mask (I bit) in the condition code register, enabling interrupts
- Disables the CPU clock

After exit from wait mode by interrupt, the I bit remains clear.

After exit from wait mode by reset, the I bit is set.

### 6.3.3 COP Watchdog

Effects of STOP and WAIT on the COP watchdog are discussed here.

#### 6.3.3.1 STOP

The STOP instruction:

- Clears the COP watchdog counter
- Disables the COP watchdog clock

**NOTE**

*To prevent the STOP instruction from disabling the COP watchdog, program the stop-to-wait conversion bit (SWAIT) in the mask option register to logic 1.*

After exit from stop mode by external interrupt, the COP watchdog counter immediately begins counting from $0000 and continues counting throughout the oscillator stabilization delay.

**NOTE**

*Immediately after exiting stop mode by external interrupt, service the COP to ensure a full COP timeout period.*

After exit from stop mode by reset:

- The COP watchdog counter immediately begins counting from $0000.
- The COP watchdog counter is cleared at the end of the oscillator stabilization delay and begins counting from $0000 again.

#### 6.3.3.2 WAIT

The WAIT instruction has no effect on the COP watchdog.

**NOTE**

*To prevent a COP timeout during wait mode, exit wait mode periodically to service the COP.*

### 6.3.4 Timer

Effects of STOP and WAIT on the timer are discussed here.

#### 6.3.4.1 STOP

The STOP instruction:

- Clears the RTIE, TOFE, RTIF, and TOF bits in the timer status and control register, disabling timer interrupt requests and removing any pending timer interrupt requests
- Disables the clock to the timer

After exiting stop mode by external interrupt, the timer immediately resumes counting from the last value before the STOP instruction and continues counting throughout the oscillator stabilization delay.

After exiting stop mode by reset and after the oscillator stabilization delay, the timer resumes operation from its reset state.

#### 6.3.4.2 WAIT

The WAIT instruction has no effect on the timer.

### 6.3.5 EPROM/OTPROM

Effects of STOP and WAIT on the EPROM/OTPROM are discussed here.

#### 6.3.5.1 STOP

The STOP instruction during EPROM programming clears the EPGM bit in the EPROM programming register, removing the programming voltage from the EPROM.

#### 6.3.5.2 WAIT

The WAIT instruction has no effect on EPROM/OTPROM operation.

## 6.4 Data-Retention Mode

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

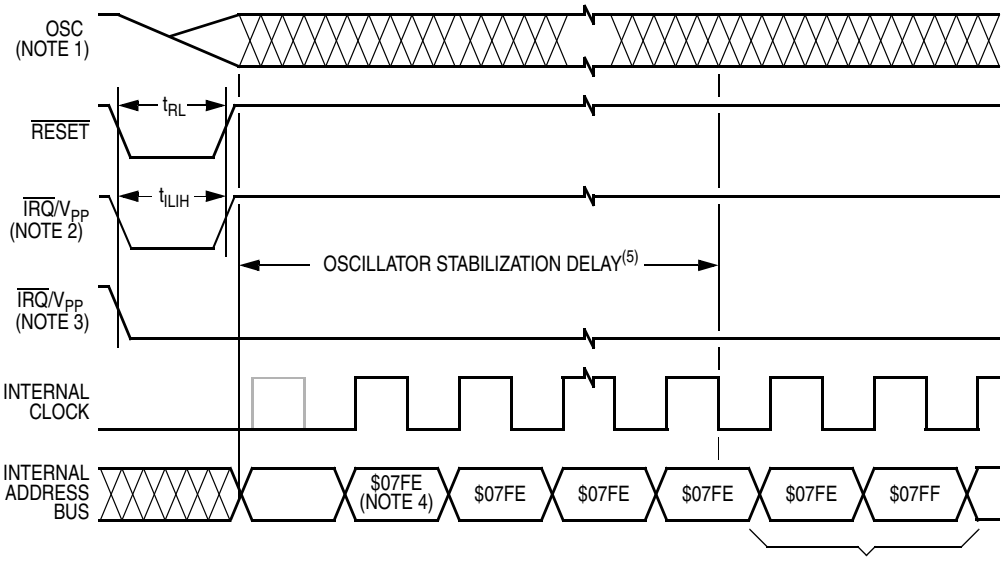
To put the MCU in data-retention mode:

1. Drive the $\overline{\text{RESET}}$ pin to 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 logic 1.

## 6.5 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
5. 4064 cycles or 128 cycles, depending on state of SOSCD bit in MOR

**Figure 6-1. Stop Mode Recovery Timing**

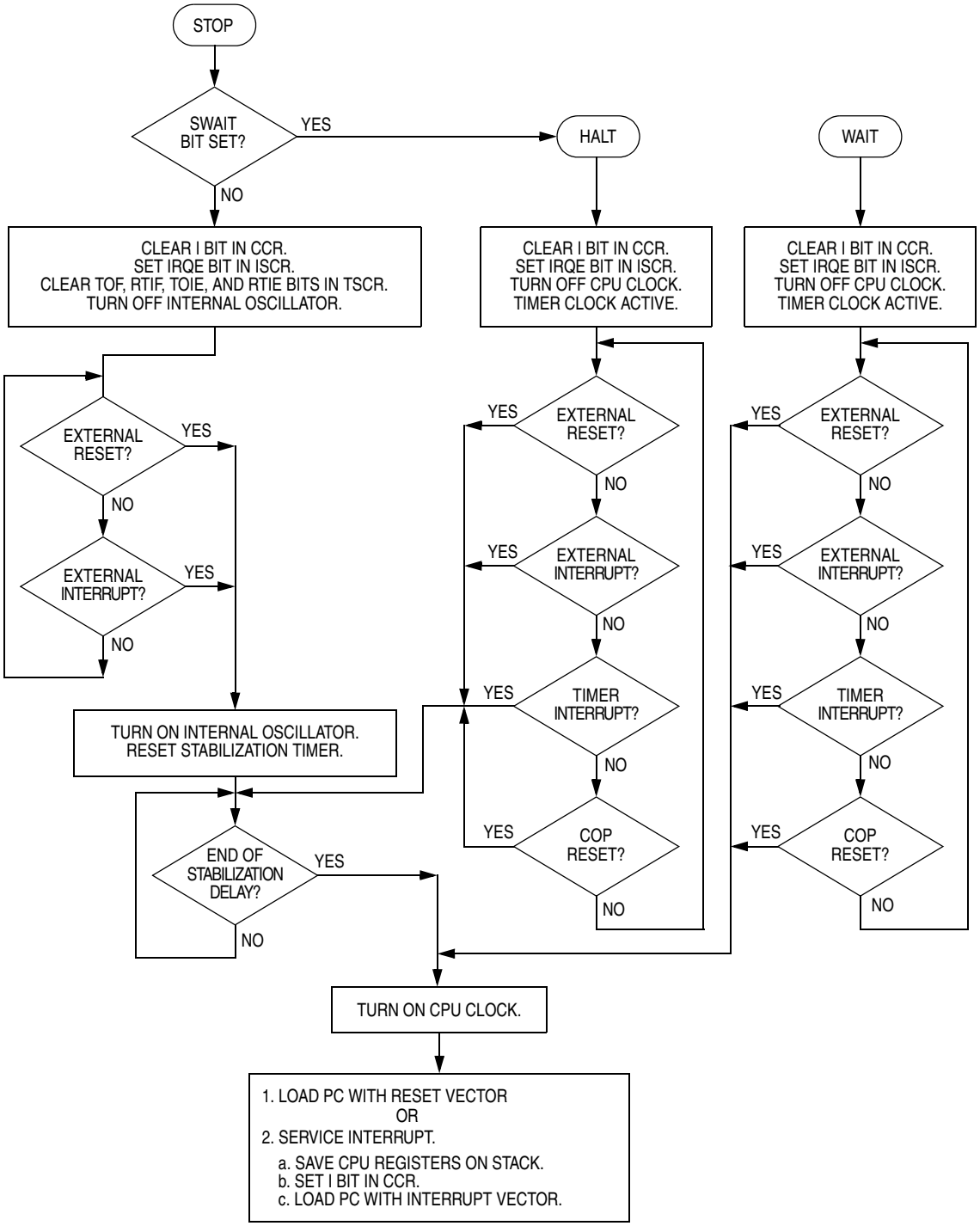



**Figure 6-2. STOP/HALT/WAIT Flowchart**

# Chapter 7
# Parallel I/O Ports (PORTS)

## 7.1 Introduction

Ten bidirectional pins form one 8-bit input/output (I/O) port and one 2-bit I/O port. All the bidirectional port pins are programmable as inputs or outputs.

***NOTE***

*Connect any unused I/O pins to an appropriate logic level, either $V_{DD}$ or $V_{SS}$. Although the I/O ports do not require termination for proper operation, termination reduces excess current consumption and the possibility of electrostatic damage.*

| 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 66. | Read: | 0 | 0 | See Note | | PB3 | PB2 | See Note | |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| $0004 | Data Direction Register A (DDRA) See page 64. | 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 67. | Read: | 0 | 0 | See Note | | DDRB3 | DDRB2 | See Note | |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $0010 | Port A Pulldown Register (PDRA) See page 65. | Read: | | | | | | | | |
| | | Write: | PDIA7 | PDIA6 | PDIA5 | PDIA4 | PDIA3 | PDIA2 | PDIA1 | PDIA0 |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $0011 | Port B Pulldown Register (PDRB) See page 68. | Read: | | | | | | | | |
| | | Write: | | | See Note | | PDIB3 | PDIB2 | See Note | |
| | | Reset: | | | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented

Note:
PB5, PB4, PB1, and PB0 should be configured as inputs at all times. These bits are available for read/write but are not available externally. Configuring them as inputs will ensure that the pulldown devices are enabled, thus properly terminating them.

**Figure 7-1. Parallel I/O Port Register Summary**

## 7.2 Port A

Port A is an 8-bit bidirectional port.

### 7.2.1 Port A Data Register

The port A data register contains a latch for each port A pin.

Address: $0000

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Read:<br>Write: | PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| Reset: | Unaffected by reset | | | | | | | |

**Figure 7-2. Port A Data Register (PORTA)**

**PA[7:0] — Port A Data Bits**

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

### 7.2.2 Data Direction Register A

Data direction register A determines whether each port A pin is an input or an output.

Address: $0004

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Read:<br>Write: | DDRA7 | DDRA6 | DDRA5 | DDRA4 | DDRA3 | DDRA2 | DDRA1 | DDRA0 |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 7-3. Data Direction Register A (DDRA)**

DDRA[7:0] — Data Direction Register A Bits

These read/write bits control port A data direction. Reset clears DDRA[7:0], configuring all port A pins as inputs.

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 data direction register A bits from 0 to 1.*

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

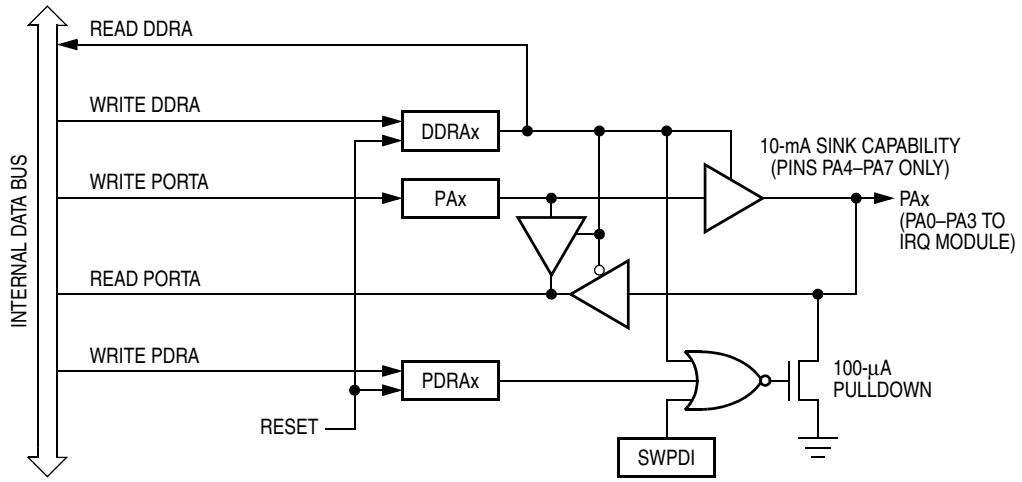



**Figure 7-4. Port A I/O Circuitry**

Writing a logic 1 to a DDRA bit enables the output buffer for the corresponding port A pin; a logic 0 disables the output buffer.

When bit DDRAx is a logic 1, reading address $0000 reads the PAx data latch. When bit DDRAx is a logic 0, reading address $0000 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 7-1 summarizes the operation of the port A pins.

**Table 7-1. Port A Pin Operation**

| Data Direction Bit | I/O Pin Mode | Accesses to Data Bit | |
|---|---|---|---|
| | | Read | Write |
| 0 | Input, high-impedance | Pin | Latch(1) |
| 1 | Output | Latch | Latch |

1. Writing affects the data register but does not affect input.

## 7.2.3 Pulldown Register A

Pulldown register A inhibits the pulldown devices on port A pins programmed as inputs.

***NOTE***

*If the SWPDI bit in the mask option register is programmed to logic 1, reset initializes all port A pins as inputs with disabled pulldown devices.*

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 7-5. Pulldown Register A (PDRA)**

**PDIA[7:0] — Pulldown Inhibit A Bits**

PDIA[7:0] disable the port A pulldown devices. Reset clears PDIA[7:0].

1 = Corresponding port A pulldown device disabled

0 = Corresponding port A pulldown device not disabled

### 7.2.4 Port LED Drive Capability

All outputs can drive light-emitting diodes (LEDs). These pins can sink approximately 10 mA of current to $V_{SS}$.

### 7.2.5 Port A I/O Pin Interrupts

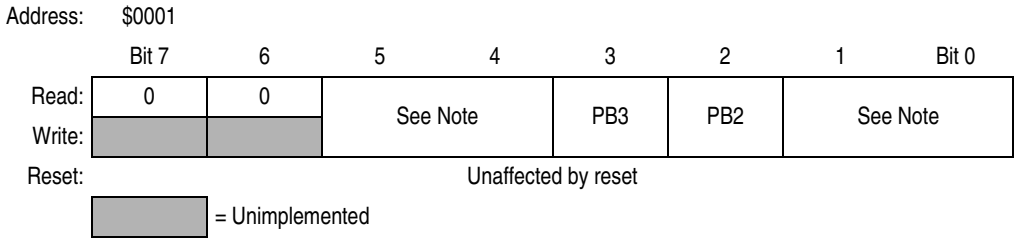
If the PIRQ bit in the mask option register is programmed to logic 1, PA0–PA3 pins function as external interrupt pins. (See Chapter 5 External Interrupt Module (IRQ).)

## 7.3 Port B

Port B is a 2-bit bidirectional port.

### 7.3.1 Port B Data Register

The port B data register contains a latch for each port B pin.




Note:
PB5, PB4, PB1, and PB0 should be configured as inputs at all times. These bits are available for read/write but are not available externally. Configuring them as inputs will ensure that the pulldown devices are enabled, thus properly terminating them.

**Figure 7-6. Port B Data Register (PORTB)**

#### PB[3:2] — Port B Data Bits

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

***NOTE***

*PB4–PB5 and PB0–PB1 should be configured as inputs at all times. These bits are available for read/write but are not available externally. Configuring them as inputs will ensure that the pulldown devices are enabled, thus properly terminating them.*

### 7.3.2 Data Direction Register B

Data direction register B determines whether each port B pin is an input or an output.

Address: $0005

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Read: | 0 | 0 | See Notes | | DDRB3 | DDRB2 | See Note | |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented

Note:
DDRB5, DDRB4, DDRB1, and DDRB0 should be configured as inputs at all times. These bits are available for read/write but are not available externally. Configuring them as inputs will ensure that the pulldown devices are enabled, thus properly terminating them.

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

**DDRB[3:2] — Data Direction Register B Bits**

These read/write bits control port B data direction. Reset clears DDRB[3:2], configuring all port B pins as inputs.

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

***NOTE***

*Avoid glitches on port B pins by writing to the port B data register before changing data direction register B bits from 0 to 1.*

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

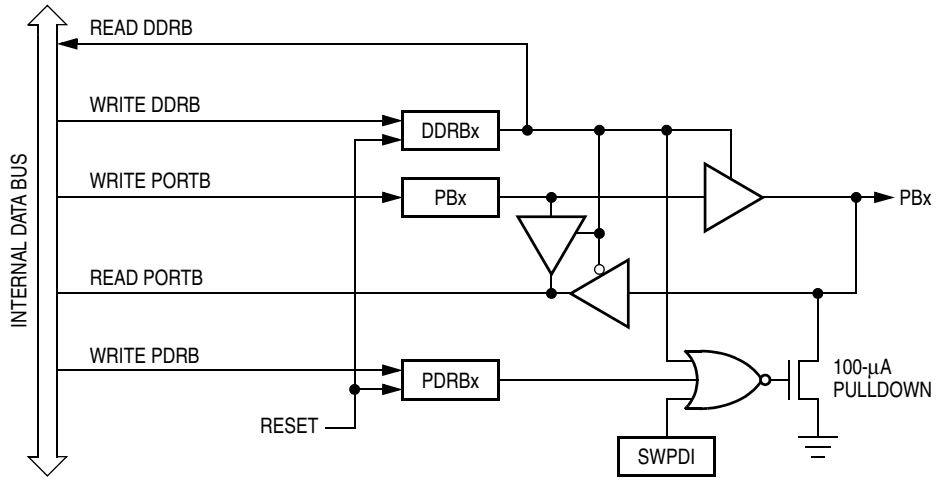



**Figure 7-8. Port B I/O Circuitry**

Writing a logic 1 to a DDRB bit enables the output buffer for the corresponding port B pin; a logic 0 disables the output buffer.

When bit DDRBx is a logic 1, reading address $0001 reads the PBx data latch. When bit DDRBx is a logic 0, reading address $0001 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 7-2 summarizes the operation of the port B pins.

**Table 7-2. Port B Pin Operation**

| Data Direction Bit | I/O Pin Mode | Accesses to Data Bit | |
|---|---|---|---|
| | | Read | Write |
| 0 | Input, high-impedance | Pin | Latch(1) |
| 1 | Output | Latch | Latch |

1. Writing affects the data register, but does not affect input.

### 7.3.3 Pulldown Register B

Pulldown register B inhibits the pulldown devices on port B pins programmed as inputs.

***NOTE***

*If the SWPDI bit in the mask option register is programmed to logic 1, reset initializes all port B pins as inputs with disabled pulldown devices.*

Address: $0011

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| Read: | | | | | | | | |
| Write: | | | See Note | | PDIB3 | PDIB2 | See Note | |
| Reset: | | | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented

Note:
These pulldown devices are permanently enabled when PB5, PB4, PB1 and PB0 are configured as inputs.

**Figure 7-9. Pulldown Register B (PDRB)**

**PDIB[3:2] — Pulldown Inhibit B Bits**

PDIB[3:2] disable the port B pulldown devices. Reset clears PDIB[3:2].

1 = Corresponding port B pulldown device disabled
0 = Corresponding port B pulldown device not disabled

## 7.4 I/O Port Electrical Characteristics

### Table 7-3. I/O Port DC Electrical Characteristics ($V_{DD}$ = 5.0 V)(1)

| Characteristic | Symbol | Min | Typ(2) | Max | Unit |
|---|---|---|---|---|---|
| Current Drain Per Pin | I | — | — | 25 | mA |
| Output High Voltage<br>($I_{Load}$ = –2.5 mA) PA4–PA7<br>($I_{Load}$ = –5.5 mA) PB2–PB3, PA0–PA3 | $V_{OH}$ | $V_{DD}$ –0.8<br>$V_{DD}$ –0.8 | —<br>— | —<br>— | V |
| Output Low Voltage<br>($I_{Load}$ = 10.0 mA) PA0–PA7, PB2–PB3 | $V_{OL}$ | — | — | 0.8 | V |
| Input High Voltage<br>PA0–PA7, PB2–PB3 | $V_{IH}$ | 0.7 x $V_{DD}$ | — | $V_{DD}$ | V |
| Input Low Voltage<br>PA0–PA7, PB2–PB3 | $V_{IL}$ | $V_{SS}$ | — | 0.2 x $V_{DD}$ | V |
| I/O Ports Hi-Z Leakage Current<br>PA0–PA7, PB2–PB3 (Without Individual Pulldown Activated) | $I_{IL}$ | — | 0.2 | ±1 | µA |
| Input Pulldown Current<br>PA0–PA7, PB2–PB3 (With Individual Pulldown Activated) | $I_{IL}$ | 35 | 80 | 200 | µA |

1. $V_{DD}$ = 5.0 Vdc ± 10%, $V_{SS}$ = 0 Vdc, $T_A$ = –40°C to +85°C, unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range, 25°C.

### Table 7-4. I/O Port DC Electrical Characteristics ($V_{DD}$ = 3.3 V)(1)

| Characteristic | Symbol | Min | Typ(2) | Max | Unit |
|---|---|---|---|---|---|
| Current Drain Per Pin | I | — | — | 25 | mA |
| Output High Voltage<br>($I_{Load}$ = –0.8 mA) PA4–PA7<br>($I_{Load}$ = –1.5 mA) PA0–PA3, PB2–PB3 | $V_{OH}$ | $V_{DD}$ –0.3<br>$V_{DD}$ –0.3 | —<br>— | —<br>— | V |
| Output Low Voltage<br>($I_{Load}$ = 5.0 mA) PA4–PA7<br>($I_{Load}$ = 3.5 mA) PA0–PA3, PB2–PB3 | $V_{OL}$ | —<br>— | —<br>— | 0.5<br>0.5 | V |
| Input High Voltage<br>PA0–PA7, PB2–PB3 | $V_{IH}$ | 0.7 x $V_{DD}$ | — | $V_{DD}$ | V |
| Input Low Voltage<br>PA0–PA7, PB2–PB3 | $V_{IL}$ | $V_{SS}$ | — | 0.2 x $V_{DD}$ | V |
| I/O Ports Hi-Z Leakage Current<br>PA0–PA7, PB2–PB3 (Without Individual Pulldown Activated) | $I_{IL}$ | — | 0.1 | ±1 | µA |
| Input Pulldown Current<br>PA0–PA7, PB2–PB3 (With Individual Pulldown Activated) | $I_{IL}$ | 12 | 30 | 100 | µA |

1. $V_{DD}$ = 3.3 Vdc ± 10%, $V_{SS}$= 0 Vdc, $T_A$ = –40°C to +85°C, unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range, 25°C.

# Chapter 8
# Resets and Interrupts

## 8.1 Introduction

Reset initializes the MCU by returning the program counter to a known address and by forcing control and status bits to known states.

Interrupts temporarily change the sequence of program execution to respond to events that occur during processing.

## 8.2 Resets

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

- Power-on reset (POR) circuit
- $\overline{\text{RESET}}$ pin
- Computer operating properly (COP) watchdog
- Illegal address

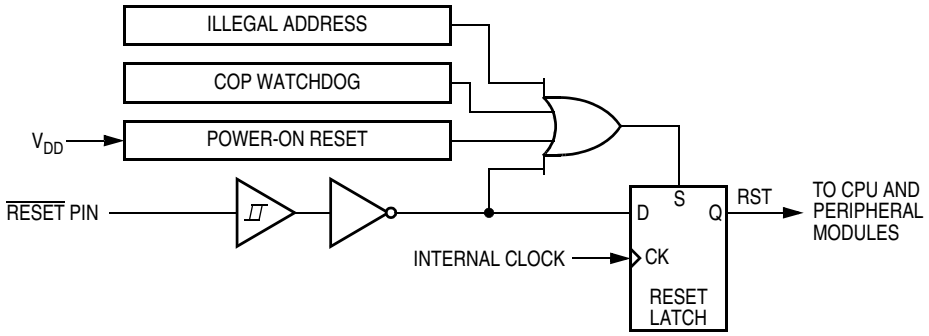



**Figure 8-1. Reset Sources**

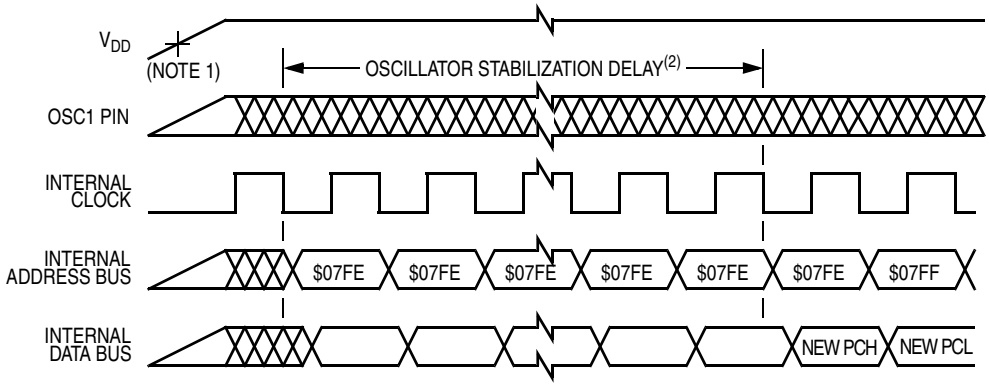
### 8.2.1 Power-On Reset

A positive transition on the $V_{DD}$ pin generates a power-on reset.

***NOTE***

*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 any reset source is active at the end of this delay, the MCU remains in the reset condition until all reset sources are inactive.

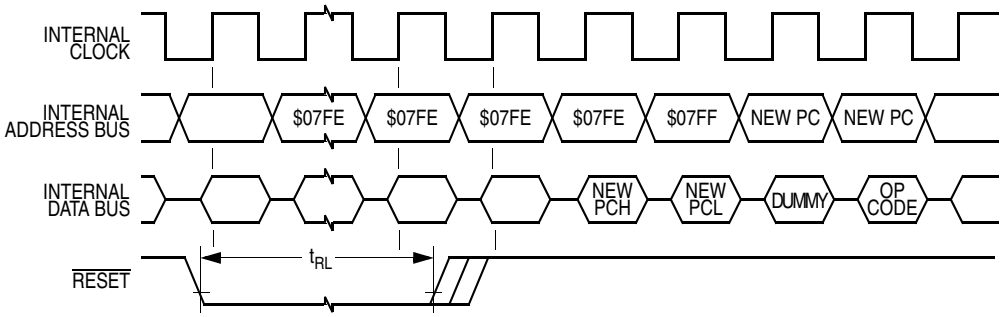



Notes:
1. Power-on reset threshold is typically between 1 V and 2 V.
2. 4064 cycles or 128 cycles, depending on state of SOSCD bit in MOR
3. Internal clock, internal address bus, and internal data bus are not available externally.

**Figure 8-2. Power-On Reset Timing**

### 8.2.2 External Reset

A logic 0 applied to the $\overline{\text{RESET}}$ pin for 1 1/2 $t_{cyc}$ generates an external reset. A Schmitt trigger senses the logic level at the $\overline{\text{RESET}}$ pin.




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 8-3. External Reset Timing**

**Table 8-1. External Reset Timing**

| Characteristic | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| $\overline{\text{RESET}}$ Pulse Width | $t_{RL}$ | 1.5 | — | $t_{cyc}$ |

### 8.2.3 COP Watchdog 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. To clear the COP watchdog and prevent a COP reset, write a logic 0 to bit 0 (COPC) of the COP register at location $07F0.

### 8.2.4 Illegal Address Reset

An opcode fetch from an address not in RAM or EPROM generates a reset.

## 8.3 Interrupts

The following sources can generate interrupts:

- SWI instruction
- External interrupt pins
  - $\overline{IRQ}/V_{PP}$ pin
  - PA0–PA3 pins
- Timer
  - Real-time interrupt flag (RTIF)
  - Timer overflow flag (TOF)

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

### 8.3.1 Software Interrupt

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

### 8.3.2 External Interrupt

An interrupt signal on the $\overline{IRQ}/V_{PP}$ pin latches an external interrupt request. When the CPU completes its current instruction, it tests the IRQ latch. If the IRQ latch is set, the CPU then tests the I bit in the condition code register. If the I bit is clear, the CPU then begins the interrupt sequence.

The CPU clears the IRQ latch during interrupt processing, so that another interrupt signal on the $\overline{IRQ}/V_{PP}$ pin can latch another interrupt request 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 8-4 shows the $\overline{IRQ}/V_{PP}$ pin interrupt logic.

Setting the I bit in the condition code register disables external interrupts.

The port A external interrupt bit (PIRQ) in the mask option register enables pins PA0–PA3 to function as external interrupt pins.

The external interrupt sensitivity bit (LEVEL) in the mask option register controls interrupt triggering sensitivity of external interrupt pins. The $\overline{IRQ}/V_{PP}$ pin can be negative-edge triggered only or negative-edge and low-level triggered. Port A external interrupt pins can be positive-edge triggered only or both positive-edge and high-level triggered. The level-sensitive triggering option allows multiple external interrupt sources to be wire-ORed to an external interrupt pin. An external interrupt request, shown in Figure 8-5, is latched as long as any source is holding an external interrupt pin low.

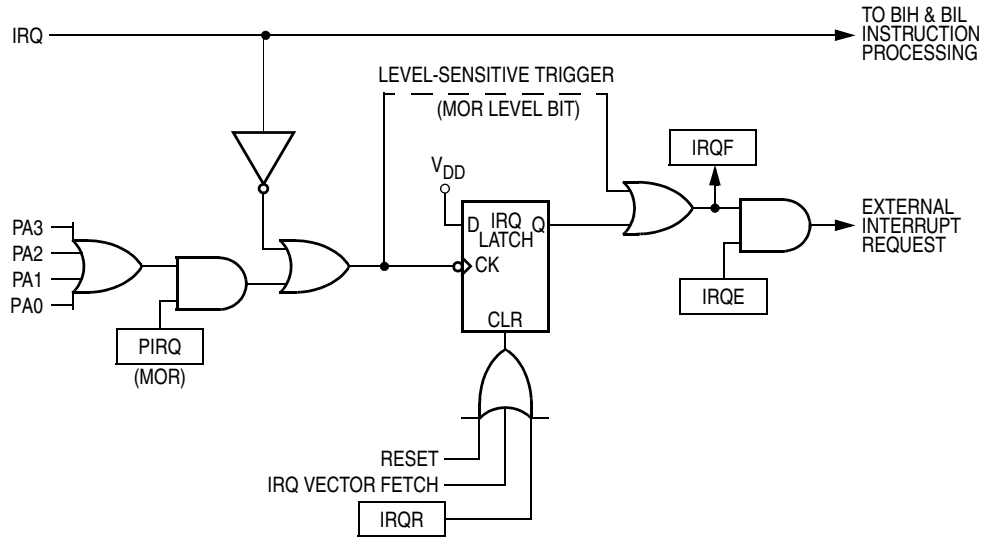



**Figure 8-4. External Interrupt Logic**

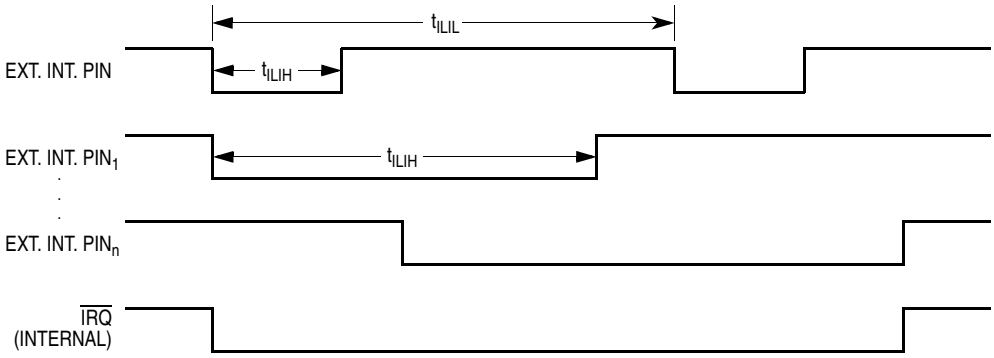



**Figure 8-5. External Interrupt Timing**

**Table 8-2. External Interrupt Timing ($V_{DD}$ = 5.0 Vdc)(1)**

| Characteristic | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| Interrupt Pulse Width Low (Edge-Triggered) | $t_{ILIH}$ | 125 | — | ns |
| Interrupt Pulse Period | $t_{ILIL}$ | Note(2) | — | $t_{cyc}$ |

1. $V_{DD}$ = 5.0 Vdc ±10%, $V_{SS}$ = 0 Vdc, $T_A$ = –40°C to +85°C, unless otherwise noted.
2. The minimum $t_{ILIL}$ should not be less than the number of interrupt service routine cycles plus 19 $t_{cyc}$.

**Table 8-3. External Interrupt Timing ($V_{DD}$ = 3.3 Vdc)(1)**

| Characteristic | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| Interrupt Pulse Width Low (Edge-Triggered) | $t_{ILIH}$ | 250 | — | ns |
| Interrupt Pulse Period | $t_{ILIL}$ | Note(2) | — | $t_{cyc}$ |

1. $V_{DD}$ = 3.3 Vdc ±10%, $V_{SS}$ = 0 Vdc, $T_A$ = –40°C to +85°C unless otherwise noted.
2. The minimum $t_{ILIL}$ should not be less than the number of interrupt service routine cycles plus 19 $t_{cyc}$.

### 8.3.3 Timer Interrupts

The timer can generate the following interrupt requests:

- Real time
- Timer overflow

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

#### *8.3.3.1 Real-Time Interrupt*

A real-time interrupt occurs if the real-time interrupt flag, RTIF, becomes set while the real-time interrupt enable bit, RTIE, is also set. RTIF and RTIE are in the timer status and control register.

#### *8.3.3.2 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. TOF and TOIE are in the timer status and control register.

### 8.3.4 Interrupt Processing

The CPU takes the following actions to begin servicing an interrupt:

- Stores the CPU registers on the stack in the order shown in Figure 8-6
- 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:
  - $07FC and $07FD (software interrupt vector)
  - $07FA and $07FB (external interrupt vector)
  - $07F8 and $07F9 (timer interrupt vector)

The return-from-interrupt (RTI) instruction causes the CPU to recover the CPU registers from the stack as shown in Figure 8-6.

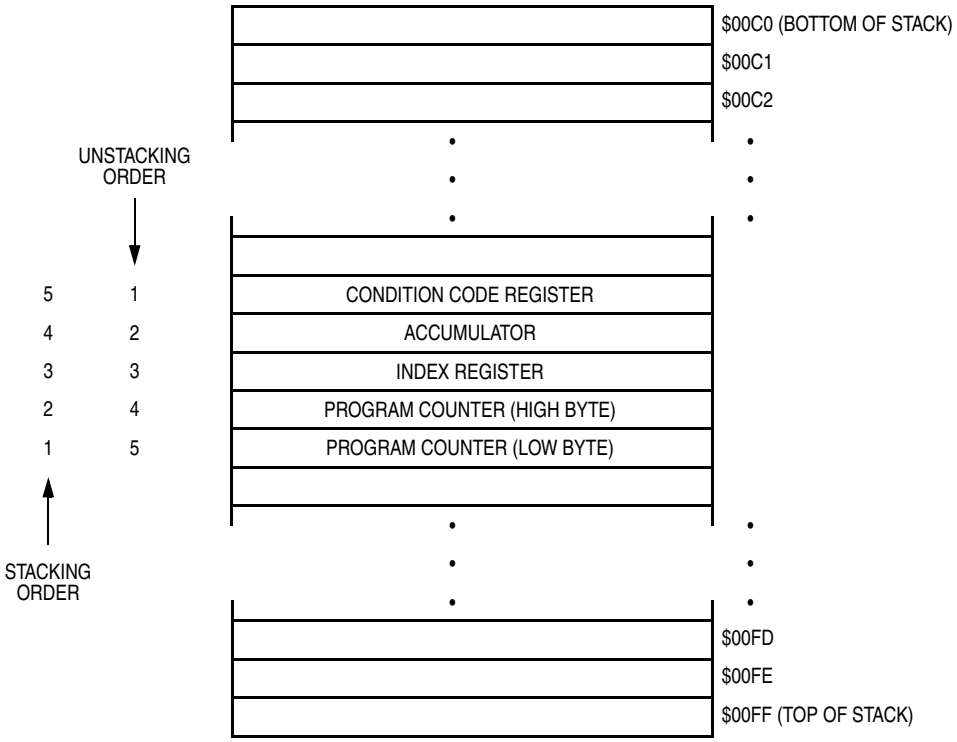



**Figure 8-6. Interrupt Stacking Order**

**Table 8-4. Reset/Interrupt Vector Addresses**

| Function | Source | Local Mask | Global Mask | Priority (1 = Highest) | Vector Address |
|---|---|---|---|---|---|
| Reset | Power-On<br>$\overline{\text{RESET}}$ Pin<br>COP Watchdog(1)<br>Illegal Address | None | None | 1 | $07FE–$07FF |
| Software Interrupt (SWI) | User Code | None | None | Same Priority as Instruction | $07FC–$07FD |
| External Interrupt | $\overline{\text{IRQ}}/V_{PP}$ Pin | IRQE | I Bit | 2 | $07FA–$07FB |
| Timer Interrupts | RTIF Bit<br>TOF Bit | RTIE Bit<br>TOIE Bit | I Bit | 3 | $07F8–$07F9 |

1. The COP watchdog is programmable in the mask option register.

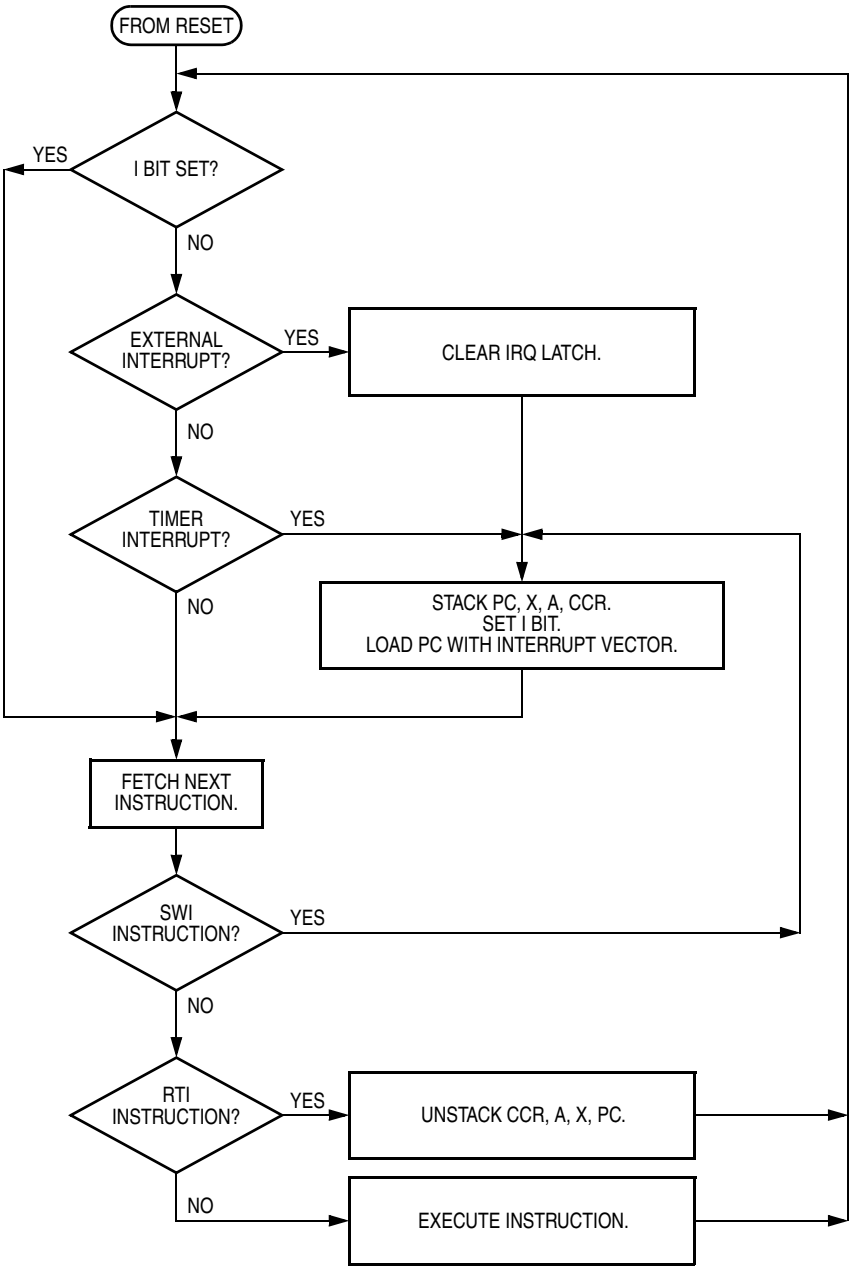



**Figure 8-7. Interrupt Flowchart**

# Chapter 9
# Multifunction Timer Module

## 9.1 Introduction

The multifunction timer provides a timing reference with programmable real-time interrupt capability. Figure 9-2 shows the timer organization.

## 9.2 Features

Features of the multifunction timer include:

- Timer overflow
- Four selectable interrupt rates
- Computer operating properly (COP) watchdog timer

## 9.3 Operation

A 15-stage ripple counter, preceded by a prescaler that divides the internal clock signal by four, provides the timing reference for the timer functions. The value of the first eight timer stages can be read at any time by accessing the timer counter register at address $0009. A timer overflow function at the eighth stage allows a timer interrupt every 1024 internal clock cycles.

The next four stages lead to the real-time interrupt (RTI) circuit. The RT1 and RT0 bits in the timer status and control register at address $0008 allow a timer interrupt every 16,384, 32,768, 65,536, or 131,072 clock cycles. The last four stages drive the selectable COP system. For information on the COP, refer to Chapter 3 Computer Operating Properly Module (COP).

| Addr. | Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $0008 | Timer Status and Control Register (TSCR) See page 81. | Read: | TOF | RTIF | TOIE | RTIE | 0 | 0 | RT1 | RT0 |
| | | Write: | | | | | TOFR | RTIFR | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| $0009 | Timer Counter Register (TCR) See page 82. | Read: | TMR7 | TMR6 | TMR5 | TMR4 | TMR3 | TMR2 | TMR1 | TMR0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | = Unimplemented | | | | | | | |

**Figure 9-1. I/O Register Summary**

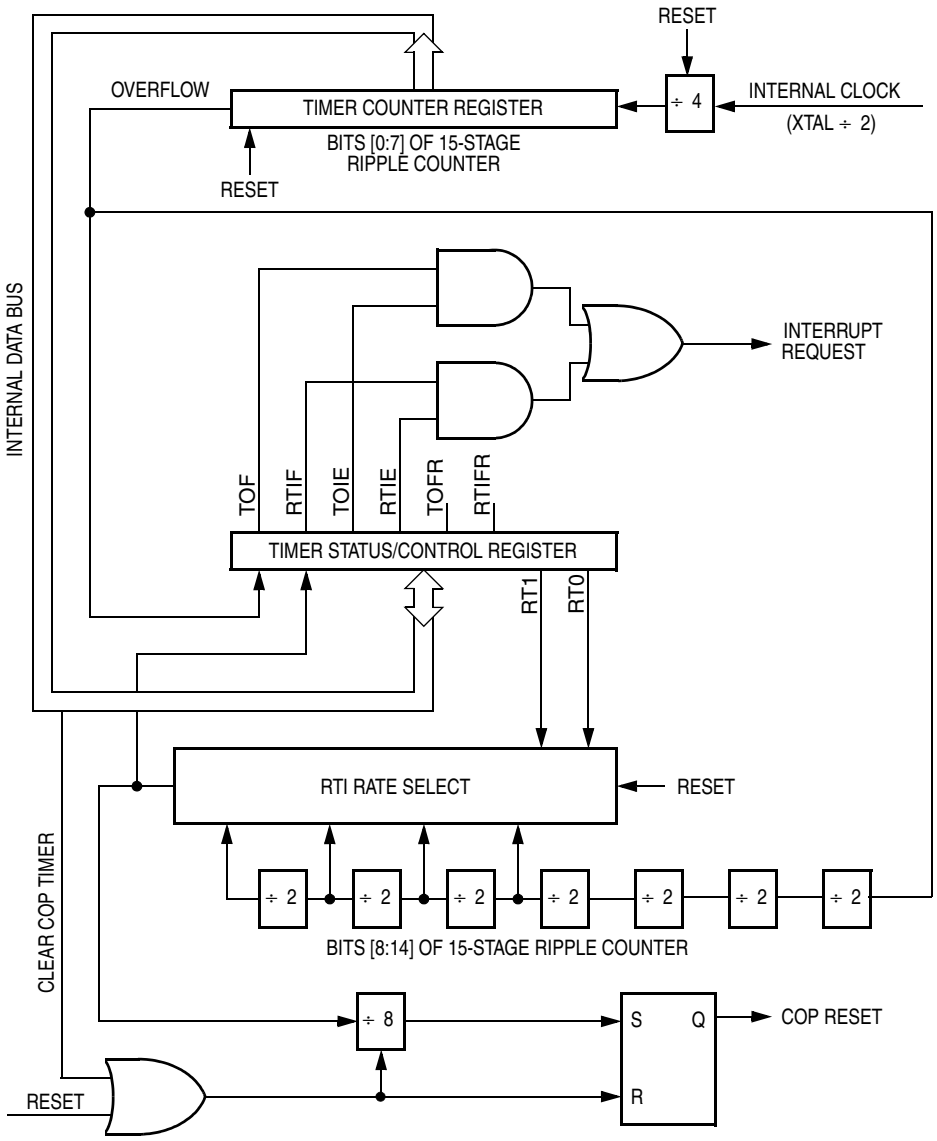



**Figure 9-2. Multifunction Timer Block Diagram**

## 9.4 Interrupts

The following timer sources can generate interrupts:

- Timer overflow flag (TOF) — The TOF bit is set when the first eight stages of the counter roll over from $FF to $00. The timer overflow interrupt enable bit, TOIE, enables TOF interrupt requests.
- Real-time interrupt flag (RTIF) — The RTIF bit is set when the selected RTI output becomes active. The real-time interrupt enable bit, RTIE, enables RTIF interrupt requests.

## 9.5 I/O Registers

The following registers control and monitor the timer operation:

- Timer status and control register (TSCR)
- Timer counter register (TCR)

### 9.5.1 Timer Status and Control Register

The read/write timer status and control register performs the following functions:

- Flags timer interrupts
- Enables timer interrupts
- Resets timer interrupt flags
- Selects real-time interrupt rates

Address: $0008

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

= Unimplemented

**Figure 9-3. 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 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 Bit**

Writing a logic 1 to this write-only bit clears the TOF bit. TOFR always reads as logic 0. Reset clears TOFR.

**RTIFR — Real-Time Interrupt Flag Reset Bit**

Writing a logic 1 to this write-only bit clears the RTIF bit. RTIFR always reads as logic 0. Reset clears RTIFR.

### RT1 and RT0 — Real-Time Interrupt Select Bits

These read/write bits select one of four real-time interrupt rates, as shown in Table 9-1. Because the selected RTI output drives the COP watchdog, changing the real-time interrupt rate also changes the counting rate of the COP watchdog. Reset sets RT1 and RT0.

***NOTE***

*Changing RT1 and RT0 when a COP timeout is imminent can cause a real-time interrupt request to be missed or an additional real-time interrupt request to be generated. To prevent this occurrence, clear the COP timer before changing RT1 and RT0.*

**Table 9-1. Real-Time Interrupt Rate Selection**

| RT1:RT0 | RTI Rate | RTI Period ($f_{OP}$ = 2 MHz) | COP Timeout Period (–0/+1 RTI Period) | Minimum COP Timeout Period ($f_{OP}$ = 2 MHz) |
|---|---|---|---|---|
| 0 0 | $f_{OP} \div 2^{14}$ | 8.2 ms | 8 x RTI Period | 65.5 ms |
| 0 1 | $f_{OP} \div 2^{15}$ | 16.4 ms | 8 x RTI Period | 131.1 ms |
| 1 0 | $f_{OP} \div 2^{16}$ | 32.8 ms | 8 x RTI Period | 262.1 ms |
| 1 1 | $f_{OP} \div 2^{17}$ | 65.5 ms | 8 x RTI Period | 524.3 ms |

## 9.5.2 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 shown in Figure 9-4.

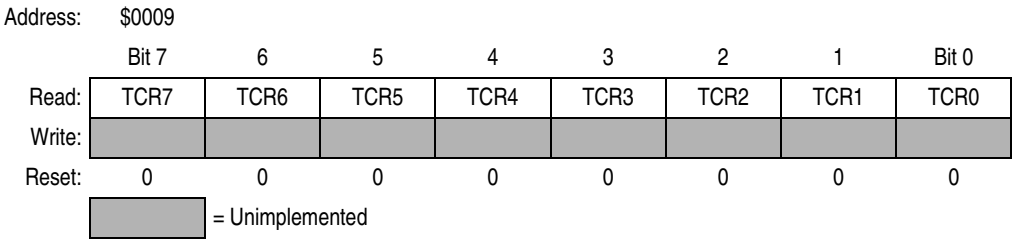



**Figure 9-4. Timer Counter Register (TCR)**

Power-on clears the entire counter chain and the internal clock begins clocking the counter. After 4064 cycles (or 16 cycles if the SOSCD bit in the mask option register is set), 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.

## 9.6 Low-Power Modes

The STOP and WAIT instructions put the MCU in low power-consumption standby states.

### 9.6.1 Stop Mode

The STOP instruction has the following effects on the timer:

- Clears the timer counter
- Clears interrupt flags (TOF and RTIF) and interrupt enable bits (TOFE and RTIE) in TSCR, removing any pending timer interrupt requests and disabling further timer interrupts

### 9.6.2 Wait Mode

The timer remains active after a WAIT instruction. Any enabled timer interrupt request can bring the MCU out of wait mode.

# Chapter 10
# Electrical Specifications

## 10.1 Maximum Ratings

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

***NOTE***

*This device is not guaranteed to operate properly at the maximum ratings. For guaranteed operating conditions, refer to 10.5 5.0-V DC Electrical Characteristics and 10.6 3.3-V DC Electrical Characteristics*

**Table 10-1. Maximum Ratings(1)**

| Rating | Symbol | Value | Unit |
|---|---|---|---|
| Supply Voltage | $V_{DD}$ | –0.3 to +7.0 | V |
| Current Drain per Pin (Excluding $V_{DD}$, $V_{SS}$) | I | 25 | mA |
| Input Voltage | $V_{In}$ | $V_{SS}$ – 0.3 to $V_{DD}$ + 0.3 | V |
| $\overline{IRQ}$/$V_{PP}$ Pin | $V_{PP}$ | $V_{SS}$ – 0.3 to 2 x $V_{DD}$ + 0.3 | V |
| Storage Temperature Range | $T_{STG}$ | –65 to +150 | °C |

1. Voltages are referenced to $V_{SS}$.

## 10.2 Operating Temperature Range

| Package Type | Symbol | Value ($T_L$ to $T_H$) | Unit |
|---|---|---|---|
| MC68HC705KJ1C(1)P(2), CDW(3), CS(4) | $T_A$ | –40 to +85 | °C |

1. C = extended temperature range
2. P = plastic dual in-line package (PDIP)
3. DW = small outline integrated circuit (SOIC)
4. S = ceramic DIP (Cerdip)

## 10.3 Thermal Characteristics

| Characteristic | Symbol | Value | Unit |
|---|---|---|---|
| Thermal Resistance<br>MC68HC705KJ1P(1)<br>MC68HC705KJ1DW(2)<br>MC68HC705KJ1S(3) | $\theta_{JA}$ | 60 | °C/W |

1. P = plastic dual in-line package (PDIP)
2. DW = small outline integrated circuit (SOIC)
3. S = ceramic DIP (Cerdip)

## 10.4 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
$\theta_{JA}$ = 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, $P_{I/O} \ll P_{INT}$ 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:

$$= P_D \text{ x } (T_A + 273°C) + \Theta J_A \text{ x } (P_D) \quad (3)$$

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 $T_A$.

## 10.5 5.0-V DC Electrical Characteristics

| Characteristic[1] | Symbol | Min | Typ[2] | Max | Unit |
|---|---|---|---|---|---|
| Output high voltage<br>($I_{Load}$ = –2.5 mA) PA4–PA7<br>($I_{Load}$ = –5.5 mA) PB2–PB3, PA0–PA3 | $V_{OH}$ | $V_{DD}$ –0.8<br>$V_{DD}$ –0.8 | —<br>— | —<br>— | V |
| Output low voltage[8]<br>($I_{Load}$ = 10.0 mA) PA0–PA7, PB2–PB3 | $V_{OL}$ | — | — | 0.8 | V |
| Input high voltage<br>PA0–PA7, PB2–PB3, $\overline{IRQ}$/$V_{PP}$, $\overline{RESET}$, OSC1 | $V_{IH}$ | 0.7 × $V_{DD}$ | — | $V_{DD}$ | V |
| Input low voltage<br>PA0–PA7, PB2–PB3, $\overline{IRQ}$/$V_{PP}$, $\overline{RESET}$, OSC1 | $V_{IL}$ | $V_{SS}$ | — | 0.2 × $V_{DD}$ | V |
| Supply current ($f_{OP}$ = 2.1 MHz; $f_{OSC}$ = 4.2 MHz)<br>Run mode[3]<br>Wait mode[4]<br>Stop mode[5] | $I_{DD}$ | <br>—<br>—<br>— | <br>4.0<br>1.0<br>0.1 | <br>6.0<br>2.8<br>5.0 | <br>mA<br>mA<br>µA |
| Supply current ($f_{OP}$ = 4.0 MHz; $f_{OSC}$ = 8.0 MHz)<br>Run mode[3]<br>Wait mode[4]<br>Stop mode[5] | $I_{DD}$ | <br>—<br>—<br>— | <br>5.2<br>1.1<br>0.1 | <br>7.0<br>3.3<br>5.0 | <br>mA<br>mA<br>µA |
| I/O Ports Hi-Z leakage current<br>PA0–PA7, PB2–PB3 (without individual pulldown activated) | $I_{IL}$ | — | 0.2 | ±1 | µA |
| Input pulldown current<br>PA0–PA7, PB2–PB3 (with individual pulldown activated) | $I_{IL}$ | 35 | 80 | 200 | µA |
| Input pullup current<br>$\overline{RESET}$ | $I_{IL}$ | –15 | –35 | –85 | µA |
| Input current[6]<br>$\overline{RESET}$, $\overline{IRQ}$/$V_{PP}$, OSC1 | $I_{In}$ | — | 0.2 | ±1 | µA |
| Capacitance<br>Ports (As Inputs or Outputs)<br>$\overline{RESET}$, $\overline{IRQ}$, OSC1, OSC2 | <br>$C_{Out}$<br>$C_{In}$ | <br>—<br>— | <br>—<br>— | <br>12<br>8 | <br>pF |
| Crystal/ceramic resonator oscillator mode internal resistor<br>OSC1 to OSC2[7] | $R_{OSC}$ | 1.0 | 2.0 | 3.0 | MΩ |

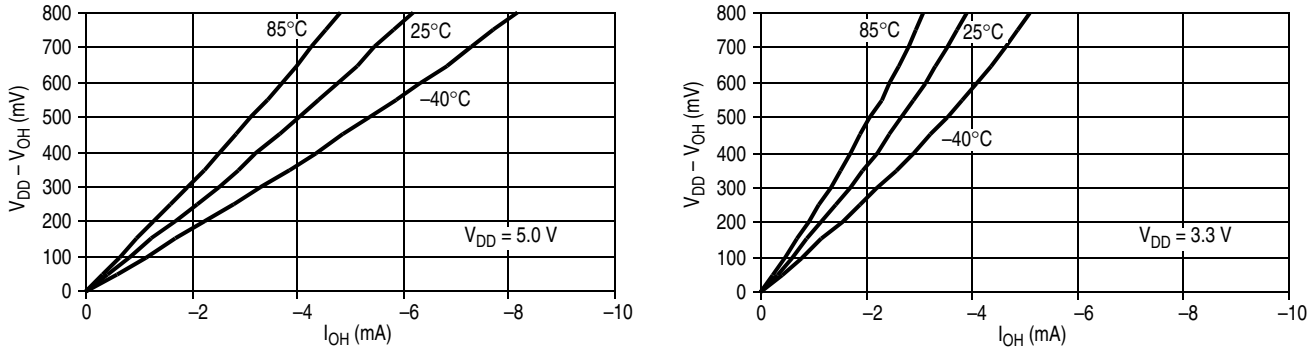
1. $V_{DD}$ = 5.0 Vdc ± 10%, $V_{SS}$ = 0 Vdc, $T_A$ = –40°C to +85°C, unless otherwise noted.
2. Typical values at midpoint of voltage range, 25°C only
3. Run mode $I_{DD}$ is measured using external square wave clock source; all inputs 0.2 V from rail; no dc loads; less than 50 pF on all outputs; $C_L$ = 20 pF on OSC2.
4. Wait mode $I_{DD}$: only timer system active. Wait mode is affected linearly by OSC2 capacitance. Wait mode is measured with all ports configured as inputs; $V_{IL}$ = 0.2 V; $V_{IH}$ = $V_{DD}$ – 0.2 V. Wait mode $I_{DD}$ is measured using external square wave clock source; all inputs 0.2 V from rail; no dc loads; less than 50 pF on all outputs; $C_L$ = 20 pF on OSC2.
5. Stop mode $I_{DD}$ is measured with OSC1 = $V_{SS}$. Stop mode $I_{DD}$ is measured with all ports configured as inputs; $V_{IL}$ = 0.2 V; $V_{IH}$ = $V_{DD}$ – 0.2 V.
6. Only input high current rated to +1 µA on $\overline{RESET}$.
7. The $R_{OSC}$ value selected for RC oscillator versions of this device is unspecified.
8. Maximum current drain for all I/O pins combined should not exceed 100 mA.

## 10.6 3.3-V DC Electrical Characteristics

| Characteristic[1] | Symbol | Min | Typ[2] | Max | Unit |
|---|---|---|---|---|---|
| Output high voltage<br>($I_{Load}$ = –0.8 mA) PA4–PA7<br>($I_{Load}$ = –1.5 mA) PA0–PA3, PB2–PB3 | $V_{OH}$ | $V_{DD}$ –0.3<br>$V_{DD}$ –0.3 | —<br>— | —<br>— | V |
| Output low voltage<br>($I_{Load}$ = 5.0 mA) PA4–PA7<br>($I_{Load}$ = 3.5 mA) PA0–PA3, PB2–PB3 | $V_{OL}$ | —<br>— | —<br>— | 0.5<br>0.5 | V |
| Input high voltage<br>PA0–PA7, PB2–PB3, $\overline{IRQ}$/$V_{PP}$, $\overline{RESET}$, OSC1 | $V_{IH}$ | 0.7 × $V_{DD}$ | — | $V_{DD}$ | V |
| Input low voltage<br>PA0–PA7, PB2–PB3, $\overline{IRQ}$/$V_{PP}$, $\overline{RESET}$, OSC1 | $V_{IL}$ | $V_{SS}$ | — | 0.2 × $V_{DD}$ | V |
| Supply current ($f_{OP}$ = 1.0 MHz; $f_{OSC}$ = 2.0 MHz)<br>Run mode[3]<br>Wait mode[4]<br>Stop mode[5] | $I_{DD}$ | <br>—<br>—<br>— | <br>1.2<br>0.3<br>0.1 | <br>2.5<br>0.8<br>5.0 | <br>mA<br>mA<br>µA |
| Supply current ($f_{OP}$ = 2.1 MHz; $f_{OSC}$ = 4.2 MHz)<br>Run mode[3]<br>Wait mode[4]<br>Stop mode[5] | $I_{DD}$ | <br>—<br>—<br>— | <br>1.4<br>0.3<br>0.1 | <br>3.0<br>1.0<br>5.0 | <br>mA<br>mA<br>µA |
| I/O ports hi-z leakage current<br>PA0–PA7, PB2–PB3 (without individual pulldown activated) | $I_{IL}$ | — | 0.1 | ±1 | µA |
| Input pulldown current<br>PA0–PA7, PB2–PB3 (with individual pulldown activated) | $I_{IL}$ | 12 | 30 | 100 | µA |
| Input pullup current<br>$\overline{RESET}$ | $I_{IL}$ | –10 | –25 | –45 | µA |
| Input current[6]<br>$\overline{RESET}$, $\overline{IRQ}$/$V_{PP}$, OSC1 | $I_{In}$ | — | 0.1 | ±1 | µA |
| Capacitance<br>Ports (as inputs or outputs)<br>$\overline{RESET}$, $\overline{IRQ}$/$V_{PP}$, OSC1, OSC2 | <br>$C_{Out}$<br>$C_{In}$ | <br>—<br>— | <br>—<br>— | <br>12<br>8 | <br>pF |
| Crystal/ceramic resonator oscillator mode internal resistor<br>OSC1 to OSC2[7] | $R_{OSC}$ | 1.0 | 2.0 | 3.0 | MΩ |

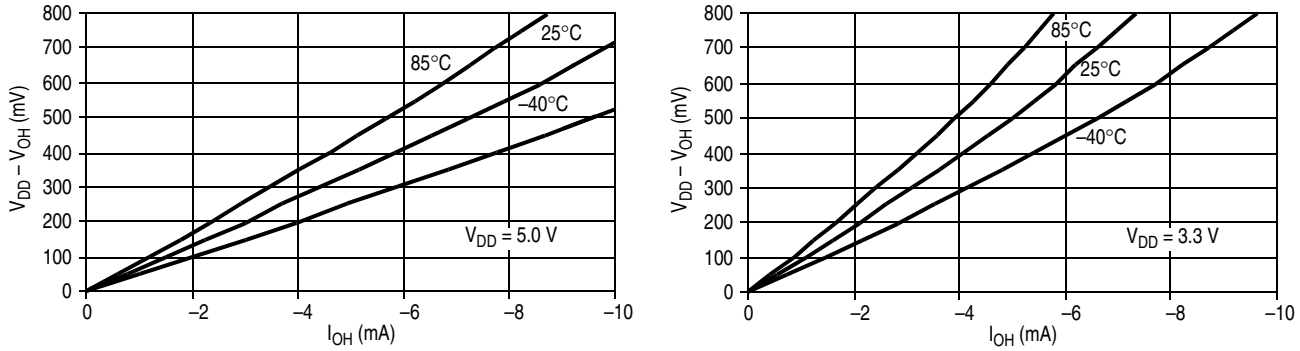
1. $V_{DD}$ = 3.3 Vdc ± 10%, $V_{SS}$ = 0 Vdc, $T_A$ = –40°C to +85°C, unless otherwise noted.
2. Typical values at midpoint of voltage range, 25°C only
3. Run mode $I_{DD}$ is measured using external square wave clock source; all inputs 0.2 V from rail; no dc loads; less than 50 pF on all outputs; $C_L$ = 20 pF on OSC2.
4. Wait mode $I_{DD}$: only timer system active. Wait mode is affected linearly by OSC2 capacitance. Wait mode is measured with all ports configured as inputs; $V_{IL}$ = 0.2 V; $V_{IH}$ = $V_{DD}$ – 0.2 V. Wait mode $I_{DD}$ is measured using external square wave clock source; all inputs 0.2 V from rail; no dc loads; less than 50 pF on all outputs; $C_L$ = 20 pF on OSC2.
5. Stop mode $I_{DD}$ is measured with OSC1 = $V_{SS}$. Stop mode $I_{DD}$ is measured with all ports configured as inputs; $V_{IL}$ = 0.2 V; $V_{IH}$ = $V_{DD}$ – 0.2 V.
6. Only input high current rated to +1 µA on $\overline{RESET}$.
7. The $R_{OSC}$ value selected for RC oscillator versions of this device is unspecified.

## 10.7 Driver Characteristics



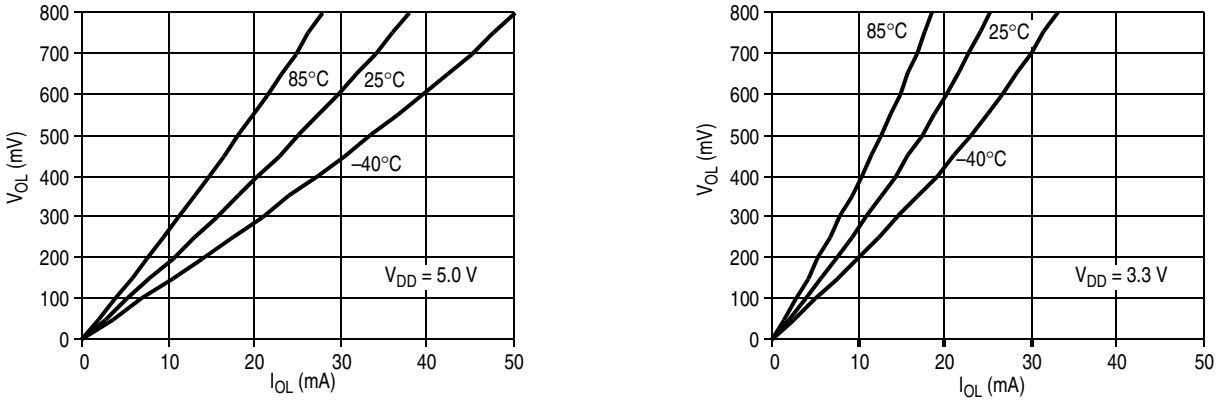

Notes:
1. At $V_{DD}$ = 5.0 V, devices are specified and tested for ($V_{DD}$ – $V_{OH}$) ≤ 800 mV @ $I_{OH}$ = –2.5 mA.
2. At $V_{DD}$ = 3.3 V, devices are specified and tested for ($V_{DD}$ – $V_{OH}$) ≤ 300 mV @ $I_{OH}$ = –0.8 mA.

**Figure 10-1. PA4–PA7 Typical High-Side Driver Characteristics**




Notes:
1. At $V_{DD}$ = 5.0 V, devices are specified and tested for ($V_{DD}$ – $V_{OH}$) ≤ 800 mV @ $I_{OH}$ = –5.5 mA.
2. At $V_{DD}$ = 3.3 V, devices are specified and tested for ($V_{DD}$ – $V_{OH}$) ≤ 300 mV @ $I_{OH}$ = –1.5 mA.

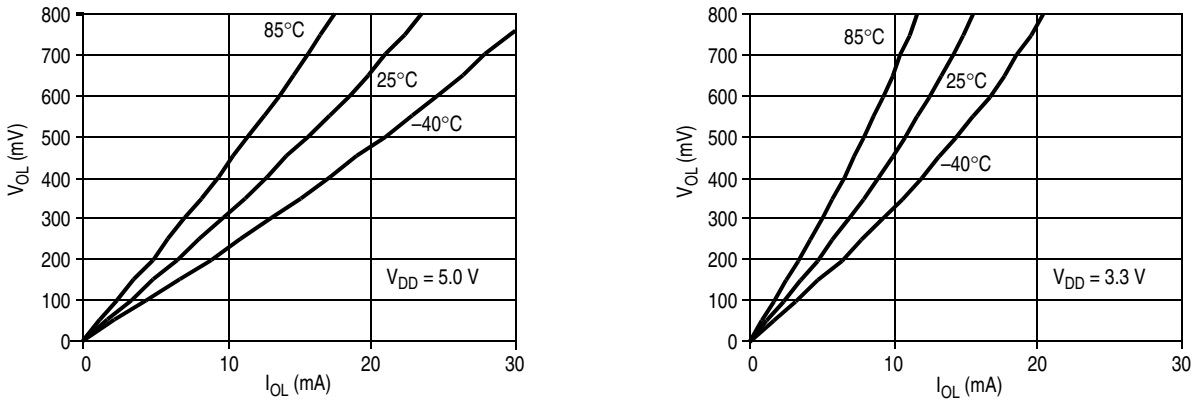
**Figure 10-2. PA0–PA3 and PB2–PB3 Typical High-Side Driver Characteristics**




Notes:

1. At $V_{DD}$ = 5.0 V, devices are specified and tested for $V_{OL} \leq 800$ mV @ $I_{OL}$ = 10.0 mA.
2. At $V_{DD}$ = 3.3 V, devices are specified and tested for $V_{OL} \leq 500$ mV @ $I_{OL}$ = 5.0 mA.

**Figure 10-3. PA4–PA7 Typical Low-Side Driver Characteristics**

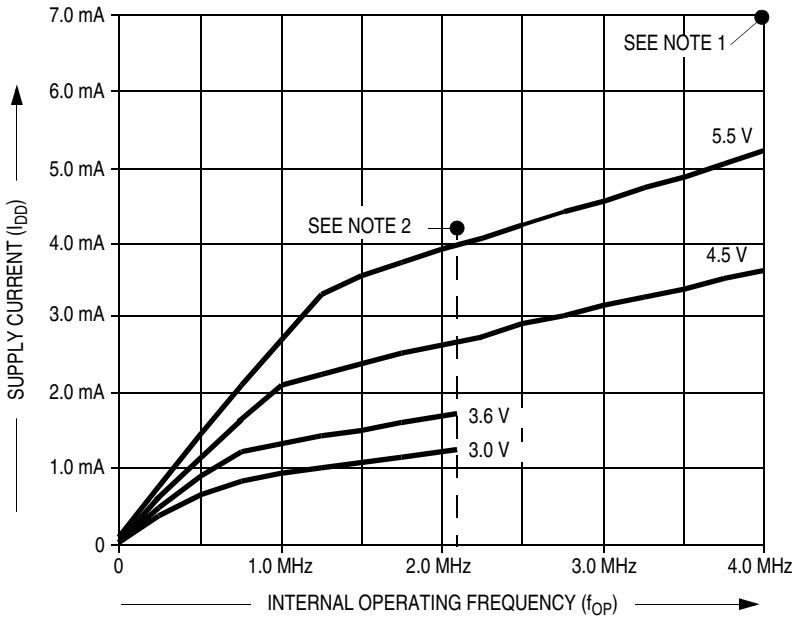



Notes:

1. At $V_{DD}$ = 5.0 V, devices are specified and tested for $V_{OL} \leq 800$ mV @ $I_{OL}$ = 10.0 mA.
2. At $V_{DD}$ = 3.3 V, devices are specified and tested for $V_{OL} \leq 500$ mV @ $I_{OL}$ = 3.5 mA.

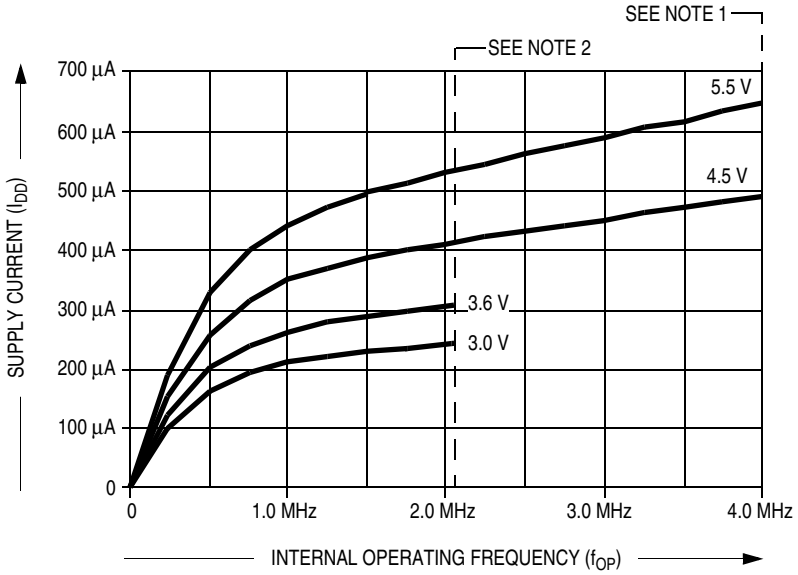
**Figure 10-4. PA0–PA3 and PB2–PB3 Typical Low-Side Driver Characteristics**

## 10.8 Typical Supply Currents




Notes:
1. At $V_{DD}$ = 5.0 V, devices are specified and tested for $I_{DD} \leq 7.0$ mA @ $f_{OP}$ = 4.0 MHz.
2. At $V_{DD}$ = 3.3 V, devices are specified and tested for $I_{DD} \leq 4.25$ mA @ $f_{OP}$ = 2.1 MHz.

**Figure 10-5. Typical Operating $I_{DD}$ (25°C)**




Notes:
1. At $V_{DD}$ = 5.0 V, devices are specified and tested for $I_{DD} \leq 3.25$ mA @ $f_{OP}$ = 4.0 MHz.
2. At $V_{DD}$ = 3.3 V, devices are specified and tested for $I_{DD} \leq 1.75$ mA @ $f_{OP}$ = 2.1 MHz.

**Figure 10-6. Typical Wait Mode $I_{DD}$ (25°C)**

## 10.9 EPROM Programming Characteristics

| Characteristic[1] | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Programming voltage<br>$\overline{IRQ}/V_{PP}$ | $V_{PP}$ | 16.0 | 16.5 | 17.0 | V |
| Programming current<br>$\overline{IRQ}/V_{PP}$ | $I_{PP}$ | —¡ | 3.0 | 10.0 | mA |
| Programming time<br>Per array byte<br>MOR | <br>$t_{EPGM}$<br>$t_{MPGM}$ | <br>4<br>4 | <br>—<br>— | <br>—<br>— | ms |

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

## 10.10 Control Timing

**Table 10-2. Control Timing ($V_{DD}$ = 5.0 Vdc)[1]**

| Characteristic | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| Oscillator frequency<br>Crystal oscillator option<br>External clock source | $f_{OSC}$ | <br>—<br>dc | <br>8.0<br>8.0 | MHz |
| Internal operating frequency ($f_{OSC}$ ÷ 2)<br>Crystal oscillator<br>External clock | $f_{OP}$ | <br>—<br>dc | <br>4.0<br>4.0 | MHz |
| Cycle time (1 ÷ $f_{OP}$) | $t_{cyc}$ | 250 | — | ns |
| $\overline{RESET}$ pulse width low | $t_{RL}$ | 1.5 | — | $t_{cyc}$ |
| $\overline{IRQ}$ interrupt pulse width low<br>(edge-triggered) | $t_{ILIH}$ | 1.5 | — | $t_{cyc}$ |
| $\overline{IRQ}$ interrupt pulse width low<br>(edge- and level-triggered) | $t_{ILIL}$ | 1.5 | Note[2] | $t_{cyc}$ |
| PA0–PA3 Interrupt pulse width high<br>(edge-triggered) | $t_{IHIL}$ | 1.5 | — | $t_{cyc}$ |
| PA0–PA3 interrupt pulse width<br>(edge- and level-triggered) | $t_{IHIH}$ | 1.5 | Note[2] | $t_{cyc}$ |
| OSC1 pulse width | $t_{OH}$, $t_{OL}$ | 100 | — | ns |

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

## Table 10-3. Control Timing ($V_{DD}$ = 3.3 Vdc)(1)

| Characteristic | Symbol | Min | Max | Unit |
|---|---|---|---|---|
| Oscillator frequency<br>Crystal oscillator option<br>External clock source | $f_{OSC}$ | —<br>dc | 4.2<br>4.2 | MHz |
| Internal operating frequency ($f_{OSC} \div 2$)<br>Crystal oscillator<br>External clock | $f_{OP}$ | —<br>dc | 2.1<br>2.1 | MHz |
| Cycle time ($1 \div f_{OP}$) | $t_{cyc}$ | 476 | — | ns |
| $\overline{\text{RESET}}$ pulse width low | $t_{RL}$ | 1.5 | — | $t_{cyc}$ |
| $\overline{\text{IRQ}}$ interrupt pulse width low (edge-triggered) | $t_{ILIH}$ | 1.5 | — | $t_{cyc}$ |
| $\overline{\text{IRQ}}$ interrupt pulse width low (edge- and level-triggered) | $t_{ILIL}$ | 1.5 | Note(2) | $t_{cyc}$ |
| PA0–PA3 interrupt pulse width high (edge-triggered) | $t_{IHIL}$ | 1.5 | — | $t_{cyc}$ |
| PA0–PA3 interrupt pulse width (edge- and level-triggered) | $t_{IHIH}$ | 1.5 | Note(2) | $t_{cyc}$ |
| OSC1 pulse width | $t_{OH}$, $t_{OL}$ | 200 | — | ns |

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

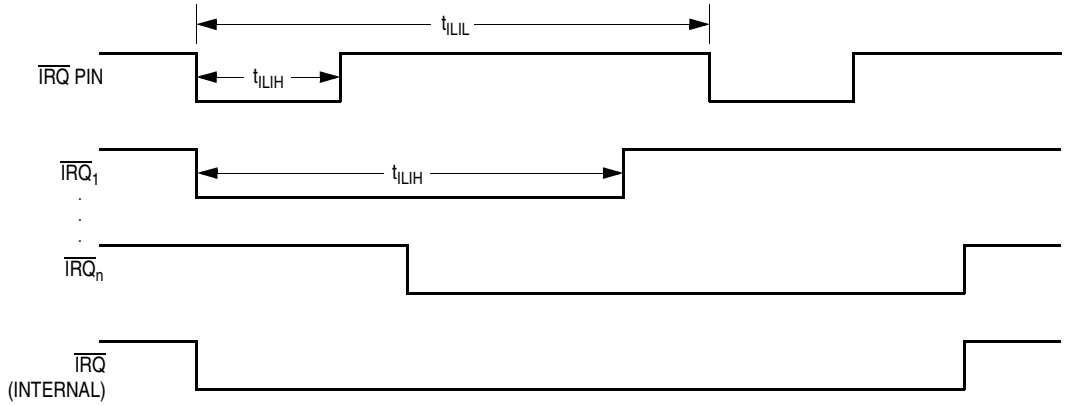
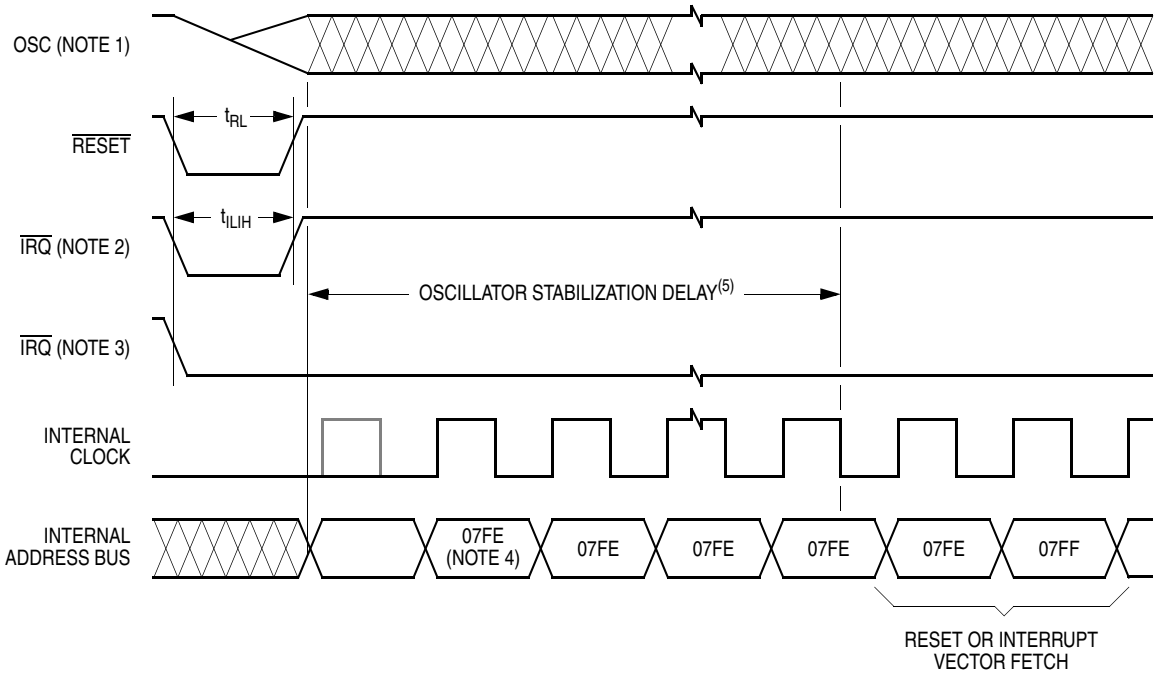


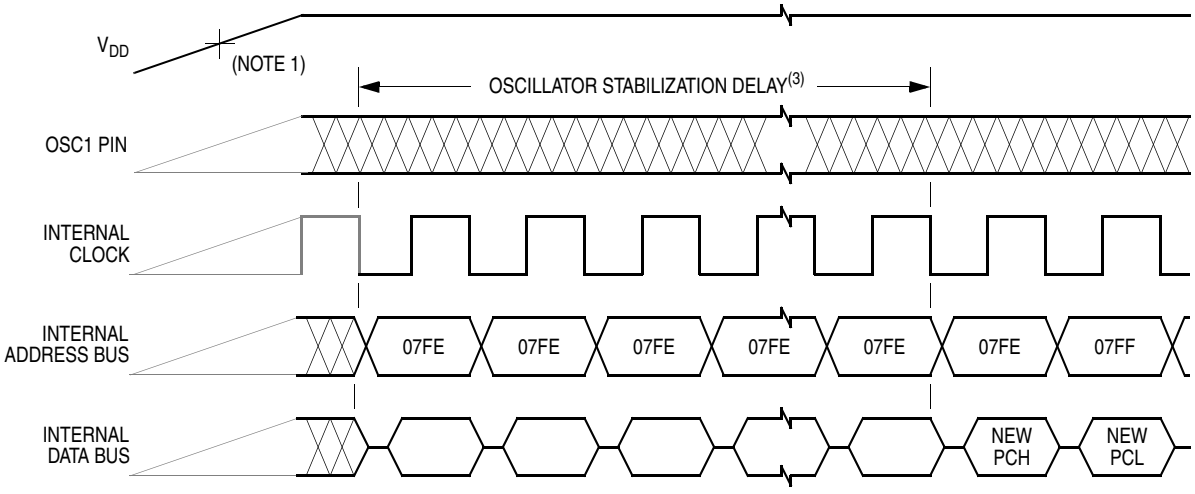

**Figure 10-7. External Interrupt Timing**




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
5. 4064 $t_{cyc}$ or 128 $t_{cyc}$, depending on the state of SOSCD bit in MOR

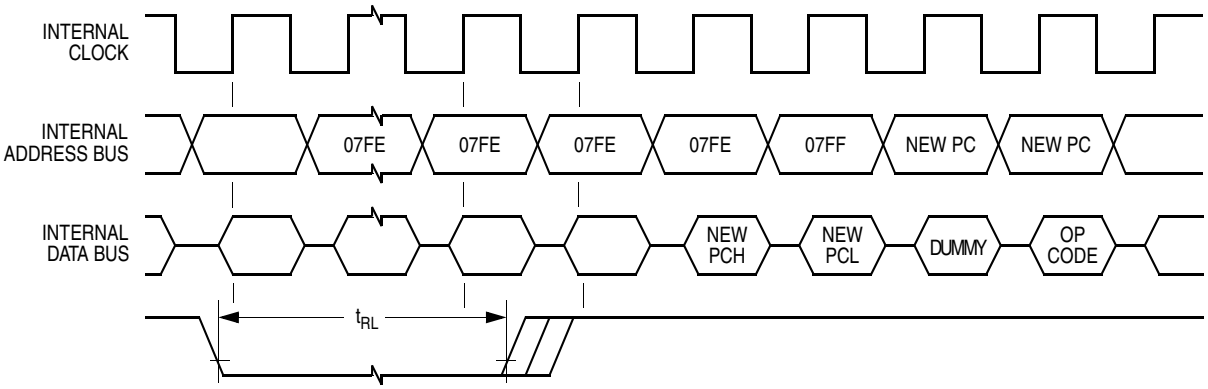
**Figure 10-8. Stop Mode Recovery Timing**




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.
3. 4064 $t_{cyc}$ or 128 $t_{cyc}$ depending on the state of SOSCD bit in MOR

**Figure 10-9. 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 10-10. External Reset Timing**

# Chapter 11
# Ordering Information and Mechanical Specifications

## 11.1 Introduction

The MC68HC705J1A, the RC oscillator, and low-speed option devices described in Appendix A MC68HRC705KJ1 and Appendix B MC68HLC705KJ1 are available in these packages:

- 648 — Plastic dual in-line package (PDIP)
- 751G — Small outline integrated circuit (SOIC)
- 620A — Ceramic DIP (Cerdip) (windowed)

This section contains ordering information and mechanical specifications for the available package types.

## 11.2 MCU Order Numbers

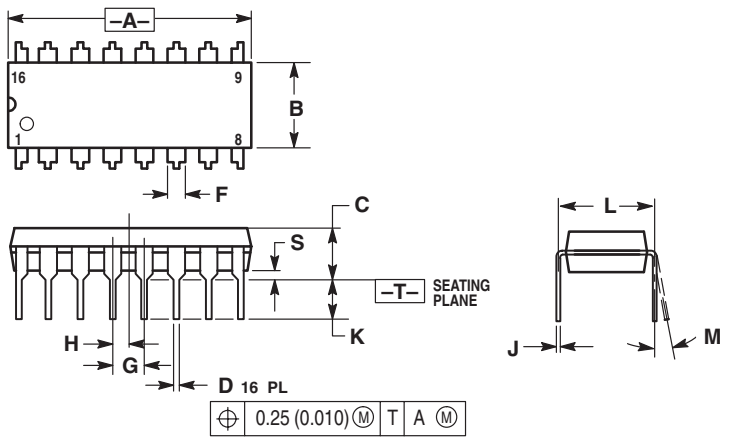
Table 11-1 lists the MC order numbers.

**Table 11-1. Order Numbers(1)**

| Package Type | Case Outline | Pin Count | Operating Temperature | Order Number |
|---|---|---|---|---|
| PDIP | 648 | 16 | –40 to +85°C | MC68HC705KJ1C(2) |
| SOIC | 751G | 16 | –40 to +85°C | MC68HC705KJ1CDW(3) |
| Cerdip | 620A | 16 | –40 to +85°C | MC68HC705KJ1CS(4) |

1. Refer to Appendix A MC68HRC705KJ1 and Appendix B MC68HLC705KJ1 for ordering information on optional low-speed and resistor-capacitor oscillator devices.
2. C = extended temperature range
3. DW = small outline integrated circuit (SOIC)
4. S = ceramic dual in-line package (Cerdip)

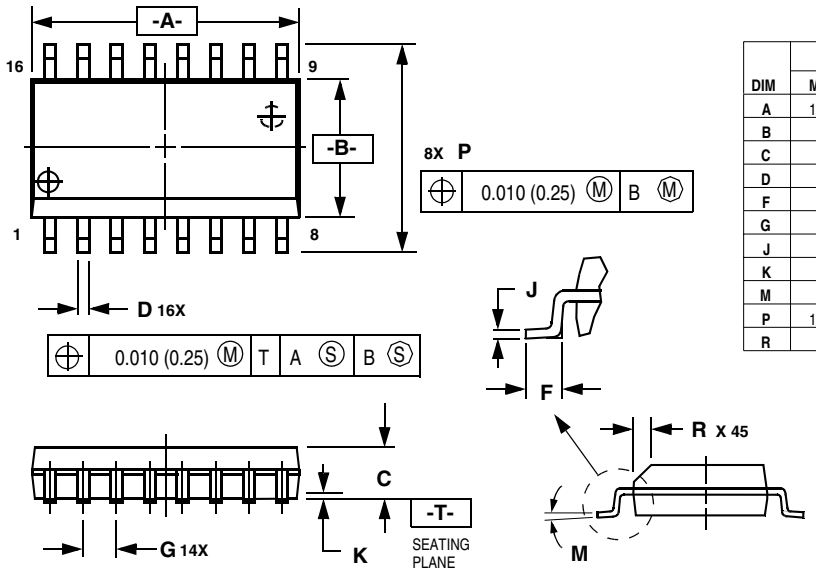
## 11.3 16-Pin PDIP — Case #648




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

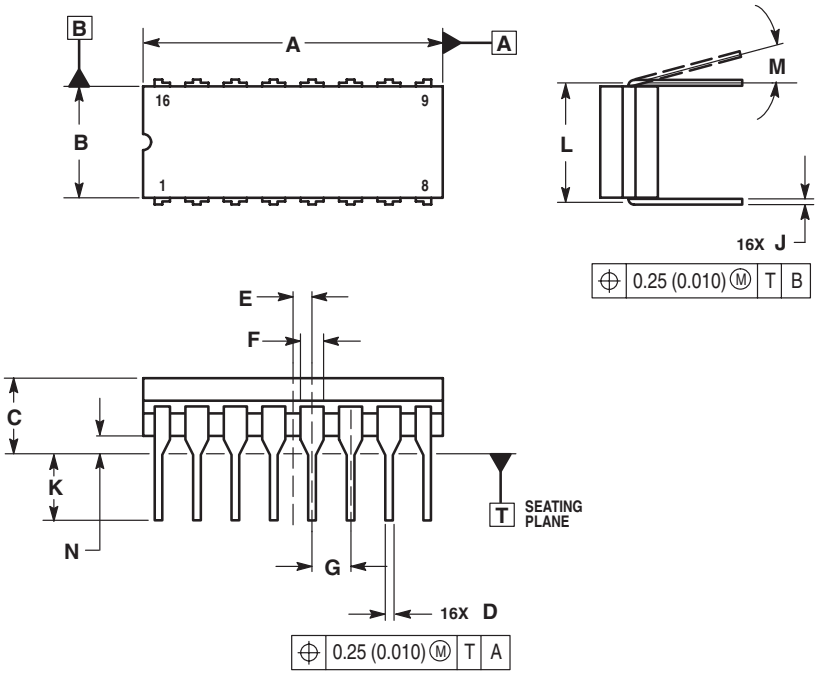
| | INCHES | | MILLIMETERS | |
|---|---|---|---|---|
| DIM | MIN | MAX | MIN | MAX |
| A | 0.740 | 0.770 | 18.80 | 19.55 |
| B | 0.250 | 0.270 | 6.35 | 6.85 |
| C | 0.145 | 0.175 | 3.69 | 4.44 |
| D | 0.015 | 0.021 | 0.39 | 0.53 |
| F | 0.040 | 0.70 | 1.02 | 1.77 |
| G | 0.100 BSC | | 2.54 BSC | |
|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

## 11.4 16-Pin SOIC — Case #751G




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

# 11.5 16-Pin Cerdip — Case #620A




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 |

# Appendix A
# MC68HRC705KJ1

## A.1 Introduction

This appendix introduces the MC68HRC705KJ1, a resistor-capacitor (RC) oscillator mask option version of the MC68HC705KJ1. All of the information in *MC68HC705KJ1 Technical Data* applies to the MC68HRC705KJ1 with the exceptions given in this appendix.

## A.2 RC Oscillator Connections

For greater cost reduction, the RC oscillator mask option allows the configuration shown in Figure A-1 to drive the on-chip oscillator. Mount the RC components as close as possible to the pins for startup stabilization and to minimize output distortion.

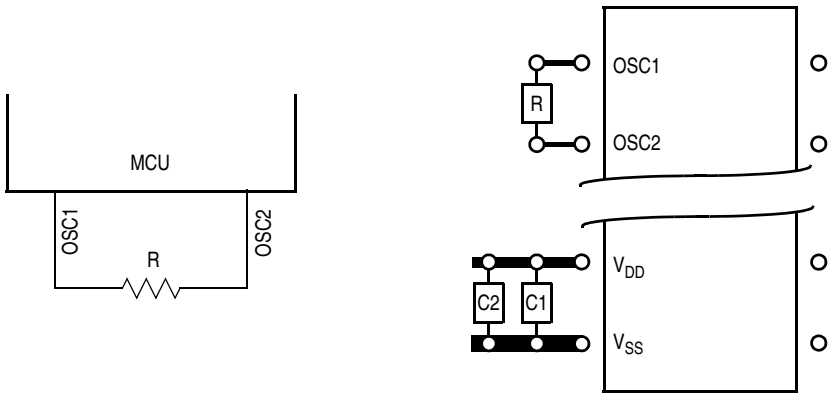



**Figure A-1. RC Oscillator Connections**

***NOTE***

*The optional internal resistor is **not** recommended for configurations that use the RC oscillator connections as shown in Figure A-1. For such configurations, the oscillator internal resistor (OSCRES) bit of the mask option register should be programmed to a logic 0.*

## A.3 Typical Internal Operating Frequency for RC Oscillator Option

Figure A-2 shows typical internal operating frequencies at 25°C for the RC oscillator option.

***NOTE***

*Tolerance for resistance is ± 50%. When selecting resistor size, consider the tolerance to ensure that the resulting oscillator frequency does not exceed the maximum operating frequency.*

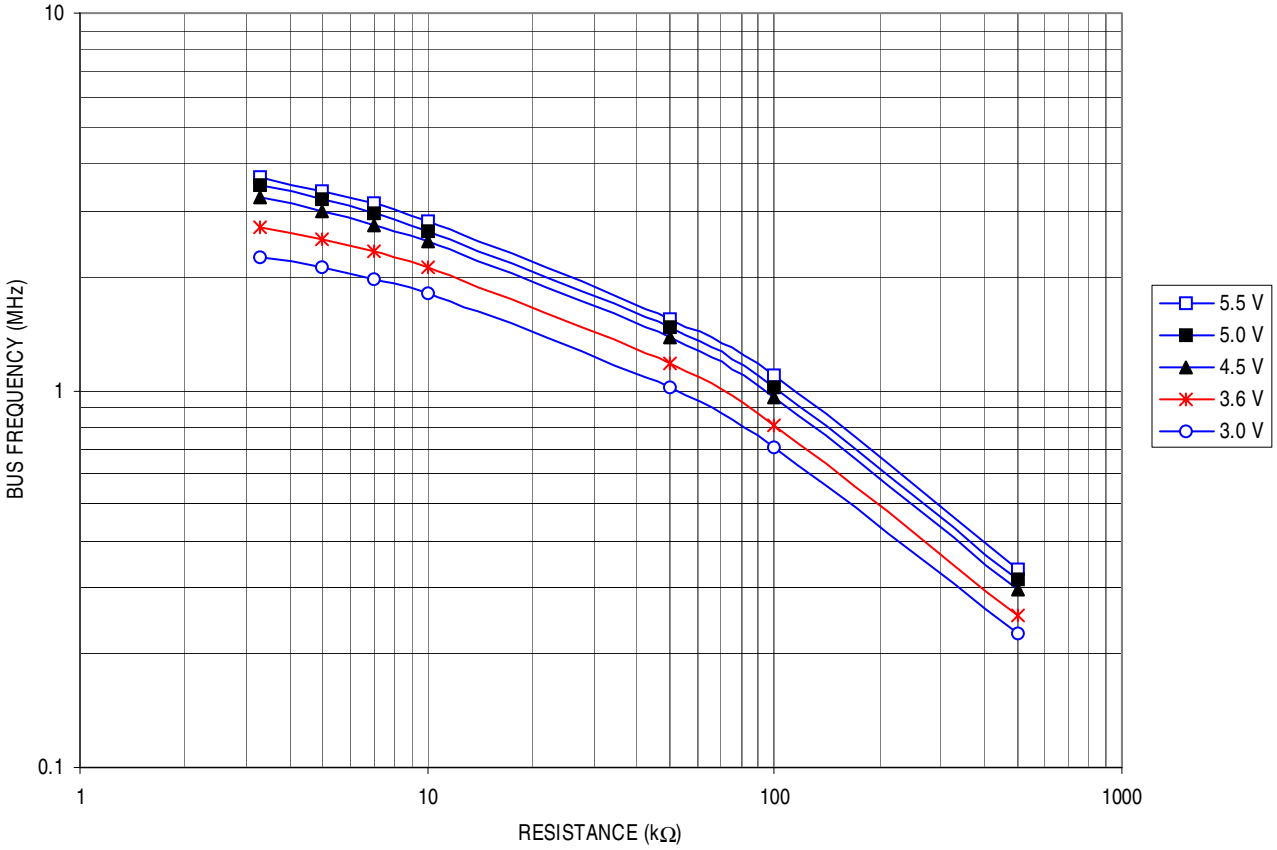



**Figure A-2. Typical Internal Operating Frequency for Various $V_{DD}$ at 25°C — RC Oscillator Option Only**

## A.4 RC Oscillator Connections (No External Resistor)

For maximum cost reduction, the RC oscillator mask connections shown in Figure A-3 allow the on-chip oscillator to be driven with **no** external components. This can be accomplished by programming the oscillator internal resistor (OSCRES) bit in the mask option register to a logic 1. When programming the OSCRES bit for the MC68HRC705KJ1, an internal resistor is selected which yields typical internal oscillator frequencies as shown in Figure A-4. The internal resistance for this device is different than the resistance of the selectable internal resistor on the MC68HC705KJ1 and the MC68HRC705KJ1 devices.

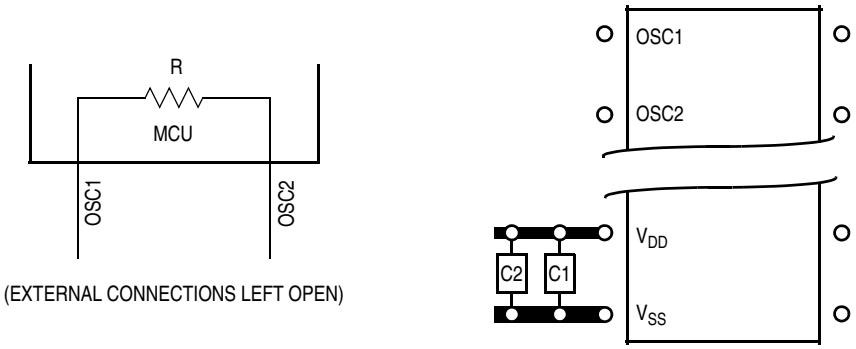



**Figure A-3. RC Oscillator Connections (No External Resistor)**

## A.5 Typical Internal Operating Frequency Versus Temperature (No External Resistor)

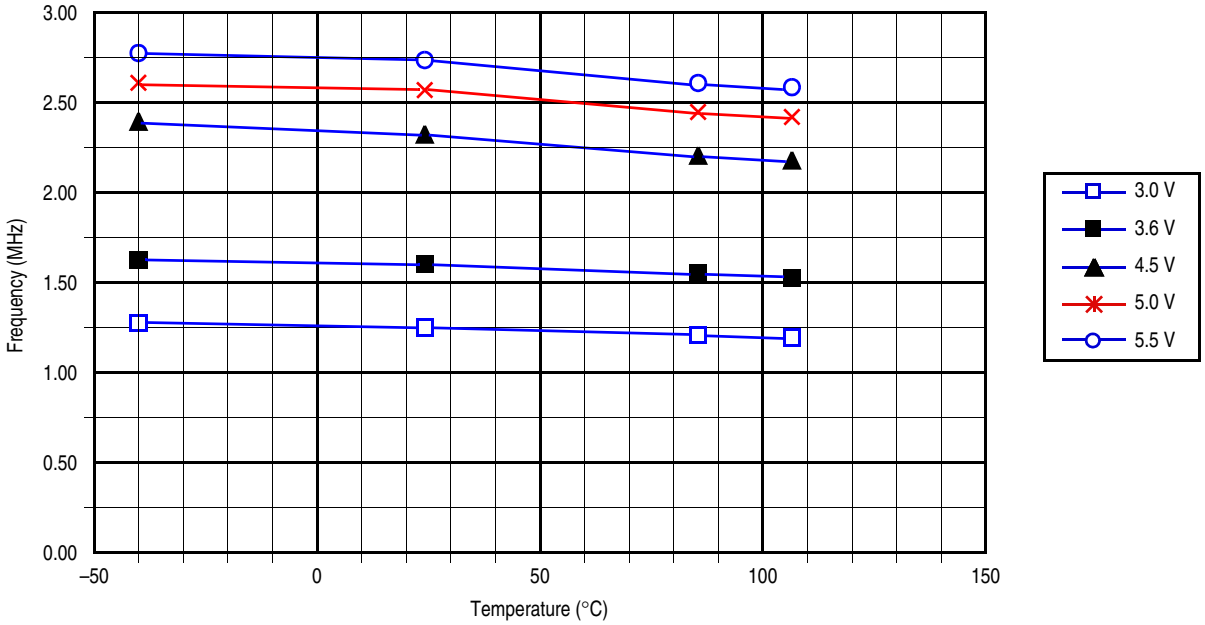



**Figure A-4. Typical Internal Operating Frequency versus Temperature (OSCRES Bit = 1)**

***NOTE***

*Due to process variations, operating voltages, and temperature requirements, the internal resistance and tolerance are unspecified. Typically for a given voltage and temperature, the frequency should not vary more than ± 500 kHz. However, this data is not guaranteed. It is the user's responsibility to ensure that the resulting internal operating frequency meets user's requirements.*

## A.6 Package Types and Order Numbers

**Table A-1. MC68HRC705KJ1 (RC Oscillator Option) Order Numbers(1)**

| Package Type | Case Outline | Pin Count | Operating Temperature | Order Number |
|---|---|---|---|---|
| PDIP | 648 | 16 | –40 to +85°C | MC68HRC705KJ1C(2)P(3) |
| SOIC | 751G | 16 | –40 to +85°C | MC68HRC705KJ1CDW(4) |
| Cerdip | 620A | 16 | –40 to +85°C | MC68HRC705KJ1CS(5) |

1. Refer to Chapter 11 Ordering Information and Mechanical Specifications for standard part ordering information.
2. C = extended temperature range
3. P = plastic dual in-line package (PDIP)
4. DW = small outline integrated circuit (SOIC)
5. S = ceramic dual in-line package (Cerdip)

# Appendix B
# MC68HLC705KJ1

## B.1 Introduction

This appendix introduces the MC68HLC705KJ1, a low-frequency version of the MC68HC705KJ1 optimized for 32-kHz oscillators. All of the information in *MC68HC705KJ1 Technical Data* applies to the MC68HLC705KJ1 with the exceptions given in this appendix.

## B.2 DC Electrical Characteristics

**Table B-1. DC Electrical Characteristics ($V_{DD}$ = 5 V)**

| Characteristic | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Supply Current ($f_{OP}$ = 16.0 kHz, $f_{OSC}$ = 32.0 kHz)<br>Run<br>Wait | $I_{DD}$ | —<br>— | 45<br>20 | 60<br>30 | µA |

**Table B-2. DC Electrical Characteristics ($V_{DD}$ = 3.3 V)**

| Characteristic | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Supply Current ($f_{OP}$ = 16.0 kHz, $f_{OSC}$ = 32.0 kHz)<br>Run<br>Wait | $I_{DD}$ | —<br>— | 25<br>10 | 35<br>15 | µA |

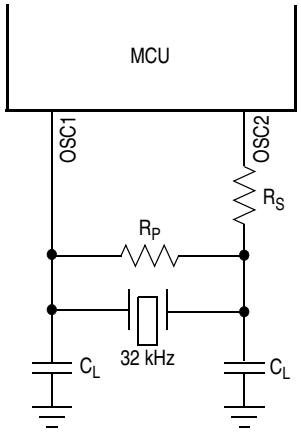



**Figure B-1. Crystal Connections**

***NOTE***

*Supply current is impacted by crystal type and external components.Since each crystal has its own characteristics, the user should consult the crystal manufacturer for appropriate values for external components.*

## B.3 Package Types and Order Numbers

**Table B-3. MC68HLC705KJ1 (Low Frequency) Order Numbers[1]**

| Package Type | Case Outline | Pin Count | Operating Temperature | Order Number |
|---|---|---|---|---|
| PDIP | 648 | 16 | –40 to +85°C | MC68HLC705KJ1C[2]P |
| SOIC | 751G | 16 | –40 to +85°C | MC68HLC705KJ1CDW[3] |
| Cerdip | 620A | 16 | –40 to +85°C | MC68HLC705KJ1CS[4] |

1. Refer to Chapter 11 Ordering Information and Mechanical Specifications for standard part ordering information.
2. C = extended temperature range
3. DW = small outline integrated circuit (SOIC)
4. S = ceramic dual in-line package (Cerdip)

## *How to Reach Us:*

**Home Page:**
www.freescale.com

**E-mail:**
support@freescale.com

**USA/Europe or Locations Not Listed:**
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
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 or +81 3 5437 9125
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
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

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

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2005. All rights reserved.

MC68HC705KJ1
Rev. 4.1, 07/2005

freescale™
semiconductor

## Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

- View MC68HC705KJ1CDW on WIN SOURCE
- Freescale Semiconductor - NXP Information

## Optimize Your Supply Chain with WIN SOURCE Solutions

- Global Sourcing Solution
- Obsolete Management
- Cost Control Management
- Shortage Management
- Alternative Solution
- Excess Inventory Management