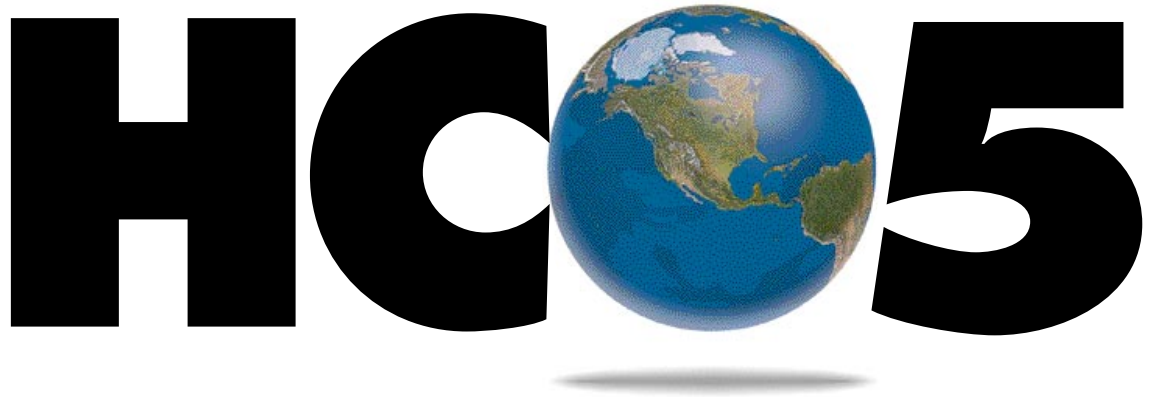


MC68HC705P9/D
REV. 3



MC68HC705P9
HCMOS Microcontroller Unit

TECHNICAL DATA



List of Sections

| | |
|---|-----|
| Table of Contents | 5 |
| Introduction | 9 |
| Pin Descriptions | 13 |
| Memory | 19 |
| Central Processor Unit (CPU) | 33 |
| Resets and Interrupts | 55 |
| Low-Power Modes | 65 |
| Parallel I/O Ports | 71 |
| Computer Operating Properly Watchdog (COP) | 85 |
| Timer | 89 |
| Serial Input/Output Port (SIOP) | 107 |
| Analog-to-Digital Converter (ADC) | 121 |
| Specifications | 131 |
| Index | 145 |
| Literature Updates | 151 |

List of Modules

All M68HC05 microcontroller units (MCUs) are customer-specified modular designs. To meet customer requirements, Motorola is constantly designing new modules and new versions of existing modules. The following table shows the version levels of the modules in the MC68HC705P9 MCU.

| Module | Version |
|--|-------------|
| Central Processor Unit (CPU) | HC05CPU |
| Timer | TIM1IC1OC_A |
| Serial Input/Output Port (SIOP) | SIOP_A |
| Computer Operating Properly Watchdog (COP) | COP0COP |
| Analog-to-Digital Converter (ADC) | ATD4X8NVRL |

Revision History

The following table summarizes differences between this revision and the previous revision of this Technical Data manual.

| | |
|--------------------------|-----------------------------------|
| Previous Revision | 2.0 |
| Current Revision | 3.0 |
| Date | 11/95 |
| Changes | Format and organizational changes |
| Location | Throughout |

Table of Contents

| | | |
|-------------------------|---|----|
| List of Sections | List of Modules | 4 |
| | Revision History | 4 |
| Introduction | Contents | 9 |
| | Features | 10 |
| | Structure | 11 |
| | Package Types and Order Numbers | 12 |
| | Programmable Options | 12 |
| Pin Descriptions | Contents | 13 |
| | Pin Assignments | 14 |
| | Pin Functions | 15 |
| Memory | Contents | 19 |
| | Features | 19 |
| | Memory Map | 20 |
| | Input/Output Register Summary | 21 |
| | RAM | 24 |
| | EPROM/OTPROM | 25 |
| | Mask Option Register | 31 |
| CPU | Contents | 33 |
| | Features | 34 |
| | Introduction | 34 |

Table of Contents

| | | |
|----------------------------------|----------------------------------|----|
| | CPU Control Unit | 36 |
| | Arithmetic/Logic Unit | 36 |
| | CPU Registers | 36 |
| | Instruction Set | 40 |
| Resets and Interrupts | Contents | 55 |
| | Resets | 56 |
| | Low-Voltage Protection | 58 |
| | Interrupts | 59 |
| Low-Power Modes | Contents | 65 |
| | Stop Mode | 65 |
| | Wait Mode | 68 |
| | Data-Retention Mode | 70 |
| Parallel I/O Ports | Contents | 71 |
| | Introduction | 72 |
| | Port A | 73 |
| | Port B | 76 |
| | Port C | 79 |
| | Port D | 82 |
| COP | Contents | 85 |
| | Features | 85 |
| | Introduction | 86 |
| | Operation | 86 |
| | Interrupts | 87 |
| | COP Register | 87 |
| | Low-Power Modes | 88 |

| | | |
|--------------|---|-----|
| Timer | Contents | 89 |
| | Features | 90 |
| | Introduction | 90 |
| | Operation. | 93 |
| | Timing | 95 |
| | Interrupts | 98 |
| | I/O Registers | 98 |
| | Low-Power Modes. | 106 |
| | | |
| SIOP | Contents | 107 |
| | Features | 108 |
| | Introduction | 108 |
| | Operation. | 110 |
| | Timing | 114 |
| | Interrupts | 115 |
| | I/O Registers | 116 |
| | Low-Power Modes. | 120 |
| | | |
| ADC | Contents | 121 |
| | Features | 121 |
| | Introduction | 122 |
| | Operation. | 123 |
| | Interrupts | 124 |
| | Timing and Electrical Characteristics | 125 |
| | I/O Registers | 126 |
| | Low-Power Modes. | 130 |

Table of Contents

| | | |
|---------------------------|---|-------|
| Specifications | Contents | 131 |
| | Maximum Ratings | 132 |
| | Operating Temperature Range | 133 |
| | Thermal Characteristics | 133 |
| | Power Considerations | 134 |
| | 5.0 V DC Electrical Characteristics | 135 |
| | 3.3 V DC Electrical Characteristics | 136 |
| | Driver Characteristics | 137 |
| | Typical Supply Current vs. Internal Clock Frequency | 138 |
| | Maximum Supply Current vs. Internal Clock Frequency | 139 |
| | 5.0 V Control Timing | 140 |
| | 3.3 V Control Timing | 141 |
| | Test Load | 142 |
| | Mechanical Specifications | 142 |
| | Index | |
| Literature Updates | Literature Distribution Centers | 151 |
| | Mfax | 152 |
| | Motorola SPS World Marketing World Wide Web Server | 152 |
| | CSIC Microcontroller Division's Web Site | 152 |

Contents

| | |
|---------------------------------------|----|
| Features | 10 |
| Structure | 11 |
| Package Types and Order Numbers | 12 |
| Programmable Options | 12 |

Features

- Four Peripheral Modules
 - 16-Bit Input Capture/Output Compare Timer
 - Synchronous Serial I/O Port (SIOP)
 - 4-Channel, 8-Bit Analog-to-Digital Converter (ADC)
 - Computer Operating Properly (COP) Watchdog
- 20 Bidirectional I/O Port Pins and One Input-Only Port Pin
- On-Chip Oscillator with Connections for:
 - Crystal
 - Ceramic Resonator
 - External Clock
- 2104 Bytes of EPROM/OTPROM
 - 48 Bytes of Page Zero EPROM/OTPROM
 - Eight Locations for User Vectors
- 128 Bytes of User RAM
- Bootloader ROM
- Memory-Mapped Input/Output (I/O) Registers
- Fully Static Operation with No Minimum Clock Speed
- Power-Saving Stop, Wait, and Data-Retention Modes

Structure

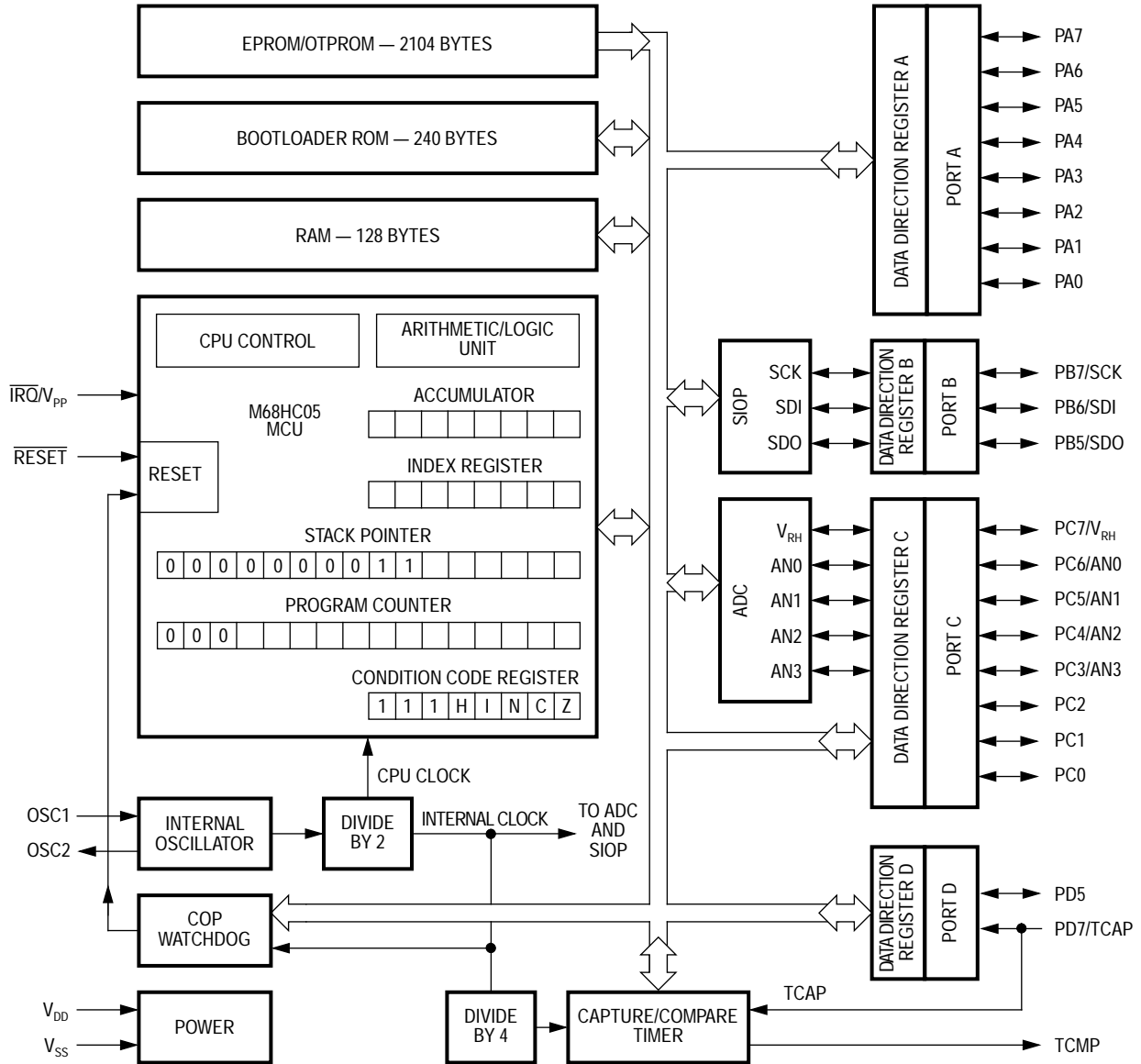


Figure 1. MC68HC705P9 Block Diagram

Package Types and Order Numbers

Table 1. Order Numbers

| Package Type | Case Outline | Pin Count | Operating Temperature | Order Number |
|----------------------------|--------------|-----------|--|---|
| Plastic DIP ⁽¹⁾ | 710 | 28 | 0 to +70 °C –40 to +85 °C –40 to +105 °C –40 to +125 °C | MC68HC705P9P MC68HC705P9CP MC68HC705P9VP MC68HC705P9MP |
| SOIC ⁽²⁾ | 733 | 28 | 0 to +70 °C –40 to +85 °C –40 to +105 °C –40 to +125 °C | MC68HC705P9DW MC68HC705P9CDW MC68HC705P9VDW MC68HC705P9MDW |
| CERDIP ⁽³⁾ | 751F | 28 | 0 to +70 °C –40 to +85 °C –40 to +105 °C –40 to +125 °C | MC68HC705P9S MC68HC705P9CS MC68HC705P9VS MC68HC705P9MS |

1. DIP = dual in-line package
2. SOIC = small outline integrated circuit
3. CERDIP = ceramic DIP

Programmable Options

The options in [Table 2](#) are programmable in the mask option register.

Table 2. Programmable Options

| Feature | Option |
|-----------------------------------|---|
| COP Watchdog | Enabled or Disabled |
| External Interrupt Pin Triggering | Negative-Edge Triggering Only or Negative-Edge and Low-Level Triggering |
| SIO Data Format | MSB First or LSB First |

Contents

| | |
|--------------------------------------|----|
| Pin Assignments | 14 |
| Pin Functions..... | 15 |
| V_{DD} and V_{SS} | 15 |
| OSC1 and OSC2..... | 15 |
| Crystal Connections | 16 |
| Ceramic Resonator Connections | 16 |
| External Clock Connections | 17 |
| $\overline{\text{RESET}}$ | 17 |
| $\overline{\text{IRQ}}/V_{PP}$ | 17 |
| PA7–PA0..... | 17 |
| PB7/SCK–PB5/SDO | 17 |
| PC7/ V_{RH} –PC0 | 17 |
| PD7/TCAP and PD5 | 18 |
| TCMP | 18 |

Pin Assignments

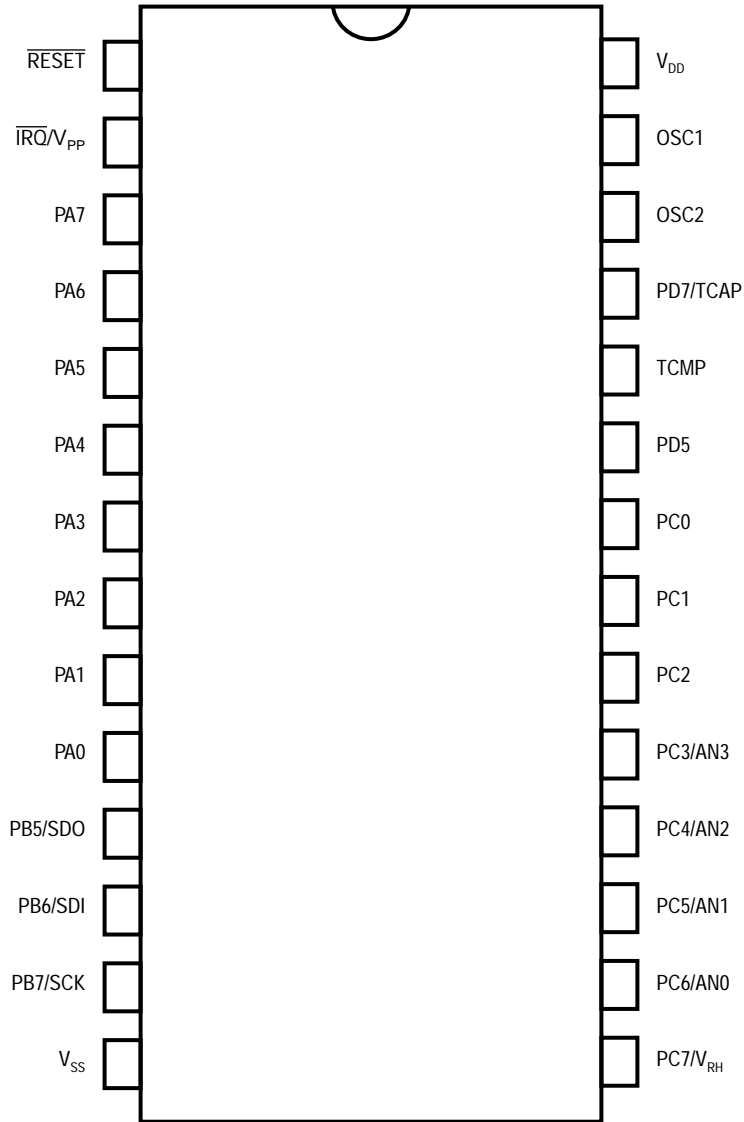


Figure 1. Pin Assignments

Pin Functions

V_{DD} and V_{SS}

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

Very fast signal transitions occur on the MCU pins, placing high short-duration current demands on the power supply. To prevent noise problems, take special care to provide good power supply bypassing at the MCU as **Figure 2** shows. Place 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.

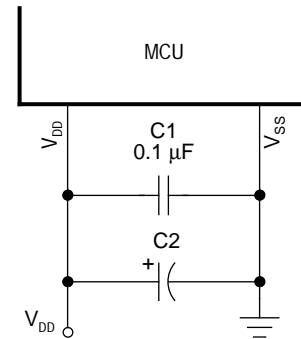


Figure 2. Bypassing Recommendation

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:

- Crystal
- Ceramic resonator
- External clock signal

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

Crystal Connections

The circuit in **Figure 3** shows a typical crystal oscillator circuit for an AT-cut, parallel resonant crystal. Follow the crystal supplier's recommendations, as the crystal parameters determine the external component values required to provide reliable 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.

NOTE: Use an AT-cut crystal. Do not use a strip or tuning fork crystal. The MCU may overdrive or have the incorrect characteristic impedance for a strip or tuning fork crystal.

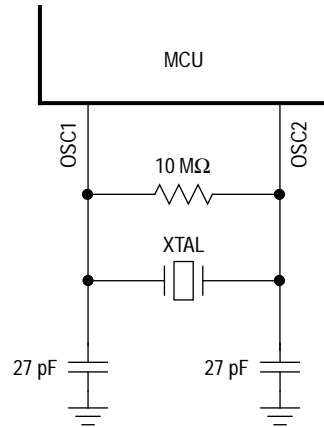


Figure 3. Crystal Connections

Ceramic Resonator Connections

To reduce cost, use a ceramic resonator in place of the crystal. **Figure 4** shows a ceramic resonator circuit. For the values of any external components, follow the recommendations of the resonator manufacturer. The load capacitance values used in the oscillator circuit design should include all stray layout capacitances. To minimize output distortion, mount the resonator and capacitors as close as possible to the pins.

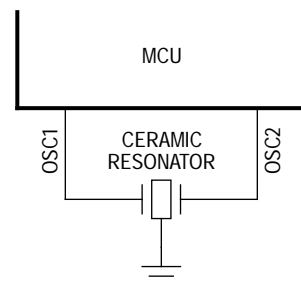


Figure 4. Ceramic Resonator Connections

NOTE: Because the frequency stability of ceramic resonators is not as high as that of crystal oscillators, using a ceramic resonator may degrade the performance of the ADC.

External Clock Connections

An external clock from another CMOS-compatible device can drive the OSC1 input, with the OSC2 pin unconnected, as **Figure 5** shows.

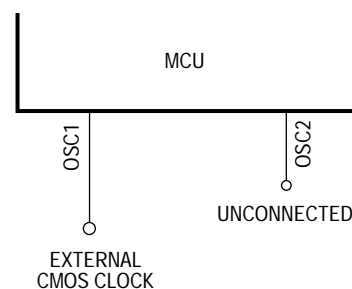


Figure 5. External Clock Connections

$\overline{\text{RESET}}$

A logic zero on the $\overline{\text{RESET}}$ pin forces the MCU to a known startup state. The $\overline{\text{RESET}}$ pin input circuit contains an internal Schmitt trigger to improve noise immunity.

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

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

- Applying asynchronous external interrupt signals
- Applying V_{PP} , the EPROM/OTPROM programming voltage

PA7–PA0

PA7–PA0 are general-purpose bidirectional I/O port pins. Use data direction register A to configure port A pins as inputs or outputs.

PB7/SCK–PB5/SDO

Port B is a 3-pin bidirectional I/O port that shares its pins with the SIOP. Use data direction register B to configure port B pins as inputs or outputs.

PC7/ V_{RH} –PC0

Port C is an 8-pin bidirectional I/O port that shares five of its pins with the ADC. Use data direction register C to configure port C pins as inputs or outputs.

| | |
|------------------|---|
| PD7/TCAP and PD5 | Port D is a 2-pin I/O port that shares one of its pins with the capture/compare timer. Use data direction register D to configure port D pins as inputs or outputs. |
| TCMP | The TCMP pin is the output compare pin for the capture/compare timer. |

Contents

| | |
|-------------------------------------|----|
| Features | 19 |
| Memory Map | 20 |
| Input/Output Register Summary | 21 |
| RAM | 24 |
| EPROM/OTPROM | 25 |
| EPROM/OTPROM Programming | 26 |
| EPROM Programming Register | 26 |
| Bootloader ROM | 27 |
| EPROM Erasing | 30 |
| Mask Option Register | 31 |

Features

- 2104 Bytes of EPROM/OTPROM
 - 48 Bytes of Page Zero EPROM/OTPROM
 - Eight Locations for User Vectors
- 128 Bytes of User RAM
- Bootloader ROM

Memory Map

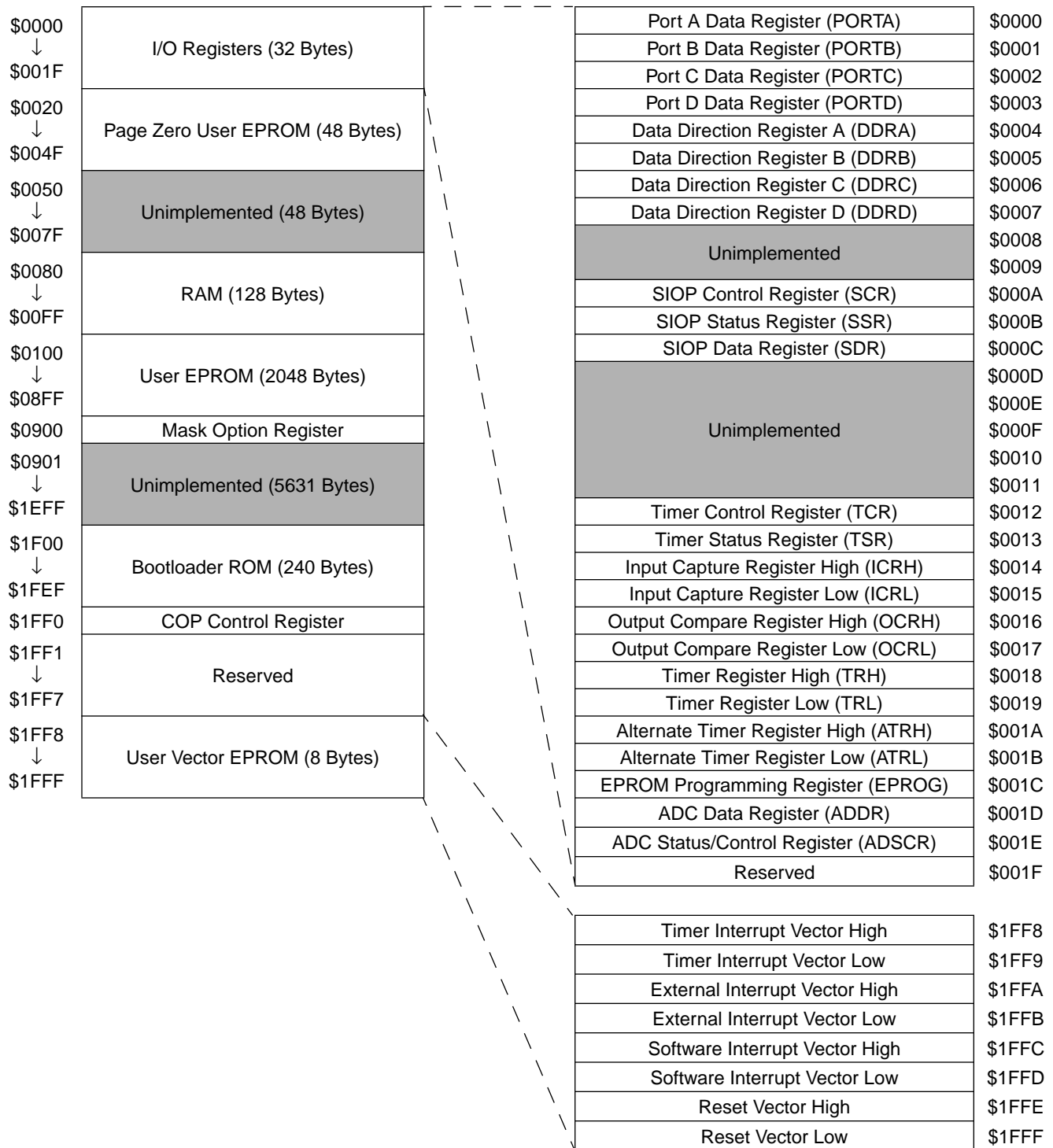


Figure 1. Memory Map

Input/Output Register Summary

| Addr. | Name | R/W | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|----------------------------------|---------------------------|---------------------|-------|-------|-------|-------|-------|-------|-------|
| \$0000 | Port A Data Register (PORTA) | Read: Write: Reset: | PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| | | | Unaffected by reset | | | | | | | |
| \$0001 | Port B Data Register (PORTB) | Read: Write: Reset: | PB7 | PB6 | PB5 | 0 | 0 | 0 | 0 | 0 |
| | | | Unaffected by reset | | | | | | | |
| \$0002 | Port C Data Register (PORTC) | Read: Write: Reset: | PC7 | PC6 | PC5 | PC4 | PC3 | PC2 | PC1 | PC0 |
| | | | Unaffected by reset | | | | | | | |
| \$0003 | Port D Data Register (PORTD) | Read: Write: Reset: | PD7 | 0 | PD5 | 1 | 0 | 0 | 0 | 0 |
| | | | Unaffected by reset | | | | | | | |
| \$0004 | Data Direction Register A (DDRA) | Read: Write: Reset: | DDRA7 | DDRA6 | DDRA5 | DDRA4 | DDRA3 | DDRA2 | DDRA1 | DDRA0 |
| | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0005 | Data Direction Register B (DDRB) | Read: Write: Reset: | DDRB7 | DDRB6 | DDRB5 | 0 | 0 | 0 | 0 | 0 |
| | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0006 | Data Direction Register C (DDRC) | Read: Write: Reset: | DDRC7 | DDRC6 | DDRC5 | DDRC4 | DDRC3 | DDRC2 | DDRC1 | DDRC0 |
| | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0007 | Data Direction Register D (DDRD) | Read: Write: Reset: | 0 | 0 | DDRD5 | 0 | 0 | 0 | 0 | 0 |
| | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0008 | Unimplemented | | | | | | | | | |
| \$0009 | Unimplemented | | | | | | | | | |

= Unimplemented R = Reserved U = Unaffected

Figure 2. I/O Register Summary

| Addr. | Name | R/W | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------------------------------------|--|--------|------|------|------|----|----|------|-------|
| \$000A | SIOPI Control Register (SCR) | Read: 0 Write: 0 Reset: 0 | 0 | SPE | 0 | MSTR | 0 | 0 | 0 | 0 |
| \$000B | SIOPI Status Register (SSR) | Read: SPIF Write: DCOL Reset: 0 | SPIF | DCOL | 0 | 0 | 0 | 0 | 0 | 0 |
| \$000C | SIOPI Data Register (SDR) | Read: Bit 7 Write: Bit 0 Reset: Unaffected by reset | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| \$000D | Unimplemented | | | | | | | | | |
| \$000E | Unimplemented | | | | | | | | | |
| \$000F | Unimplemented | | | | | | | | | |
| \$0010 | Unimplemented | | | | | | | | | |
| \$0011 | Unimplemented | | | | | | | | | |
| \$0012 | Timer Control Register (TCR) | Read: ICIE Write: OCIE Reset: 0 | ICIE | OCIE | TOIE | 0 | 0 | 0 | IEDG | OLVL |
| \$0013 | Timer Status Register (TSR) | Read: ICF Write: TOF Reset: Unaffected by reset | ICF | OCF | TOF | 0 | 0 | 0 | 0 | 0 |
| \$0014 | Input Capture Register High (ICRH) | Read: Bit 15 Write: Bit 8 Reset: | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit 8 |
| \$0015 | Input Capture Register Low (ICRL) | Read: Bit 7 Write: Bit 0 Reset: Unaffected by reset | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| \$0016 | Output Compare Register High (OCRH) | Read: Bit 15 Write: Bit 8 Reset: Unaffected by reset | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit 8 |

= Unimplemented R = Reserved U = Unaffected

Figure 2. I/O Register Summary (Continued)

| Addr. | Name | R/W | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|--------------------------------------|---------------------------|--------------------------------|------|------|----|----|-------|-----|-------|
| \$0017 | Output Compare Register Low (OCRL) | Read: Write: Reset: | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | | Unaffected by reset | | | | | | | |
| \$0018 | Timer Register High (TRH) | Read: Write: Reset: | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit 8 |
| | | | Reset initializes TRH to \$FF | | | | | | | |
| \$0019 | Timer Register Low (TRL) | Read: Write: Reset: | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | | Reset initializes TRL to \$FC | | | | | | | |
| \$001A | Alternate Timer Register High (ATRH) | Read: Write: Reset: | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit 8 |
| | | | Reset initializes ATRH to \$FF | | | | | | | |
| \$001B | Alternate Timer Register Low (ATRL) | Read: Write: Reset: | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | | Reset initializes ATRL to \$FC | | | | | | | |
| \$001C | EPROM Programming Register (EPROG) | Read: Write: Reset: | 0 | 0 | 0 | 0 | 0 | LATCH | 0 | EPM |
| | | | R | R | R | R | R | R | R | |
| | | | Unaffected by reset | | | | | | | |
| \$001D | ADC Data Register (ADDR) | Read: Write: Reset: | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | | Unaffected by reset | | | | | | | |
| \$001E | ADC Status/Control Register (ADSCR) | Read: Write: Reset: | CCF | ADRC | ADON | 0 | 0 | CH2 | CH1 | CH0 |
| | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$001F | Reserved | Read: Write: Reset: | R | R | R | R | R | R | R | R |
| | | | Unaffected by reset | | | | | | | |

= Unimplemented
 R = Reserved
 U = Unaffected

Figure 2. I/O Register Summary (Continued)

| Addr. | Name | R/W | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|----------------------------|--------|---------------------|---|---|---|---|------|-----|-------|
| \$0900 | Mask Option Register (MOR) | Read: | 0 | 0 | 0 | 0 | 0 | SIOP | IRQ | COPE |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| \$1FF0 | COP Register (COPR) | Read: | R | R | R | R | R | R | R | COPC |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |

= Unimplemented R = Reserved U = Unaffected

Figure 2. I/O Register Summary (Continued)

RAM

The 128 addresses from \$0080–\$00FF are RAM locations. The CPU uses the top 64 RAM addresses, \$00C0–\$00FF, as the stack. 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.*

EPROM/OTPROM

An MCU with a quartz window has 2104 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 programming the MCU. Ambient light may affect MCU operation.*

In an MCU without the quartz window, the EPROM cannot be erased and serves as 2104 bytes of one-time programmable ROM (OTPROM). The following addresses are user EPROM/OTPROM locations:

- \$0020–\$004F
- \$0100–\$08FF
- \$1FF8–\$1FFF (reserved for user-defined interrupt and reset vectors)

The mask option register (MOR) is an EPROM/OTPROM location at address \$0900.

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
- Activating the bootloader ROM to download the contents of an external memory device to the on-chip EPROM/OTPROM

EPROM Programming Register

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

| \$001C | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|---|---|---|---|-------|---|-------|
| Read: | 0 | 0 | 0 | 0 | 0 | LATCH | 0 | EPGM |
| Write: | R | R | R | R | R | | R | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

R = Reserved

Figure 3. EPROM Programming Register (EPROG)

LATCH — EPROM Bus Latch

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

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

0 = Address and data buses configured for normal operation

EPGM bit— EPROM Programming

This read/write bit applies the voltage from the \overline{IRQ}/V_{PP} pin to the EPROM/OTPROM. To write the EPGM bit, the LATCH bit must already be set. Clearing the LATCH bit also clears the EPGM bit. Resets clear the EPGM bit.

1 = EPROM/OTPROM programming power switched on

0 = EPROM/OTPROM programming power switched off

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

Bits 7–3 and Bit 1— Reserved

Bits 7–3 and bit 1 are factory test bits that always read as logic zeros.

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

1. Apply 16.5 V to the $\overline{\text{IRQ}}/V_{\text{PP}}$ pin.
2. Set the LATCH bit.
3. Write to any EPROM/OTPROM address.
4. Set the EPGM bit for a time, t_{EPGM} , to apply the programming voltage.
5. Clear the LATCH bit.

Bootloader ROM

The bootloader ROM, located at addresses \$1F00–\$1FEF, contains routines for copying an external EPROM to the on-chip EPROM/OTPROM.

The bootloader copies to the following EPROM/OTPROM addresses:

- \$0020–\$004F
- \$0100–\$0900
- \$1FF0–\$1FFF

The addresses of the code in the external EPROM must match the MC68HC705P9 addresses. The bootloader ignores all other addresses.

Figure 4 shows the circuit for downloading to the on-chip EPROM/OTPROM from a 2764 EPROM. The bootloader circuit includes an external 12-bit counter to address the external EPROM. Operation is fastest when unused external EPROM addresses contain \$00. The bootloader function begins when a rising edge occurs on the $\overline{\text{RESET}}$ pin while the V_{PP} voltage is on the $\overline{\text{IRQ}}/V_{\text{PP}}$ pin, and the PD7/TCAP pin is at logic one.

The logical states of the PC4/AN2 and PC3/AN3 pins select the bootloader function, as **Table 1** shows.

Table 1. Bootloader Function Selection

| PC4/AN2 | PC3/AN3 | Function |
|---------|---------|--------------------|
| 1 | 1 | Program and Verify |
| 1 | 0 | Verify Only |

Complete the following steps to bootload the MCU:

1. Turn off all power to the circuit.
2. Install the EPROM containing the code to be downloaded.
3. Install the MCU.
4. Select the bootloader function:
 - a. Open switches S2 and S3 to select the program and verify function.
 - b. Open only switch S2 to select only the verify function.
5. Close switch S1.
6. Turn on the V_{DD} power supply.

CAUTION: Turn on the V_{DD} power supply **before** turning on the V_{PP} power supply.

7. Turn on the V_{PP} power supply.
8. Open switch S1. The bootloader code begins to execute. If the PROGRAM function is selected, the PROGRAM LED turns on during programming. If the VERIFY function is selected, the VERIFY LED turns on when verification is successful. The PROGRAM and VERIFY functions take about 10 seconds.
9. Close switch S1.
10. Turn off the V_{PP} power supply.

CAUTION: Turn off the V_{PP} power supply **before** turning off the V_{DD} power supply.

11. Turn off the V_{DD} power supply.

EPROM Erasing

The erased state of an EPROM bit is zero. 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.

Cerdip packages have a transparent window for erasing the EPROM with ultraviolet light. In the windowless PDIP and SOIC packages, the 2104 EPROM bytes function as one-time programmable ROM (OTPROM).

Mask Option Register

The mask option register (MOR) is an EPROM/OTPROM byte that is programmable only with the bootloader function. The MOR controls the following options:

- LSB first or MSB first SIOP data transfer
- Edge-triggered or edge- and level-triggered external interrupt pin
- Enabled or disabled COP watchdog

To program the MOR, use the 5-step procedure given in the section [EPROM Programming Register](#) on page 26. Write to address \$0900 in step 3.

| \$0900 | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---------|---------------------|---|---|---|---|------|-----|-------|
| Read: | 0 | 0 | 0 | 0 | 0 | SIOP | IRQ | COPE |
| Write: | | | | | | | | |
| Reset: | Unaffected by reset | | | | | | | |
| Erased: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |


 = Unimplemented

Figure 5. Mask Option Register (MOR)

SIOP — Serial I/O Port

The SIOP bit controls the shift direction into and out of the SIOP shift register.

- 1 = SIOP data transferred LSB first (bit 0 first)
- 0 = SIOP data transferred MSB first (bit 7 first)

IRQ — Interrupt Request

The IRQ bit makes the external interrupt function of the $\overline{\text{IRQ}}/V_{\text{PP}}$ pin level-triggered as well as edge-triggered.

- 1 = $\overline{\text{IRQ}}/V_{\text{PP}}$ pin negative-edge triggered and low-level triggered
- 0 = $\overline{\text{IRQ}}/V_{\text{PP}}$ pin negative-edge triggered only

COPE — COP Enable

COPE enables the COP watchdog. In applications that have wait cycles longer than the COP watchdog timeout period, the COP watchdog can be disabled by not programming the COPE bit to logic one.

1 = COP watchdog enabled

0 = COP watchdog disabled

Central Processor Unit CPU

Contents

| | |
|--------------------------------------|----|
| Features | 34 |
| Introduction | 34 |
| CPU Control Unit | 36 |
| Arithmetic/Logic Unit | 36 |
| CPU Registers | 36 |
| Accumulator | 37 |
| Index Register | 37 |
| Stack Pointer | 37 |
| Program Counter | 38 |
| Condition Code Register | 38 |
| Instruction Set | 40 |
| Addressing Modes | 40 |
| Inherent | 40 |
| Immediate | 40 |
| Direct | 41 |
| Extended | 41 |
| Indexed, No Offset | 41 |
| Indexed, 8-Bit Offset | 41 |
| Indexed, 16-Bit Offset | 42 |
| Relative | 42 |
| Instruction Types | 43 |
| Register/Memory Instructions | 43 |
| Read-Modify-Write Instructions | 44 |
| Jump/Branch Instructions | 45 |
| Bit Manipulation Instructions | 47 |
| Control Instructions | 47 |
| Instruction Set Summary | 48 |

Features

- 2.1-MHz Bus Frequency
- 8-Bit Accumulator
- 8-Bit Index Register
- 13-Bit Program Counter
- 6-Bit Stack Pointer
- Condition Code Register with Five Status Flags
- 62 Instructions
- Eight Addressing Modes
- Power-Saving Stop, Wait, and Data-Retention Modes

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.

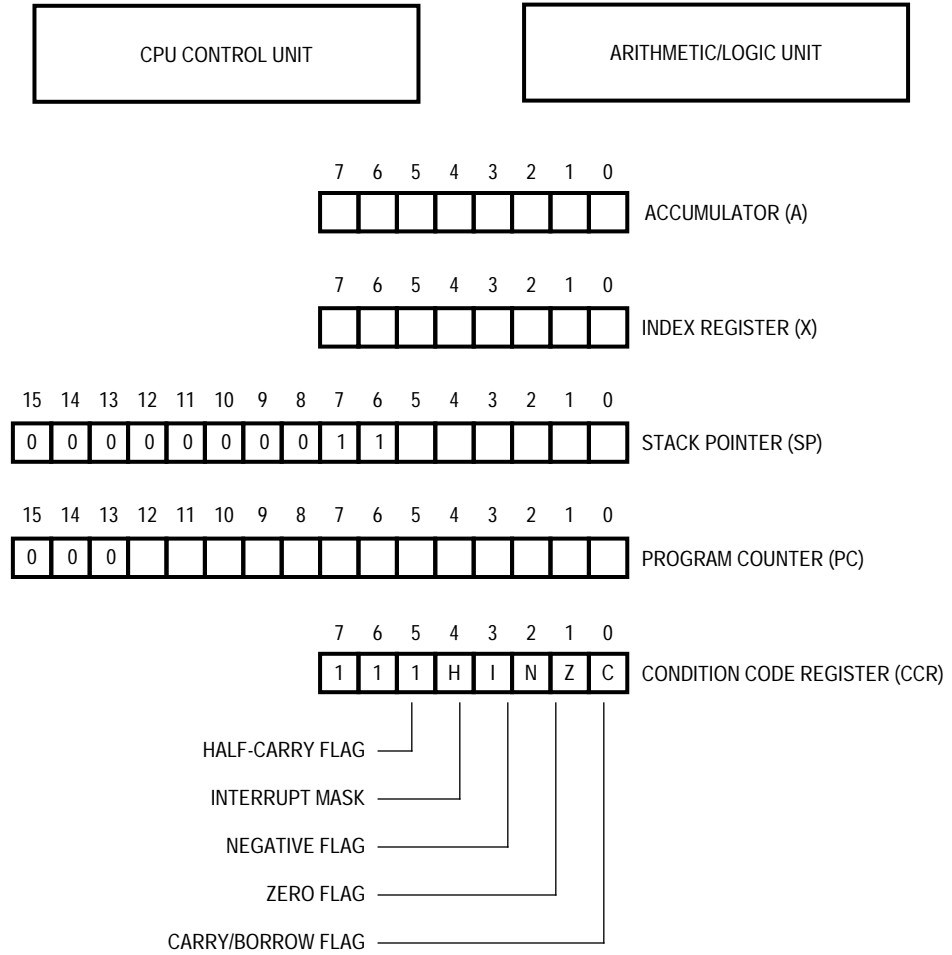


Figure 1. CPU Programming Model

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.

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).

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.

Accumulator

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

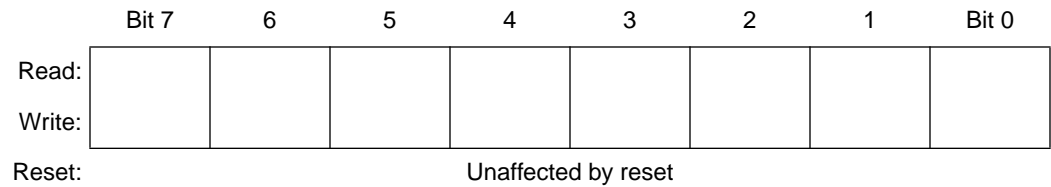


Figure 2. Accumulator (A)

Index Register

The index register can be used for data storage or as a counter. In the indexed addressing modes, the CPU uses the byte in the index register to determine the effective address of the operand.



Figure 3. Index Register (X)

Stack Pointer

The stack pointer is a 16-bit register that contains the address of the next stack location to be used. 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.

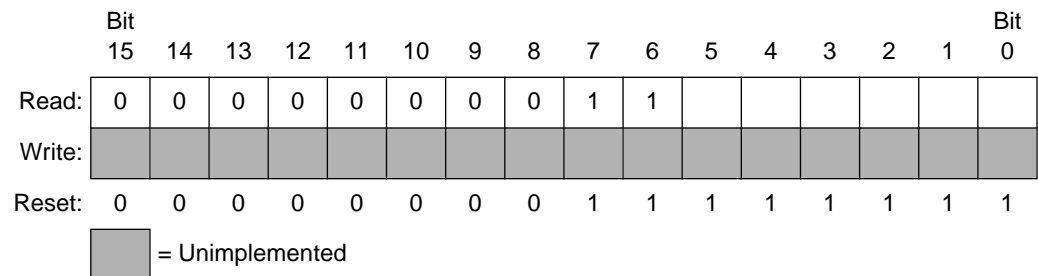


Figure 4. Stack Pointer (SP)

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.

Program Counter

The program counter is a 16-bit register that contains the address of the next instruction or operand to be fetched. The three most significant bits of the program counter are ignored internally and appear as 000.

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.

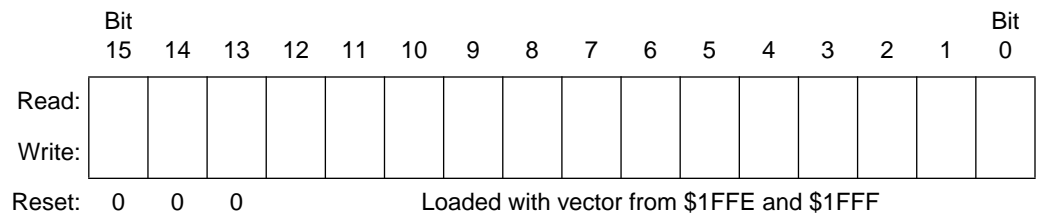


Figure 5. Program Counter (PC)

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.

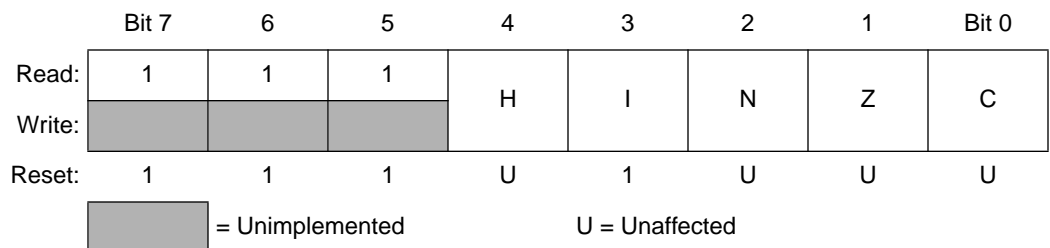


Figure 6. Condition Code Register (CCR)

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

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

N — Negative Flag

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

Z — Zero Flag

The CPU sets the zero flag when an arithmetic operation, logical operation, or data manipulation 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.

Instruction Set

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

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

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.

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.

- 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.
- 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 Motorola assembler, the programmer does not need to specify whether an instruction is direct or extended. The assembler automatically selects the shortest form of the instruction.
- Indexed,
No Offset* Indexed instructions with no offset are 1-byte instructions that can access data with variable addresses within the first 256 memory locations. The index register contains the low byte of the effective address of the operand. The CPU automatically uses \$00 as the high byte, so these instructions can address locations \$0000–\$00FF.
- Indexed, no offset instructions are often used to move a pointer through a table or to hold the address of a frequently used RAM or I/O location.
- 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.

*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 Motorola assembler determines the shortest form of indexed addressing.

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

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

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 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 |

Read-Modify-Write Instructions

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

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

Table 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.

*Jump/Branch
Instructions*

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

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

Table 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 |

*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. Bit Manipulation Instructions

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

*Control
Instructions*

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

Table 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 |

Instruction Set Summary

Table 6. Instruction Set Summary

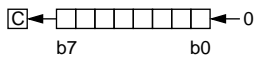
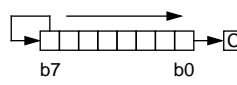
| Source Form | Operation | Description | Effect on CCR | | | | | Address Mode | Opcode | Operand | Cycles |
|--|---------------------------------------|--|---------------|----|----|----|----|--|--|--|--------------------------------------|
| | | | H | I | N | Z | C | | | | |
| ADC #opr ADC opr ADC opr ADC opr,X ADC opr,X ADC ,X | Add with Carry | $A \leftarrow (A) + (M) + (C)$ | ↑x— | ↑x | ↑x | ↑x | ↑x | IMM DIR EXT IX2 IX1 IX | A9 B9 C9 D9 E9 F9 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| ADD #opr ADD opr ADD opr ADD opr,X ADD opr,X ADD ,X | Add without Carry | $A \leftarrow (A) + (M)$ | ↑x— | ↑x | ↑ | ↑ | ↑ | IMM DIR EXT IX2 IX1 IX | AB BB CB DB EB FB | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| AND #opr AND opr AND opr AND opr,X AND opr,X AND ,X | Logical AND | $A \leftarrow (A) \wedge (M)$ | — | — | ↑x | ↑ | — | IMM DIR EXT IX2 IX1 IX | A4 B4 C4 D4 E4 F4 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| ASL opr ASLA ASLX ASL opr,X ASL ,X | Arithmetic Shift Left (Same as LSL) |  | — | — | ↑x | ↑ | ↑ | DIR INH INH IX1 IX | 38 48 58 68 78 | dd dd dd ff | 5 3 3 6 5 |
| ASR opr ASRA ASRX ASR opr,X ASR ,X | Arithmetic Shift Right |  | — | — | ↑x | ↑ | ↑ | DIR INH INH IX1 IX | 37 47 57 67 77 | dd dd dd ff | 5 3 3 6 5 |
| BCC rel | Branch if Carry Bit Clear | $PC \leftarrow (PC) + 2 + rel ? C = 0$ | — | — | — | — | — | REL | 24 | rr | 3 |
| BCLR n opr | Clear Bit n | $M_n \leftarrow 0$ | — | — | — | — | — | DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7) | 11 13 15 17 19 1B 1D 1F | dd dd dd dd dd dd dd dd | 5 5 5 5 5 5 5 5 |
| BCS rel | Branch if Carry Bit Set (Same as BLO) | $PC \leftarrow (PC) + 2 + rel ? C = 1$ | — | — | — | — | — | REL | 25 | rr | 3 |
| BEQ rel | Branch if Equal | $PC \leftarrow (PC) + 2 + rel ? Z = 1$ | — | — | — | — | — | REL | 27 | rr | 3 |
| BHCC rel | Branch if Half-Carry Bit Clear | $PC \leftarrow (PC) + 2 + rel ? H = 0$ | — | — | — | — | — | REL | 28 | rr | 3 |
| BHCS rel | Branch if Half-Carry Bit Set | $PC \leftarrow (PC) + 2 + rel ? H = 1$ | — | — | — | — | — | REL | 29 | rr | 3 |

Table 6. Instruction Set Summary (Continued)

| Source Form | Operation | Description | Effect on CCR | | | | | Address Mode | Opcode | Operand | Cycles |
|--|---------------------------------------|---|---------------|-------|-------|---|-----|--------------|--------|---------|--------|
| | | | H | I | N | Z | C | | | | |
| BHI <i>rel</i> | Branch if Higher | $PC \leftarrow (PC) + 2 + rel ? C \vee Z = 0$ | — | — | — | — | — | REL | 22 | rr | 3 |
| BHS <i>rel</i> | Branch if Higher or Same | $PC \leftarrow (PC) + 2 + rel ? C = 0$ | — | — | — | — | — | REL | 24 | rr | 3 |
| BIH <i>rel</i> | Branch if IRQ Pin High | $PC \leftarrow (PC) + 2 + rel ? IRQ = 1$ | — | — | — | — | — | REL | 2F | rr | 3 |
| BIL <i>rel</i> | Branch if IRQ Pin Low | $PC \leftarrow (PC) + 2 + rel ? IRQ = 0$ | — | — | — | — | — | REL | 2E | rr | 3 |
| BIT # <i>opr</i> BIT <i>opr</i> BIT <i>opr</i> BIT <i>opr</i> ,X BIT <i>opr</i> ,X BIT ,X | Bit Test Accumulator with Memory Byte | $(A) \wedge (M)$ | — | — | ↕x | ↑ | — | IMM | A5 | ii | 2 |
| DIR | | | B5 | dd | 3 | | | | | | |
| EXT | | | C5 | hh ll | 4 | | | | | | |
| IX2 | | | D5 | ee ff | 5 | | | | | | |
| IX1 | | | E5 | ff | 4 | | | | | | |
| IX | | | F5 | | 3 | | | | | | |
| BLO <i>rel</i> | Branch if Lower (Same as BCS) | $PC \leftarrow (PC) + 2 + rel ? C = 1$ | — | — | — | — | — | REL | 25 | rr | 3 |
| BLS <i>rel</i> | Branch if Lower or Same | $PC \leftarrow (PC) + 2 + rel ? C \vee Z = 1$ | — | — | — | — | — | REL | 23 | rr | 3 |
| BMC <i>rel</i> | Branch if Interrupt Mask Clear | $PC \leftarrow (PC) + 2 + rel ? I = 0$ | — | — | — | — | — | REL | 2C | rr | 3 |
| BMI <i>rel</i> | Branch if Minus | $PC \leftarrow (PC) + 2 + rel ? N = 1$ | — | — | — | — | — | REL | 2B | rr | 3 |
| BMS <i>rel</i> | Branch if Interrupt Mask Set | $PC \leftarrow (PC) + 2 + rel ? I = 1$ | — | — | — | — | — | REL | 2D | rr | 3 |
| BNE <i>rel</i> | Branch if Not Equal | $PC \leftarrow (PC) + 2 + rel ? Z = 0$ | — | — | — | — | — | REL | 26 | rr | 3 |
| BPL <i>rel</i> | Branch if Plus | $PC \leftarrow (PC) + 2 + rel ? N = 0$ | — | — | — | — | — | REL | 2A | rr | 3 |
| BRA <i>rel</i> | Branch Always | $PC \leftarrow (PC) + 2 + rel ? 1 = 1$ | — | — | — | — | — | REL | 20 | rr | 3 |
| BRCLR <i>n opr rel</i> | Branch if Bit n Clear | $PC \leftarrow (PC) + 2 + rel ? Mn = 0$ | — | — | — | — | ↕x | DIR (b0) | 01 | dd rr | 5 |
| | | | DIR (b1) | 03 | dd rr | 5 | | | | | |
| | | | DIR (b2) | 05 | dd rr | 5 | | | | | |
| | | | DIR (b3) | 07 | dd rr | 5 | | | | | |
| | | | DIR (b4) | 09 | dd rr | 5 | | | | | |
| | | | DIR (b5) | 0B | dd rr | 5 | | | | | |
| | | | DIR (b6) | 0D | dd rr | 5 | | | | | |
| | | | DIR (b7) | 0F | dd rr | 5 | | | | | |
| BRN <i>rel</i> | Branch Never | $PC \leftarrow (PC) + 2 + rel ? 1 = 0$ | — | — | — | — | REL | 21 | rr | 3 | |
| BRSET <i>n opr rel</i> | Branch if Bit n Set | $PC \leftarrow (PC) + 2 + rel ? Mn = 1$ | — | — | — | — | ↕x | DIR (b0) | 00 | dd rr | 5 |
| | | | DIR (b1) | 02 | dd rr | 5 | | | | | |
| | | | DIR (b2) | 04 | dd rr | 5 | | | | | |
| | | | DIR (b3) | 06 | dd rr | 5 | | | | | |
| | | | DIR (b4) | 08 | dd rr | 5 | | | | | |
| | | | DIR (b5) | 0A | dd rr | 5 | | | | | |
| | | | DIR (b6) | 0C | dd rr | 5 | | | | | |
| | | | DIR (b7) | 0E | dd rr | 5 | | | | | |
| BSET <i>n opr</i> | Set Bit n | $Mn \leftarrow 1$ | — | — | — | — | — | DIR (b0) | 10 | dd | 5 |
| | | | DIR (b1) | 12 | dd | 5 | | | | | |
| | | | DIR (b2) | 14 | dd | 5 | | | | | |
| | | | DIR (b3) | 16 | dd | 5 | | | | | |
| | | | DIR (b4) | 18 | dd | 5 | | | | | |
| | | | DIR (b5) | 1A | dd | 5 | | | | | |
| | | | DIR (b6) | 1C | dd | 5 | | | | | |
| | | | DIR (b7) | 1E | dd | 5 | | | | | |

Table 6. Instruction Set Summary (Continued)

| Source Form | Operation | Description | Effect on CCR | | | | | Address Mode | Opcode | Operand | Cycles |
|--|---|---|---------------|---|----|----|---|---------------------------------------|----------------------------------|----------------------------------|----------------------------|
| | | | H | I | N | Z | C | | | | |
| BSR <i>rel</i> | Branch to Subroutine | PC ← (PC) + 2; push (PCL) SP ← (SP) - 1; push (PCH) SP ← (SP) - 1 PC ← (PC) + <i>rel</i> | — | — | — | — | — | REL | AD | rr | 6 |
| CLC | Clear Carry Bit | C ← 0 | — | — | — | — | 0 | INH | 98 | | 2 |
| CLI | Clear Interrupt Mask | I ← 0 | — | 0 | — | — | — | INH | 9A | | 2 |
| CLR <i>opr</i> CLRA CLR X CLR <i>opr,X</i> CLR ,X | Clear Byte | M ← \$00 A ← \$00 X ← \$00 M ← \$00 M ← \$00 | — | — | 0 | 1 | — | DIR INH INH IX1 IX | 3F 4F 5F 6F 7F | dd ff | 5 3 3 6 5 |
| CMP # <i>opr</i> CMP <i>opr</i> CMP <i>opr</i> CMP <i>opr,X</i> CMP <i>opr,X</i> CMP ,X | Compare Accumulator with Memory Byte | (A) - (M) | — | — | ↓x | ↑ | ↑ | IMM DIR EXT IX2 IX1 IX | A1 B1 C1 D1 E1 F1 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| COM <i>opr</i> COMA COM X COM <i>opr,X</i> COM ,X | Complement Byte (One's Complement) | M ← (M̄) = \$FF - (M) A ← (Ā) = \$FF - (A) X ← (X̄) = \$FF - (X) M ← (M̄) = \$FF - (M) M ← (M̄) = \$FF - (M) | — | — | ↓x | ↓x | 1 | DIR INH INH IX1 IX | 33 43 53 63 73 | dd ff | 5 3 3 6 5 |
| CPX # <i>opr</i> CPX <i>opr</i> CPX <i>opr</i> CPX <i>opr,X</i> CPX <i>opr,X</i> CPX ,X | Compare Index Register with Memory Byte | (X) - (M) | — | — | ↓x | ⊗ | ⊗ | IMM DIR EXT IX2 IX1 IX | A3 B3 C3 D3 E3 F3 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| DEC <i>opr</i> DECA DEC X DEC <i>opr,X</i> DEC ,X | Decrement Byte | M ← (M) - 1 A ← (A) - 1 X ← (X) - 1 M ← (M) - 1 M ← (M) - 1 | — | — | ↓x | ↓x | — | DIR INH INH IX1 IX | 3A 4A 5A 6A 7A | dd ff | 5 3 3 6 5 |
| EOR # <i>opr</i> EOR <i>opr</i> EOR <i>opr</i> EOR <i>opr,X</i> EOR <i>opr,X</i> EOR ,X | EXCLUSIVE OR Accumulator with Memory Byte | A ← (A) ⊕ (M) | — | — | ↓x | ↑ | — | IMM DIR EXT IX2 IX1 IX | A8 B8 C8 D8 E8 F8 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| INC <i>opr</i> INCA INC X INC <i>opr,X</i> INC ,X | Increment Byte | M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1 | — | — | ↓x | ↓x | — | DIR INH INH IX1 IX | 3C 4C 5C 6C 7C | dd ff | 5 3 3 6 5 |

Table 6. Instruction Set Summary (Continued)

| Source Form | Operation | Description | Effect on CCR | | | | | Address Mode | Opcode | Operand | Cycles |
|--|--------------------------------------|--|---------------|---|---|---|-----|--------------|--------|---------|--------|
| | | | H | I | N | Z | C | | | | |
| JMP <i>opr</i> JMP <i>opr</i> JMP <i>opr,X</i> JMP <i>opr,X</i> JMP ,X | Unconditional Jump | PC ← Jump Address | | | | | | DIR | BC | dd | 2 |
| | | | | | | | EXT | CC | hh ll | 3 | |
| | | | | | | | IX2 | DC | ee ff | 4 | |
| | | | | | | | IX1 | EC | ff | 3 | |
| | | | | | | | IX | FC | | 2 | |
| JSR <i>opr</i> JSR <i>opr</i> JSR <i>opr,X</i> JSR <i>opr,X</i> JSR ,X | Jump to Subroutine | PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) - 1 Push (PCH); SP ← (SP) - 1 PC ← Effective Address | | | | | | DIR | BD | dd | 5 |
| | | | | | | | EXT | CD | hh ll | 6 | |
| | | | | | | | IX2 | DD | ee ff | 7 | |
| | | | | | | | IX1 | ED | ff | 6 | |
| | | | | | | | IX | FD | | 5 | |
| LDA # <i>opr</i> LDA <i>opr</i> LDA <i>opr</i> LDA <i>opr,X</i> LDA <i>opr,X</i> LDA ,X | Load Accumulator with Memory Byte | A ← (M) | | | | | | IMM | A6 | ii | 2 |
| | | | | | | | DIR | B6 | dd | 3 | |
| | | | | | | | EXT | C6 | hh ll | 4 | |
| | | | | | | | IX2 | D6 | ee ff | 5 | |
| | | | | | | | IX1 | E6 | ff | 4 | |
| | | | | | | | IX | F6 | | 3 | |
| LDX # <i>opr</i> LDX <i>opr</i> LDX <i>opr</i> LDX <i>opr,X</i> LDX <i>opr,X</i> LDX ,X | Load Index Register with Memory Byte | X ← (M) | | | | | | IMM | AE | ii | 2 |
| | | | | | | | DIR | BE | dd | 3 | |
| | | | | | | | EXT | CE | hh ll | 4 | |
| | | | | | | | IX2 | DE | ee ff | 5 | |
| | | | | | | | IX1 | EE | ff | 4 | |
| | | | | | | | IX | FE | | 3 | |
| LSL <i>opr</i> LSLA LSLX LSL <i>opr,X</i> LSL ,X | Logical Shift Left (Same as ASL) | | | | | | | DIR | 38 | dd | 5 |
| | | | | | | | INH | 48 | | 3 | |
| | | | | | | | INH | 58 | | 3 | |
| | | | | | | | IX1 | 68 | ff | 6 | |
| | | | | | | | IX | 78 | | 5 | |
| | | | | | | | | | | | |
| LSR <i>opr</i> LSRA LSRX LSR <i>opr,X</i> LSR ,X | Logical Shift Right | | | | | | | DIR | 34 | dd | 5 |
| | | | | | | | INH | 44 | | 3 | |
| | | | | | | | INH | 54 | | 3 | |
| | | | | | | | IX1 | 64 | ff | 6 | |
| | | | | | | | IX | 74 | | 5 | |
| | | | | | | | | | | | |
| MUL | Unsigned Multiply | X : A ← (X) × (A) | 0 | | | | 0 | INH | 42 | | 11 |
| NEG <i>opr</i> NEGA NEGX NEG <i>opr,X</i> NEG ,X | Negate Byte (Two's Complement) | $M \leftarrow -(M) = \$00 - (M)$ $A \leftarrow -(A) = \$00 - (A)$ $X \leftarrow -(X) = \$00 - (X)$ $M \leftarrow -(M) = \$00 - (M)$ $M \leftarrow -(M) = \$00 - (M)$ | | | | | | DIR | 30 | dd | 5 |
| | | | | | | | INH | 40 | | 3 | |
| | | | | | | | INH | 50 | | 3 | |
| | | | | | | | IX1 | 60 | ff | 6 | |
| | | | | | | | IX | 70 | | 5 | |
| NOP | No Operation | | | | | | | INH | 9D | | 2 |
| ORA # <i>opr</i> ORA <i>opr</i> ORA <i>opr</i> ORA <i>opr,X</i> ORA <i>opr,X</i> ORA ,X | Logical OR Accumulator with Memory | A ← (A) ∨ (M) | | | | | | IMM | AA | ii | 2 |
| | | | | | | | DIR | BA | dd | 3 | |
| | | | | | | | EXT | CA | hh ll | 4 | |
| | | | | | | | IX2 | DA | ee ff | 5 | |
| | | | | | | | IX1 | EA | ff | 4 | |
| | | | | | | | IX | FA | | 3 | |

Table 6. Instruction Set Summary (Continued)

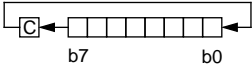
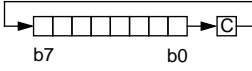
| Source Form | Operation | Description | Effect on CCR | | | | | Address Mode | Opcode | Operand | Cycles |
|--|---|---|---------------|---|----|---|---|---------------------------------------|----------------------------------|----------------------------------|----------------------------|
| | | | H | I | N | Z | C | | | | |
| ROL <i>opr</i> ROLA ROLX ROL <i>opr,X</i> ROL ,X | Rotate Byte Left through Carry Bit |  | — | — | ↕x | ↕ | ↕ | DIR INH INH IX1 IX | 39 49 59 69 79 | dd ff | 5 3 3 6 5 |
| ROR <i>opr</i> RORA RORX ROR <i>opr,X</i> ROR ,X | Rotate Byte Right through Carry Bit |  | — | — | ↕x | ↕ | ↕ | DIR INH INH IX1 IX | 36 46 56 66 76 | dd ff | 5 3 3 6 5 |
| RSP | Reset Stack Pointer | SP ← \$00FF | — | — | — | — | — | INH | 9C | | 2 |
| RTI | Return from Interrupt | SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL) | ↕x | ↕ | ↕ | ↕ | ↕ | INH | 80 | | 9 |
| RTS | Return from Subroutine | SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL) | — | — | — | — | — | INH | 81 | | 6 |
| SBC # <i>opr</i> SBC <i>opr</i> SBC <i>opr</i> SBC <i>opr,X</i> SBC <i>opr,X</i> SBC ,X | Subtract Memory Byte and Carry Bit from Accumulator | A ← (A) – (M) – (C) | — | — | ⊗ | ↕ | ↕ | IMM DIR EXT IX2 IX1 IX | A2 B2 C2 D2 E2 F2 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |
| SEC | Set Carry Bit | C ← 1 | — | — | — | — | 1 | INH | 99 | | 2 |
| SEI | Set Interrupt Mask | I ← 1 | — | 1 | — | — | — | INH | 9B | | 2 |
| STA <i>opr</i> STA <i>opr</i> STA <i>opr,X</i> STA <i>opr,X</i> STA ,X | Store Accumulator in Memory | M ← (A) | — | — | ↕x | ↕ | — | DIR EXT IX2 IX1 IX | B7 C7 D7 E7 F7 | dd hh ll ee ff ff | 4 5 6 5 4 |
| STOP | Stop Oscillator and Enable IRQ Pin | | — | 0 | — | — | — | INH | 8E | | 2 |
| STX <i>opr</i> STX <i>opr</i> STX <i>opr,X</i> STX <i>opr,X</i> STX ,X | Store Index Register In Memory | M ← (X) | — | — | ↕x | ↕ | — | DIR EXT IX2 IX1 IX | BF CF DF EF FF | dd hh ll ee ff ff | 4 5 6 5 4 |
| SUB # <i>opr</i> SUB <i>opr</i> SUB <i>opr</i> SUB <i>opr,X</i> SUB <i>opr,X</i> SUB ,X | Subtract Memory Byte from Accumulator | A ← (A) – (M) | — | — | ↕ | ↕ | ↕ | IMM DIR EXT IX2 IX1 IX | A0 B0 C0 D0 E0 F0 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 |

Table 6. Instruction Set Summary (Continued)

| Source Form | Operation | Description | Effect on CCR | | | | | Address Mode | Opcode | Operand | Cycles |
|---|--|--|---------------|----|---|---|---|--------------------------------|----------------------------|----------|-----------------------|
| | | | H | I | N | Z | C | | | | |
| SWI | Software Interrupt | PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte | — | 1 | — | — | — | INH | 83 | | 10 |
| TAX | Transfer Accumulator to Index Register | X ← (A) | — | — | — | — | — | INH | 97 | | 2 |
| TST <i>opr</i> TSTA TSTX TST <i>opr,X</i> TST <i>,X</i> | Test Memory Byte for Negative or Zero | (M) – \$00 | — | — | ↓ | ↓ | — | DIR INH INH IX1 IX | 3D 4D 5D 6D 7D | dd ff | 4 3 3 5 4 |
| TXA | Transfer Index Register to Accumulator | A ← (X) | — | — | — | — | — | INH | 9F | | 2 |
| WAIT | Stop CPU Clock and Enable Interrupts | | — | 0x | — | — | — | INH | 8F | | 2 |

- | | | | |
|----------|---|------------|--------------------------------------|
| A | Accumulator | <i>opr</i> | 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 | <i>rel</i> | 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 |
| <i>n</i> | Any bit | — | Not affected |

Table 7. Opcode Map

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

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 | 0 |
| LSB | BRSET0 5 DIR |
| 0 | 3 |

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

Resets and Interrupts

Contents

| | |
|--------------------------------|----|
| Resets | 56 |
| Power-On Reset | 56 |
| External Reset | 57 |
| COP Watchdog Reset | 58 |
| Low-Voltage Protection | 58 |
| Interrupts | 59 |
| Software Interrupt | 59 |
| External Interrupt | 59 |
| Timer Interrupts | 61 |
| Input Capture Interrupt | 61 |
| Output Compare Interrupt | 61 |
| Timer Overflow Interrupt | 62 |
| Interrupt Processing | 62 |

Resets

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

- Power-on reset (POR) circuit
- $\overline{\text{RESET}}$ pin
- COP watchdog

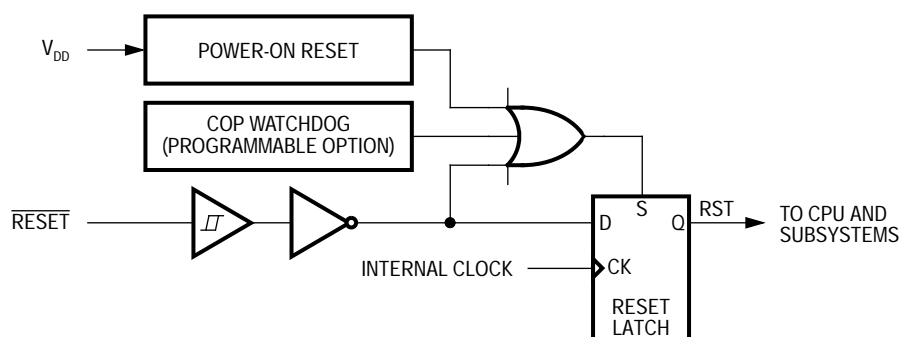


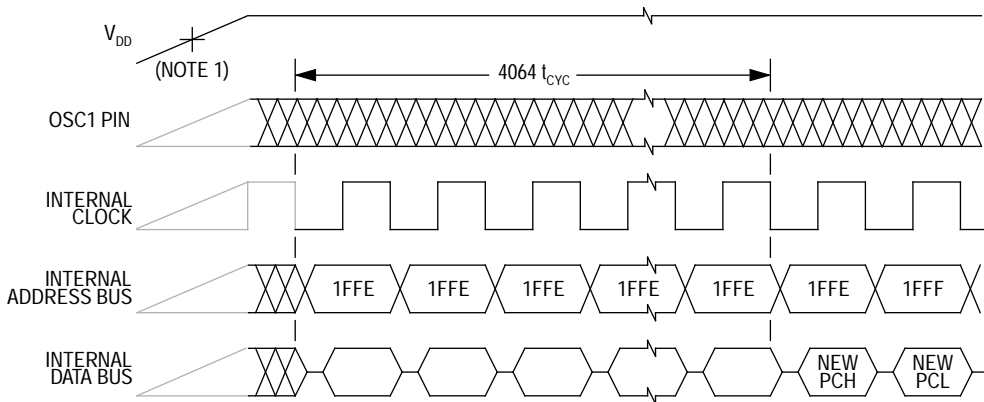
Figure 1. Reset Sources

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 the $\overline{\text{RESET}}$ pin is at logic zero at the end of $4064 t_{CYC}$, the MCU remains in the reset condition until the signal on the $\overline{\text{RESET}}$ pin goes to logic one.

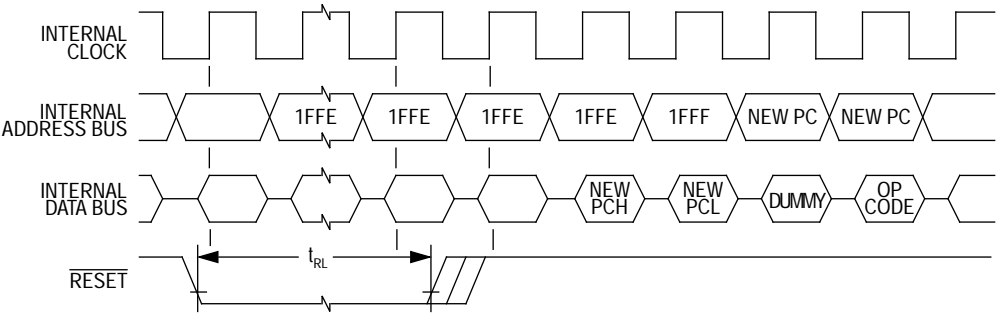


- NOTES:
1. Power-on reset threshold is typically between 1 V and 2 V.
 2. Internal clock, internal address bus, and internal data bus are not available externally.

Figure 2. Power-On Reset Timing

External Reset

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



- 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 RESET initiates the reset sequence.

Figure 3. External Reset Timing

Table 1. External Reset Timing

| Characteristic | Symbol | Min | Max | Unit |
|-------------------|-----------------|-----|-----|------------------|
| RESET Pulse Width | t _{RL} | 1.5 | — | t _{cyc} |

**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 zero to bit 0 (COPC) of the COP register at location \$1FF0.

Low-Voltage Protection

A drop in power supply voltage below the minimum operating V_{DD} voltage is called a brownout condition. A brownout while the MCU is in a non-reset state can corrupt MCU operation and necessitate a power-on reset to resume operation.

The best protection against brownout is an undervoltage sensing circuit that pulls the $\overline{\text{RESET}}$ pin low when it detects a low-power supply voltage. The undervoltage sensing circuit may be made of discrete components or an integrated circuit can be used.

For information about brownout and the COP watchdog, see the [Computer Operating Properly Watchdog](#) section.

Interrupts

The following sources can generate interrupts:

- SWI instruction
- $\overline{\text{IRQ}}/V_{\text{PP}}$ pin
- Capture/compare timer

An interrupt temporarily stops normal program execution 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.

Software Interrupt

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

External Interrupt

An interrupt signal on the $\overline{\text{IRQ}}/V_{\text{PP}}$ 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 $\overline{\text{IRQ}}/V_{\text{PP}}$ 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 4](#) shows the $\overline{\text{IRQ}}/V_{\text{PP}}$ pin interrupt logic.

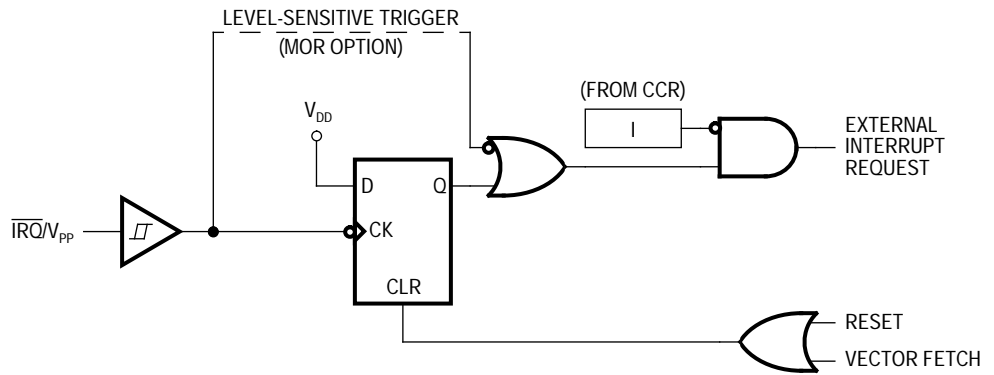


Figure 4. External Interrupt Logic

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

Interrupt triggering sensitivity of the $\overline{\text{IRQ}}/V_{\text{PP}}$ pin is a programmable option. The $\overline{\text{IRQ}}/V_{\text{PP}}$ pin can be negative-edge triggered or negative-edge- and low-level triggered. The level-sensitive triggering option allows multiple external interrupt sources to be wire-ORed to the $\overline{\text{IRQ}}/V_{\text{PP}}$ pin. An external interrupt request, shown in **Figure 5**, is latched as long as any source is holding the $\overline{\text{IRQ}}/V_{\text{PP}}$ pin low.

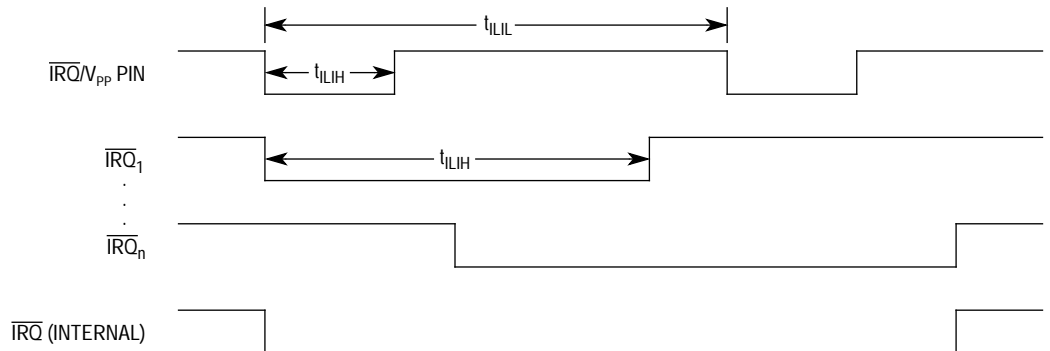


Figure 5. External Interrupt Timing

Table 2. External Interrupt Timing ($V_{DD} = 5.0 \text{ Vdc}$)⁽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} |

- $V_{DD} = 5.0 \text{ Vdc} \pm 10\%$, $V_{SS} = 0 \text{ Vdc}$, $T_A = T_L$ to T_H
- The minimum t_{ILIL} should not be less than the number of interrupt service routine cycles plus $19 t_{CYC}$.

Table 3. External Interrupt Timing ($V_{DD} = 3.3 \text{ Vdc}$)⁽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} |

- $V_{DD} = 3.3 \text{ Vdc} \pm 10\%$, $V_{SS} = 0 \text{ Vdc}$, $T_A = T_L$ to T_H
- The minimum t_{ILIL} should not be less than the number of interrupt service routine cycles plus $19 t_{CYC}$.

Timer Interrupts

The capture/compare timer can generate the following interrupts:

- Input capture interrupt
- Output compare interrupt
- Timer overflow interrupt

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

Input Capture Interrupt

An input capture interrupt request occurs if the input capture flag, ICF, becomes set while the input capture interrupt enable bit, ICIE, is also set. ICF is in the timer status register, and ICIE is in the timer control register.

Output Compare Interrupt

An output compare interrupt request occurs if the output compare flag, OCF, becomes set while the output compare interrupt enable bit, OCIE, is also set. OCF is in the timer status register, and OCIE is in the timer control register.

*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 is in the timer status register, and TOIE is in the timer control register.

**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 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:
 - \$1FFC and \$1FFD (software interrupt vector)
 - \$1FFA and \$1FFB (external interrupt vector)
 - \$1FF8 and \$1FF9 (timer interrupt vector)

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

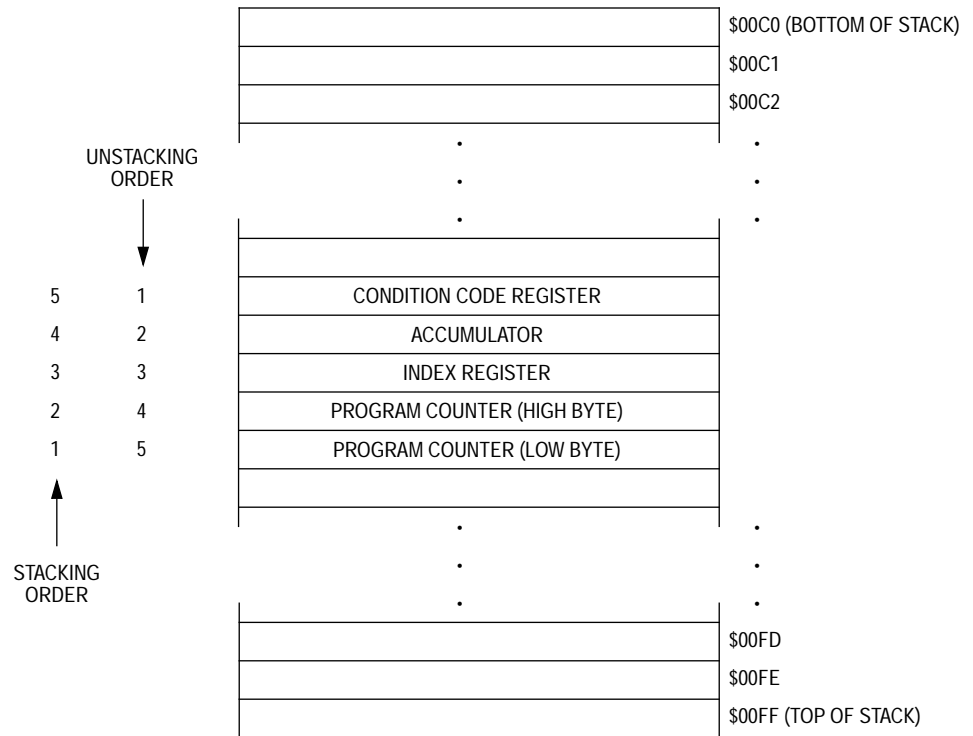


Figure 6. Interrupt Stacking Order

Table 4. Reset/Interrupt Vector Addresses

| Function | Source | Local Mask | Global Mask | Priority (1 = Highest) | Vector Address |
|--------------------------|--|----------------------------------|-------------|------------------------------|----------------|
| Reset | Power-On $\overline{\text{RESET}}$ Pin COP Watchdog ⁽¹⁾ | None | None | 1 | \$1FFE–\$1FFF |
| | | | None | 1 | |
| | | | None | 1 | |
| Software Interrupt (SWI) | User Code | None | None | Same Priority as Instruction | \$1FFC–\$1FFD |
| External Interrupt | $\overline{\text{IRQ}}/V_{PP}$ Pin | None | I Bit | 2 | \$1FFA–\$1FFB |
| Timer Interrupts | ICF Bit OCF Bit TOF Bit | ICIE Bit OCIE Bit TOIE Bit | I Bit | 3 | \$1FF8–\$1FF9 |

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

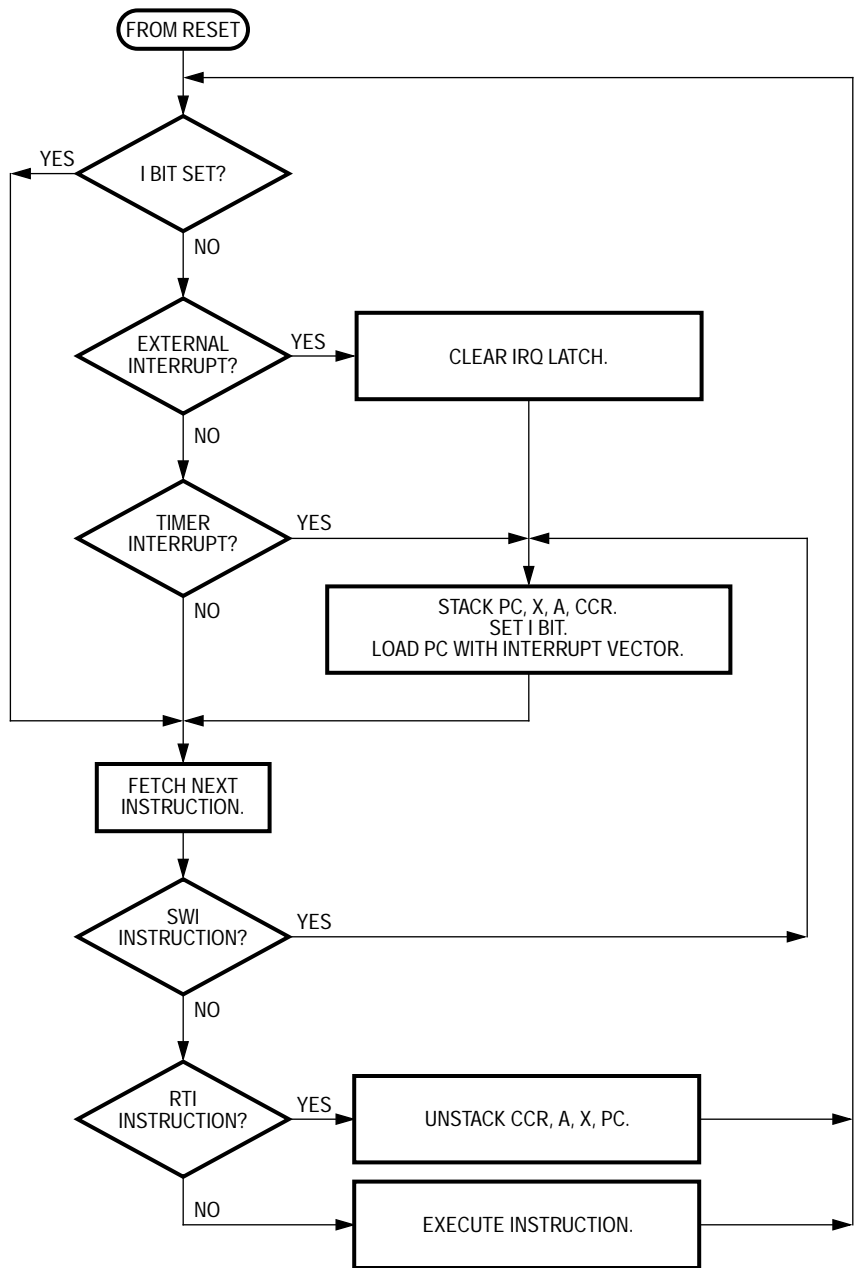


Figure 7. Interrupt Flowchart

Contents

| | |
|-------------------------------|----|
| Stop Mode | 65 |
| Wait Mode | 68 |
| Data-Retention Mode | 70 |

Stop Mode

The STOP instruction puts the MCU in its lowest power-consumption mode and has the following effects on the MCU:

- Stops the internal oscillator, the CPU clock, and the internal clock, turning off the capture/compare timer, the COP watchdog, the SIOF, and the ADC
- Clears the I bit in the condition code register, enabling external interrupts
- Clears the ICIE, OCIE, and TOIE bits in the timer control register, disabling further timer interrupts

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

The following events bring the MCU out of stop mode:

- An external interrupt signal on the $\overline{\text{IRQ}}/V_{\text{PP}}$ pin — A high-to-low transition on the $\overline{\text{IRQ}}/V_{\text{PP}}$ pin loads the program counter with the contents of locations \$1FFA and \$1FFB. The timer resumes counting from the last value before the STOP instruction.
- External reset — A logic zero on the $\overline{\text{RESET}}$ pin resets the MCU and loads the program counter with the contents of locations \$1FFE and \$1FFF. The timer begins counting from \$FFFC.

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

An active edge on the PD7/TCAP pin during stop mode sets the ICF flag when an external interrupt brings the MCU out of stop mode. An external interrupt also latches the value in the timer registers into the input capture registers.

If a reset brings the MCU out of stop mode, then an active edge on the PD7/TCAP pin during stop mode has no effect on the ICF flag or the input capture registers.

See **Figure 1** for stop recovery timing information.

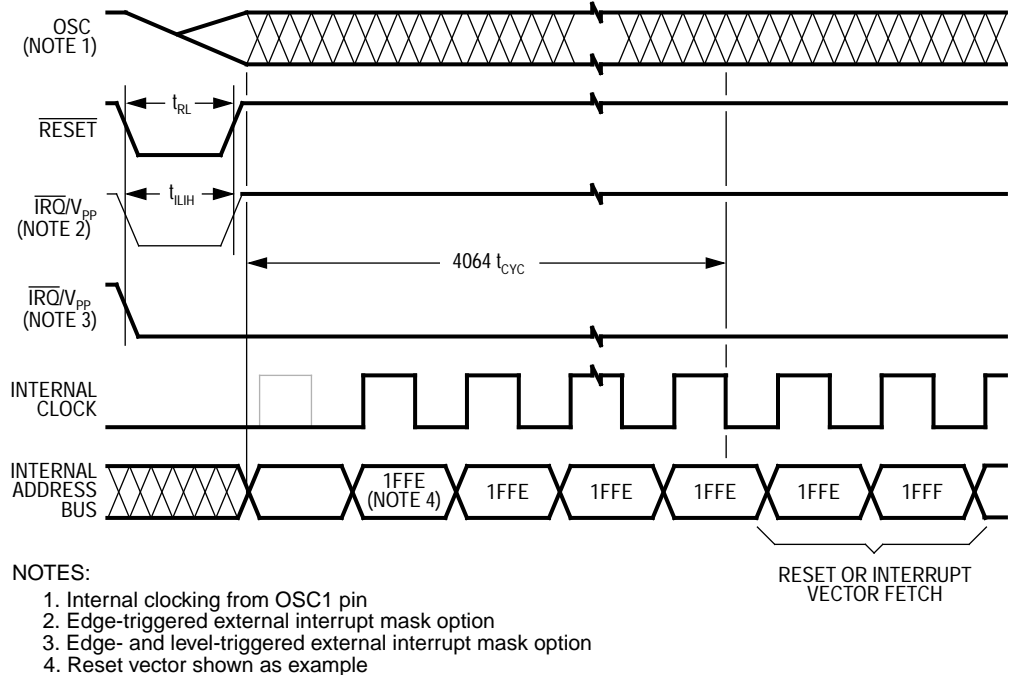


Figure 1. Stop Recovery Timing

Figure 2 shows the sequence of events caused by the STOP instruction.

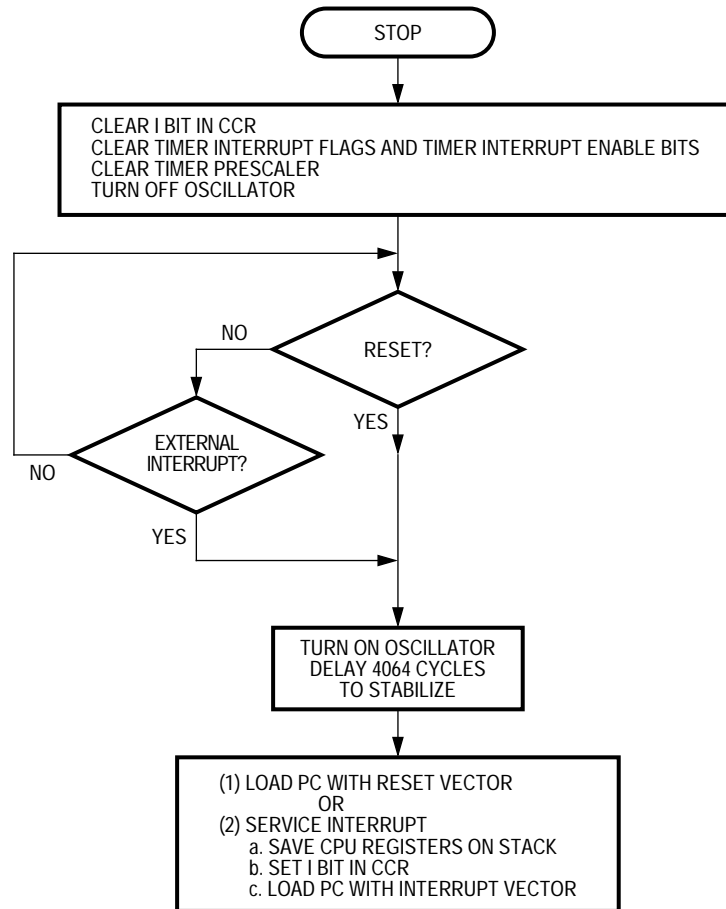


Figure 2. STOP Instruction Flowchart

Wait Mode

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

- Clears the I bit in the condition code register, enabling interrupts
- Stops the CPU clock, but allows the internal clock to drive the capture/compare timer, the COP watchdog, and the ADC

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

The following conditions restart the CPU clock and bring the MCU out of wait mode:

- External interrupt — A high-to-low transition on the $\overline{\text{IRQ}}/V_{PP}$ pin loads the program counter with the contents of locations \$1FFA and \$1FFB.
- Timer interrupt — Input capture, output compare, and timer overflow interrupt requests load the program counter with the contents of locations \$1FF8 and \$1FF9.
- COP watchdog reset — A timeout of the COP watchdog resets the MCU and loads the program counter with the contents of locations \$1FFE and \$1FFF. Software can enable timer interrupts so that the MCU can periodically exit wait mode to reset the COP watchdog.
- External reset — A logic zero on the $\overline{\text{RESET}}$ pin resets the MCU and loads the program counter with the contents of locations \$1FFE and \$1FFF.

Figure 3 shows the sequence of events caused by the WAIT instruction.

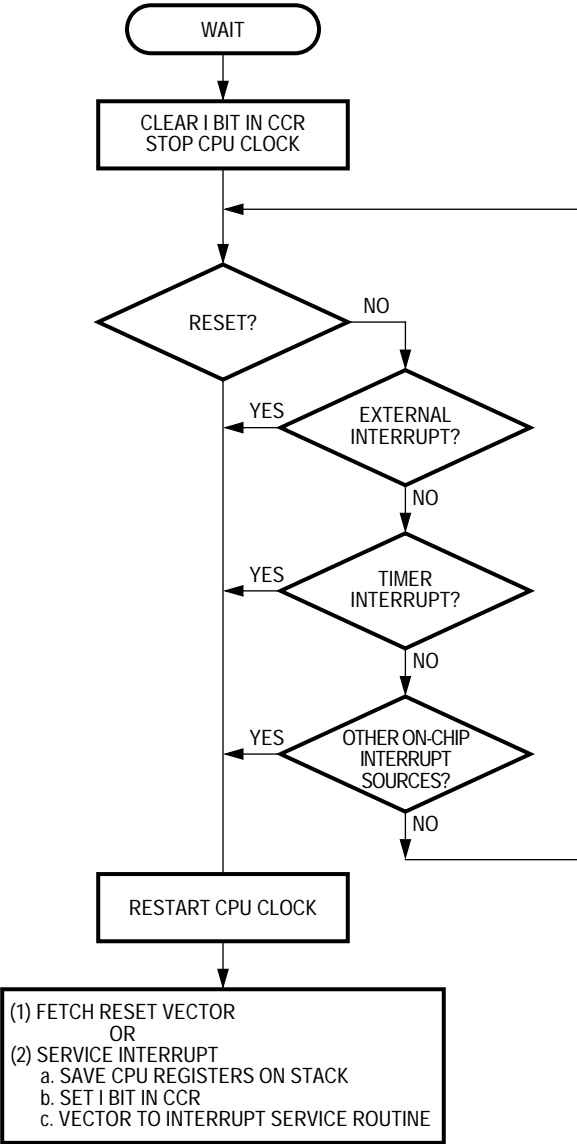


Figure 3. WAIT Instruction Flowchart

Figure 4 shows the effect of the STOP and WAIT instructions on the CPU clock and the timer clock.

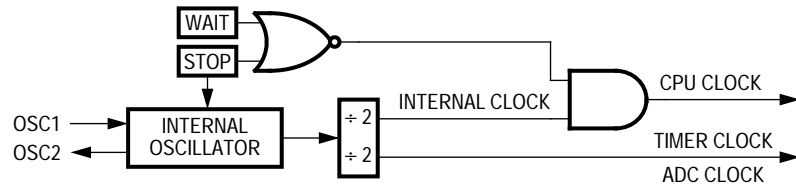


Figure 4. STOP/WAIT Clock Logic

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 zero.
2. Lower the V_{DD} voltage. The $\overline{\text{RESET}}$ pin must remain low continuously during data-retention mode.

To take the MCU out of data-retention mode:

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

Contents

| | |
|--|----|
| Introduction | 72 |
| Port A | 73 |
| Port A Data Register (PORTA) | 73 |
| Data Direction Register A (DDRA) | 74 |
| Port B | 76 |
| Port B Data Register (PORTB) | 76 |
| Data Direction Register B (DDRB) | 77 |
| Port C | 79 |
| Port C Data Register (PORTC) | 79 |
| Data Direction Register C (DDRC) | 80 |
| Port D | 82 |
| Port D Data Register (PORTD) | 82 |
| Data Direction Register D (DDRD) | 83 |

Introduction

Twenty bidirectional pins and one input-only pin form four parallel input/output (I/O) ports. 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. | Name: | R/W | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|----------------------------------|---------------------------|---------------------|-------|-------|-------|-------|-------|-------|-------|
| \$0000 | Port A Data Register (PORTA) | Read: Write: Reset: | PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| | | | Unaffected by reset | | | | | | | |
| \$0001 | Port B Data Register (PORTB) | Read: Write: Reset: | PB7 | PB6 | PB5 | 0 | 0 | 0 | 0 | 0 |
| | | | Unaffected by reset | | | | | | | |
| \$0002 | Port C Data Register (PORTC) | Read: Write: Reset: | PC7 | PC6 | PC5 | PC4 | PC3 | PC2 | PC1 | PC0 |
| | | | Unaffected by reset | | | | | | | |
| \$0003 | Port D Data Register (PORTD) | Read: Write: Reset: | PD7 | 0 | PD5 | 1 | 0 | 0 | 0 | 0 |
| | | | Unaffected by reset | | | | | | | |
| \$0004 | Data Direction Register A (DDRA) | Read: Write: Reset: | DDRA7 | DDRA6 | DDRA5 | DDRA4 | DDRA3 | DDRA2 | DDRA1 | DDRA0 |
| | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0005 | Data Direction Register B (DDRB) | Read: Write: Reset: | DDRB7 | DDRB6 | DDRB5 | 0 | 0 | 0 | 0 | 0 |
| | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented

Figure 1. Parallel I/O Port Register Summary

| Addr. | Name: | R/W | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-----------------------------------|--------|-------|-------|--------|-------|-------|-------|-------|-------|
| \$0006 | Data Direction Register C (DDRC) | Read: | DDRC7 | DDRC6 | DDRC5 | DDRC4 | DDRC3 | DDRC2 | DDRC1 | DDRC0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0007 | Data Direction Register D (DDR D) | Read: | 0 | 0 | DDR D5 | 0 | 0 | 0 | 0 | 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented

Figure 1. Parallel I/O Port Register Summary (Continued)

Port A

Port A is an 8-bit general-purpose I/O port.

Port A Data Register (PORTA)

The port A data register contains a latch for each of the eight port A pins.

| \$0000 | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|---------------------|-----|-----|-----|-----|-----|-----|-------|
| Read: | PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| Write: | | | | | | | | |
| Reset: | Unaffected by reset | | | | | | | |

Figure 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.

Data Direction Register A (DDRA)

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

| | | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| \$0004 | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Read: | DDRA7 | DDRA6 | DDRA5 | DDRA4 | DDRA3 | DDRA2 | DDRA1 | DDRA0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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

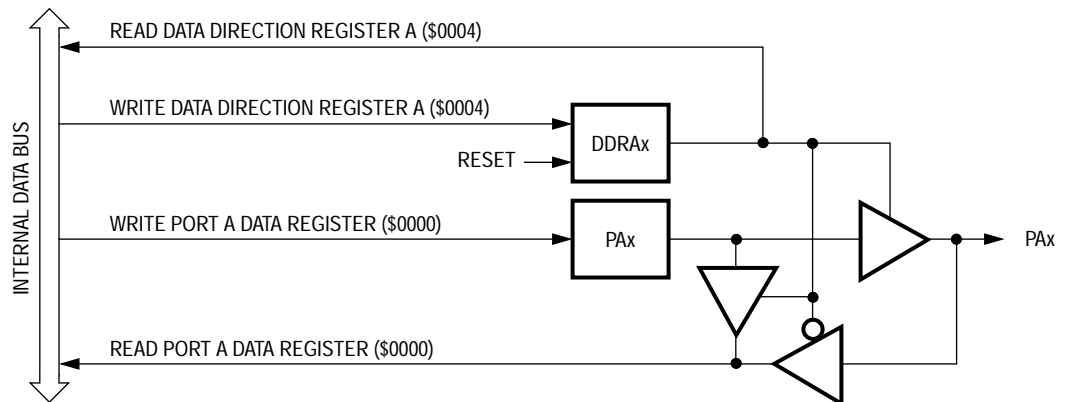


Figure 4. Port A I/O Circuit

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

When bit DDRAx is a logic one, reading address \$0000 reads the PAX data latch. When bit DDRAx is a logic zero, 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 1** summarizes the operation of the port A pins.

Table 1. Port A Pin Operation

| Data Direction Bit | I/O Pin Mode | Accesses to Data Bit | |
|--------------------|----------------------------|----------------------|----------------------|
| | | Read | Write |
| 0 | Input, Hi-Z ⁽¹⁾ | Pin | Latch ⁽²⁾ |
| 1 | Output | Latch | Latch |

1. Hi-Z = high impedance

2. Writing affects data register, but does not affect input.

Port B

Port B is a 3-bit I/O port that shares its pins with the serial I/O port (SIOP).

NOTE: Do not use port B for general-purpose I/O while the SIOP is enabled.

Port B Data Register (PORTB)

The port B data register contains a latch for each of the three port B pins.

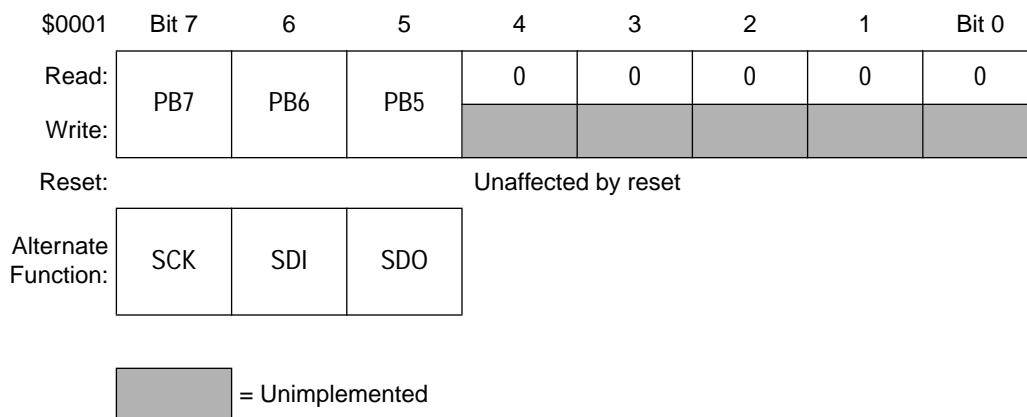


Figure 5. Port B Data Register (PORTB)

PB[7:5] — Port B Data Bits

These read/write bits are software programmable bits. 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: Writing to data direction register B does not affect the data direction of port B pins that are being used by the SIOP. However, data direction register B always determines whether reading port B returns the states of the latches or the states of the pins.

SCK — Serial Clock

When the SIOP is enabled, SCK is the SIOP clock output (in master mode) or the SIOP clock input (in slave mode).

SDI — Serial Data Input

When the SIOP is enabled, SDI is the SIOP data input.

SDO — Serial Data Output

When the SIOP is enabled, SDO is the SIOP data output.

Data Direction Register B (DDRB)

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

NOTE: *Enabling and then disabling the SIOP configures data direction register B for SIOP operation and can also change the port B data register. After disabling the SIOP, initialize data direction register B and the port B data register as your application requires.*

| \$0005 | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-------|-------|---|---|---|---|-------|
| Read: | DDRB7 | DDRB6 | DDRB5 | 0 | 0 | 0 | 0 | 0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |


 = Unimplemented

Figure 6. Data Direction Register B (DDRB)

DDRB[7:5] — Data Direction Register B Bits

These read/write bits control port B data direction. Reset clears DDRB[7:5], configuring all three 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 shows the I/O logic of port B.

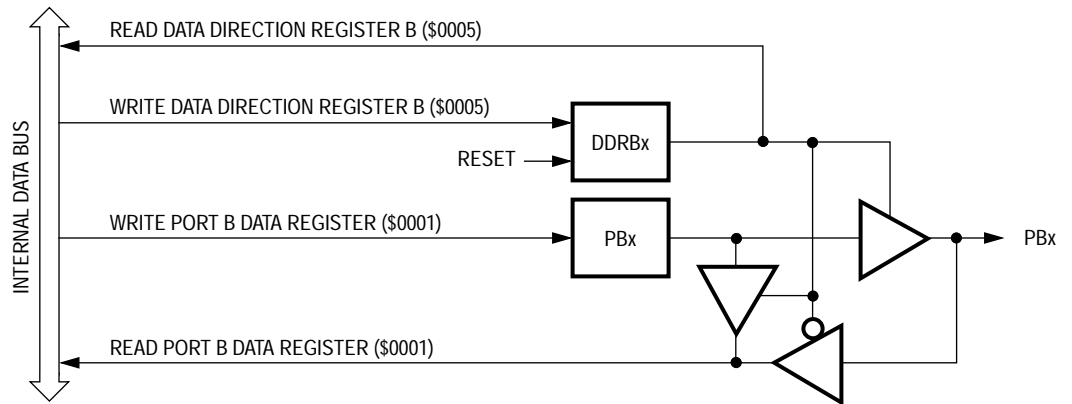


Figure 7. Port B I/O Logic

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

When bit DDR_{Bx} is a logic one, reading address \$0001 reads the PB_x data latch. When bit DDR_{Bx} is a logic zero, 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 1](#) summarizes the operation of the port B pins.

Table 2. Port B Pin Operation

| Data Direction Bit | I/O Pin Mode | Accesses to Data Bit | |
|--------------------|----------------------------|----------------------|----------------------|
| | | Read | Write |
| 0 | Input, Hi-Z ⁽¹⁾ | Pin | Latch ⁽²⁾ |
| 1 | Output | Latch | Latch |

1. Hi-Z = high impedance

2. Writing affects data register, but does not affect input.

Port C

Port C is an 8-bit I/O port that shares five of its pins with the A/D converter (ADC). The five shared pins are available for general-purpose I/O functions when the ADC is disabled.

Port C Data Register (PORTC)

The port C data register contains a latch for each of the eight port C pins.

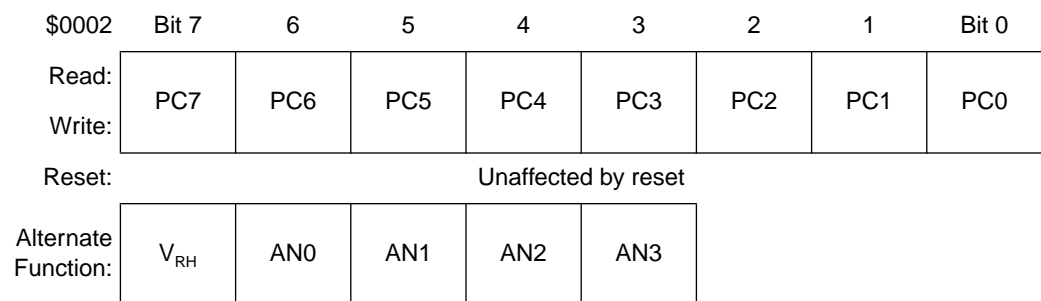


Figure 8. Port C Data Register (PORTC)

PC[7:0] — Port C Data Bits

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

V_{RH} — Voltage Reference High Bit

When the ADC is turned on, the PC7/ V_{RH} pin is the positive ADC reference voltage.

AN[3:0] — Analog Input Bits

When the ADC is turned on, the AN0–AN3 pins are software-selectable analog inputs. Unused analog inputs can be used as digital inputs, but pins PC3/AN3, PC4/AN2, PC5/AN1, and PC6/AN0 cannot be used as digital outputs while the ADC is on. Only pins PC0, PC1, and PC2 can be used as digital outputs when the ADC is on.

The port C data register reads normally while the ADC is on, except that the bit corresponding to the currently selected ADC input pin reads as logic zero.

Writing to bits PC7–PC3 while the ADC is on can produce unpredictable ADC results.

Data Direction Register C (DDRC)

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

| \$0006 | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| Read: | DDRC7 | DDRC6 | DDRC5 | DDRC4 | DDRC3 | DDRC2 | DDRC1 | DDRC0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 9. Data Direction Register C (DDRC)

DDRC[7:0] — Data Direction Register C Bits

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

1 = Corresponding port C pin configured as output

0 = Corresponding port C pin configured as input

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

Writing to bits DDRC7–DDRC3 while the ADC is on can produce unpredictable ADC results.

Figure 10 shows the I/O logic of port C.

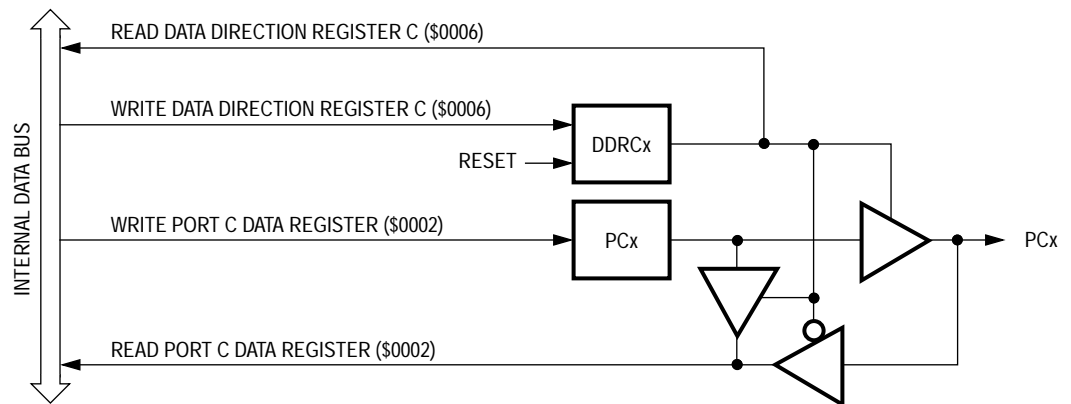


Figure 10. Port C I/O Logic

Writing a logic one to a DDRC bit enables the output buffer for the corresponding port C pin; a logic zero disables the output buffer.

When bit DDRCx is a logic one, reading address \$0002 reads the PCx data latch. When bit DDRCx is a logic zero, reading address \$0002 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. [Table 1](#) summarizes the operation of the port C pins.

Table 3. Port C Pin Operation

| Data Direction Bit | I/O Pin Mode | Accesses to Data Bit | |
|--------------------|----------------------------|----------------------|----------------------|
| | | Read | Write |
| 0 | Input, Hi-Z ⁽¹⁾ | Pin | Latch ⁽²⁾ |
| 1 | Output | Latch | Latch |

1. Hi-Z = high impedance

2. Writing affects data register, but does not affect input.

Port D

Port D is a 2-bit port with one I/O pin and one input-only pin. Port D shares the input-only pin, PD7/TCAP, with the capture/compare timer. PD7/TCAP is the timer input capture pin. The PD7/TCAP pin can always be a general-purpose input, even if input capture interrupts are enabled.

Port D Data Register (PORTD)

The port D data register contains a latch for each of the two port D pins.

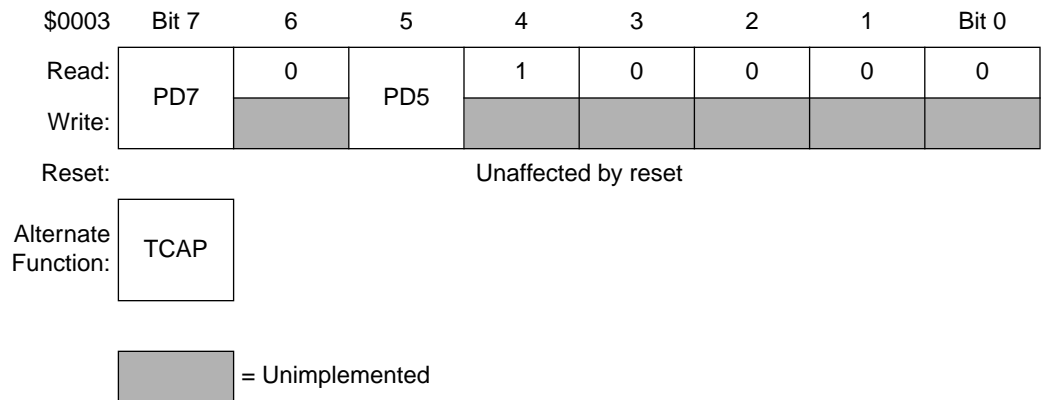


Figure 11. Port D Data Register (PORTD)

PD7 and PD5 — Port D Data Bits

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

TCAP — Timer Capture

TCAP is the input capture pin for the timer.

Data Direction Register D (DDRD)

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

| \$0007 | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|---|-------|---|---|---|---|-------|
| Read: | 0 | 0 | DDRD5 | 0 | 0 | 0 | 0 | 0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |


 = Unimplemented

Figure 12. Data Direction Register D (DDRD)

DDRD5 — Data Direction Register D Bit

This read/write bit controls the data direction of pin PD5. Reset clears DDRD5, configuring PD5 as an input.

- 1 = PD5 configured as output
- 0 = PD5 configured as input

NOTE: Avoid glitches on port D pins by writing to the port D data register before changing data direction register D bits from 0 to 1.

Figure 13 shows the I/O logic of port D.

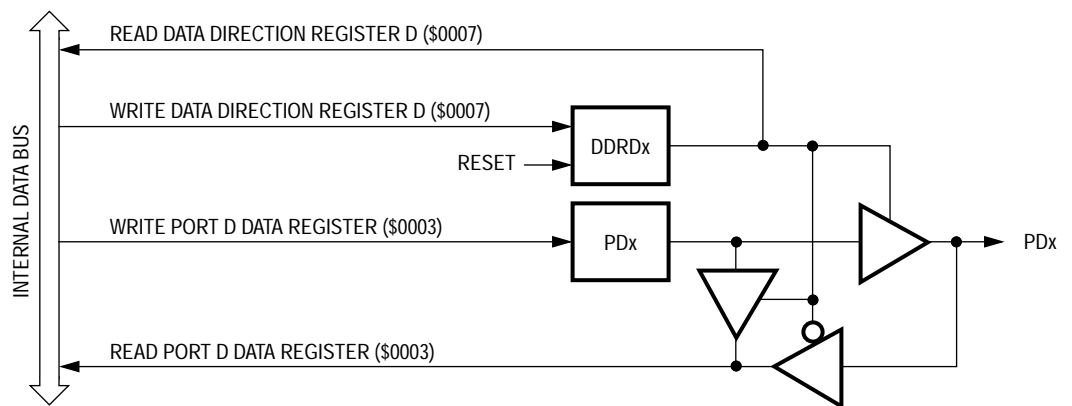


Figure 13. Port D I/O Logic

Writing a logic one to a DDRD bit enables the output buffer for the corresponding port D pin; a logic zero disables the output buffer.

When bit DDRDx is a logic one, reading address \$0003 reads the PDx data latch. When bit DDRDx is a logic zero, reading address \$0003 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. [Table 1](#) summarizes the operation of the port D pins.

Table 4. Port D Pin Operation

| Data Direction Bit | I/O Pin Mode | Accesses to Data Bit | |
|--------------------|----------------------------|----------------------|----------------------|
| | | Read | Write |
| 0 | Input, Hi-Z ⁽¹⁾ | Pin | Latch ⁽²⁾ |
| 1 | Output | Latch | Latch |

1. Hi-Z = high impedance

2. Writing affects data register, but does not affect input.

Computer Operating Properly Watchdog COP

Contents

| | |
|-----------------------------------|----|
| Features | 85 |
| Introduction | 86 |
| Operation | 86 |
| COP Watchdog Timeout | 86 |
| COP Watchdog Timeout Period | 86 |
| Clearing the COP Watchdog | 87 |
| Interrupts | 87 |
| COP Register | 87 |
| Low-Power Modes | 88 |
| Stop Mode | 88 |
| Wait Mode | 88 |

Features

- Protection from runaway software
- 65.5-ms timeout period (with 2-MHz bus frequency)
- Wait mode operation

Introduction

The purpose of the computer operating properly (COP) watchdog is to reset the MCU in case of software failure. Software that is operating properly periodically services the COP watchdog and prevents the reset from occurring. The COP watchdog function is programmable in the mask option register.

Operation

COP Watchdog Timeout

The COP watchdog is a 16-bit counter that generates a reset if allowed to time out. Periodically clearing the counter starts a new timeout period and prevents the COP from resetting the MCU. 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 also depends on a power supply voltage at or above a minimum specification and is not guaranteed to protect against brownout. For information about brownout protection, see the [Resets and Interrupts](#) section.

COP Watchdog Timeout Period

Use the following formula to calculate the COP timeout period:

$$\text{COP Timeout Period} = \frac{131,072 \text{ cycles}}{f_{\text{BUS}}}$$

where

$$f_{\text{BUS}} = \frac{\text{crystal frequency}}{2}$$

Clearing the COP Watchdog

To clear the COP watchdog and prevent a COP reset, write a logic zero to bit 0 (COPC) of the COP register at location \$1FF0.

If the main program executes within the COP timeout period, the clearing routine needs to 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.

Interrupts

The COP watchdog does not generate interrupts.

COP Register

The COP register is a write-only register that returns the contents of EPROM location \$1FF0 when read.

| \$1FF0 | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|----|----|----|----|----|----|-------|
| Read: | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| Write: | | | | | | | | COPC |
| Reset: | U | U | U | U | U | U | U | 0 |

= Unimplemented
 U = Unaffected

Figure 1. COP Register (COPR)

COPC — COP Clear

COPC is a write-only bit. Periodically writing a logic zero to COPC prevents the COP watchdog from resetting the MCU. Reset clears the COPC bit.

Low-Power Modes

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

Stop Mode

The STOP instruction clears the COP watchdog counter. Upon exit from stop mode by external reset:

- The counter begins counting from \$0000.
- The counter is cleared again after the 4064-cycle oscillator stabilization delay.

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 has a count of 4064 when the program resumes.

Wait Mode

The COP watchdog continues to operate normally after a WAIT instruction. Software should periodically take the MCU out of wait mode and write to the COPC bit to prevent a COP watchdog timeout.

Contents

| | |
|---------------------------------|-----|
| Features | 90 |
| Introduction | 90 |
| Operation | 93 |
| Pin Functions | 93 |
| PD7/TCAP | 93 |
| TCMP | 93 |
| Input Capture | 93 |
| Output Compare | 94 |
| Timing | 95 |
| Interrupts | 98 |
| I/O Registers | 98 |
| Timer Control Register | 99 |
| Timer Status Register | 100 |
| Timer Registers | 102 |
| Alternate Timer Registers | 103 |
| Input Capture Registers | 104 |
| Output Compare Registers | 105 |
| Low-Power Modes | 106 |
| Stop Mode | 106 |
| Wait Mode | 106 |

Features

- Programmable Polarity of Input Capture Edge
- Programmable Polarity of Output Compare Signal
- Alternate Counter Registers
- 16-Bit Counter
- Interrupt-Driven Operation with Three Maskable Interrupt Flags:
 - Input Capture
 - Output Compare
 - Timer Overflow

Introduction

The timer provides a timing reference for MCU operations. The input capture and output compare functions provide a means to latch the times at which external events occur, to measure input waveforms, and to generate output waveforms and timing delays. [Figure 1](#) shows the structure of the timer module.

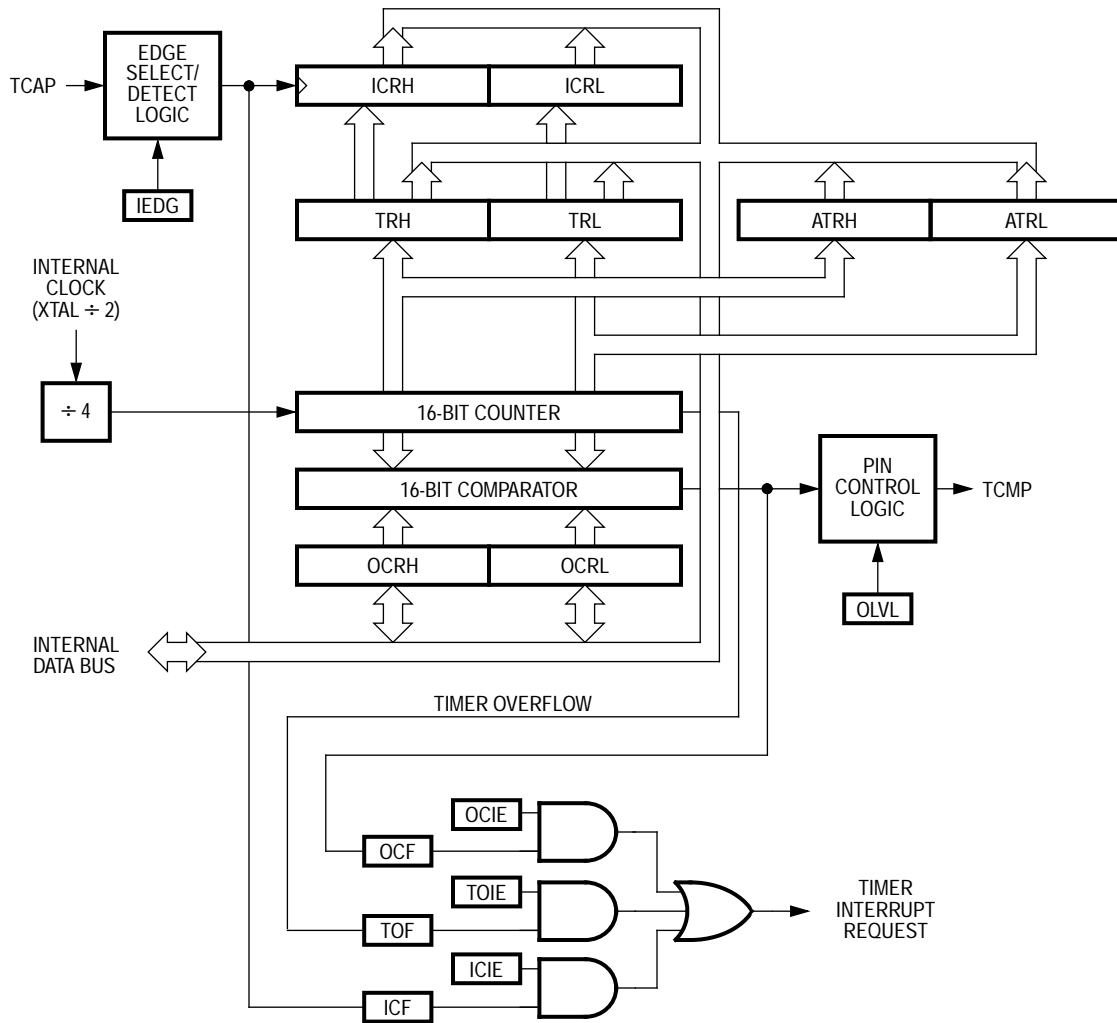


Figure 1. Timer Block Diagram

| Addr. | Name | R/W | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|--------------------------------------|---------------------------|--------|------|------|----|----|----|------|-------|
| \$0012 | Timer Control Register (TCR) | Read: Write: Reset: | ICIE | OCIE | TOIE | 0 | 0 | 0 | IEDG | OLVL |
| \$0013 | Timer Status Register (TSR) | Read: Write: Reset: | ICF | OCF | TOF | 0 | 0 | 0 | 0 | 0 |
| \$0014 | Input Capture Register High (ICRH) | Read: Write: Reset: | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit 8 |
| \$0015 | Input Capture Register Low (ICRL) | Read: Write: Reset: | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| \$0016 | Output Compare Register High (OCRH) | Read: Write: Reset: | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit 8 |
| \$0017 | Output Compare Register Low (OCRL) | Read: Write: Reset: | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| \$0018 | Timer Register High (TRH) | Read: Write: Reset: | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit 8 |
| \$0019 | Timer Register Low (TRL) | Read: Write: Reset: | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| \$001A | Alternate Timer Register High (ATRH) | Read: Write: Reset: | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit 8 |
| \$001B | Alternate Timer Register Low (ATRL) | Read: Write: Reset: | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |

 = Unimplemented

U = Unaffected

Figure 2. Timer I/O Register Summary

Operation

The timing reference for the input capture and output compare functions is a 16-bit free-running counter. The counter is preceded by a divide-by-four prescaler and rolls over every 2^{18} cycles. Timer resolution with a 4-MHz crystal is 2 μ s. Software can read the value in the counter at any time without affecting the counter sequence.

Because of the 16-bit timer architecture, the I/O registers for the input capture and output compare functions are pairs of 8-bit registers.

Pin Functions

The timer uses two pins.

PD7/TCAP

PD7/TCAP is the input capture pin. When an active edge occurs on PD7/TCAP, the timer transfers the current counter value to the input capture registers. PD7/TCAP is also an I/O port pin.

TCMP

TCMP is the output-only output compare pin. When the counter value matches the value written in the output compare registers, the timer transfers the output level bit, OLVL, to the TCMP pin.

Input Capture

The input capture function is a means to record the time at which an external event occurs. When the input capture circuitry detects an active edge on the PD7/TCAP pin, it latches the contents of the timer registers into the input capture registers. The polarity of the active edge is programmable.

Latching values into the input capture registers at successive edges of the same polarity measures the period of the input signal on the PD7/TCAP pin. Latching the counter values at successive edges of opposite polarity measures the pulse width of the signal. **Figure 3** shows the logic of the input capture function.

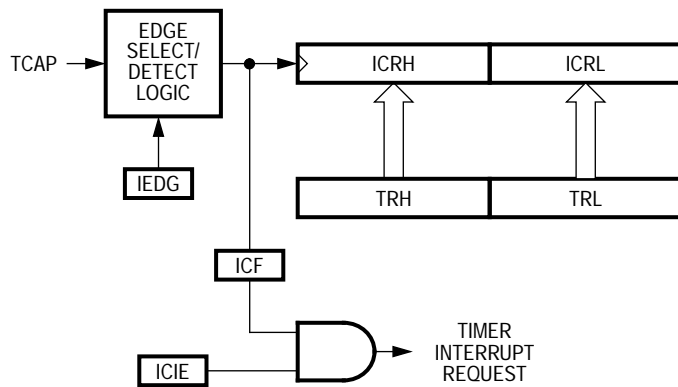


Figure 3. Input Capture Operation

Output Compare

The output compare function is a means of generating an output signal when the 16-bit counter reaches a selected value. Software writes the selected value into the output compare registers. On every fourth internal clock cycle the output compare circuitry compares the value of the counter to the value written in the output compare registers. When a match occurs, the timer transfers the programmable output level bit (OLVL) from the timer control register to the TCMP pin.

Software can use the output compare register to measure time periods, to generate timing delays, or to generate a pulse of specific duration or a pulse train of specific frequency and duty cycle on the TCMP pin.

Figure 4 shows the logic of the output compare function.

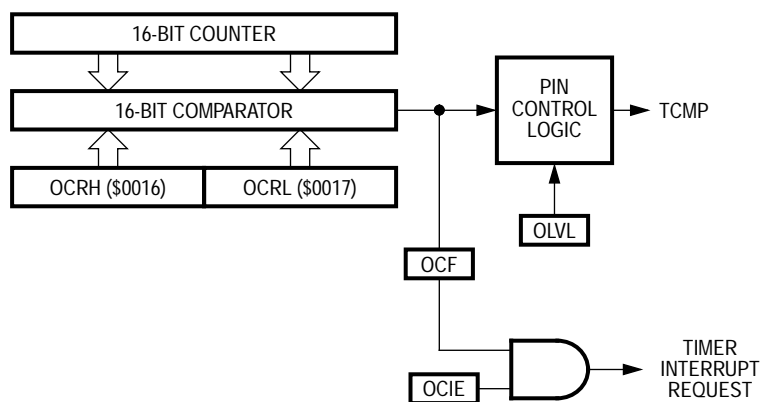


Figure 4. Output Compare Operation

Timing

Table 1. Timer Characteristics ($V_{DD} = 5.0 \text{ Vdc}$)⁽¹⁾

| Characteristic | Symbol | Min | Max | Unit |
|---------------------------------|------------|----------|-----|-----------|
| Timer Resolution ⁽²⁾ | t_{RESL} | 4.0 | — | t_{CYC} |
| Input Capture Pulse Width | t_H, t_L | 125 | — | ns |
| Input Capture Pulse Period | t_{TLTL} | Note (3) | — | t_{CYC} |

1. $V_{DD} = 5.0 \text{ Vdc} \pm 10\%$, $T_A = T_L$ to T_H unless otherwise noted.
2. A 2-bit prescaler in the timer is the limiting factor as it counts 4 t_{CYC} .
3. The minimum t_{TLTL} should not be less than the number of interrupt service routine cycles plus 19 t_{CYC} .

Table 2. Timer Characteristics ($V_{DD} = 3.3 \text{ Vdc}$)⁽¹⁾

| Characteristic | Symbol | Min | Max | Unit |
|---------------------------------|------------|----------|-----|-----------|
| Timer Resolution ⁽²⁾ | t_{RESL} | 4.0 | — | t_{CYC} |
| Input Capture Pulse Width | t_H, t_L | 250 | — | ns |
| Input Capture Pulse Period | t_{TLTL} | Note (3) | — | t_{CYC} |

1. $V_{DD} = 3.3 \text{ Vdc} \pm 10\%$, $T_A = T_L$ to T_H unless otherwise noted.
2. A 2-bit prescaler in the timer is the limiting factor as it counts 4 t_{CYC} .
3. The minimum t_{TLTL} should not be less than the number of interrupt service routine cycles plus 19 t_{CYC} .

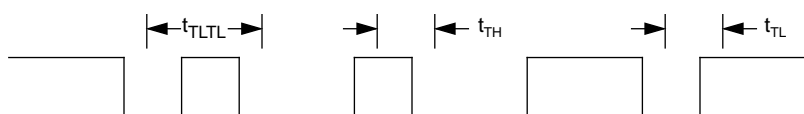


Figure 5. Input Capture Characteristics

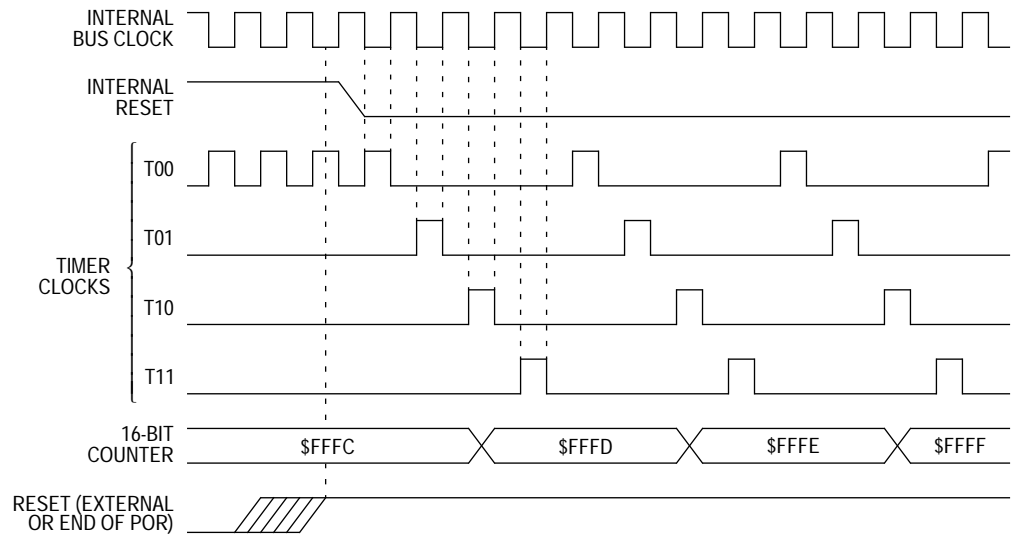
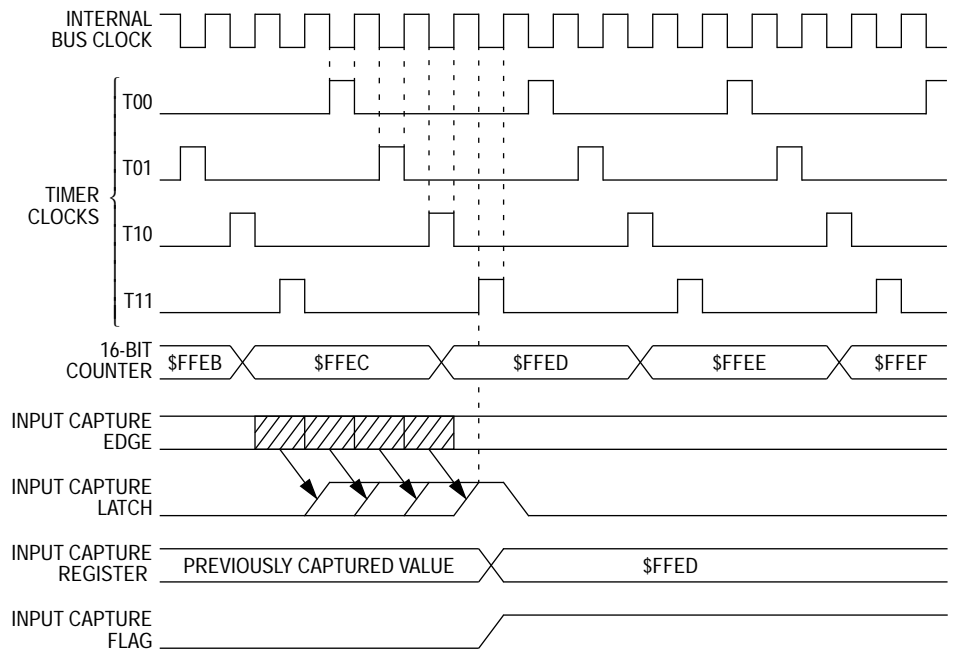


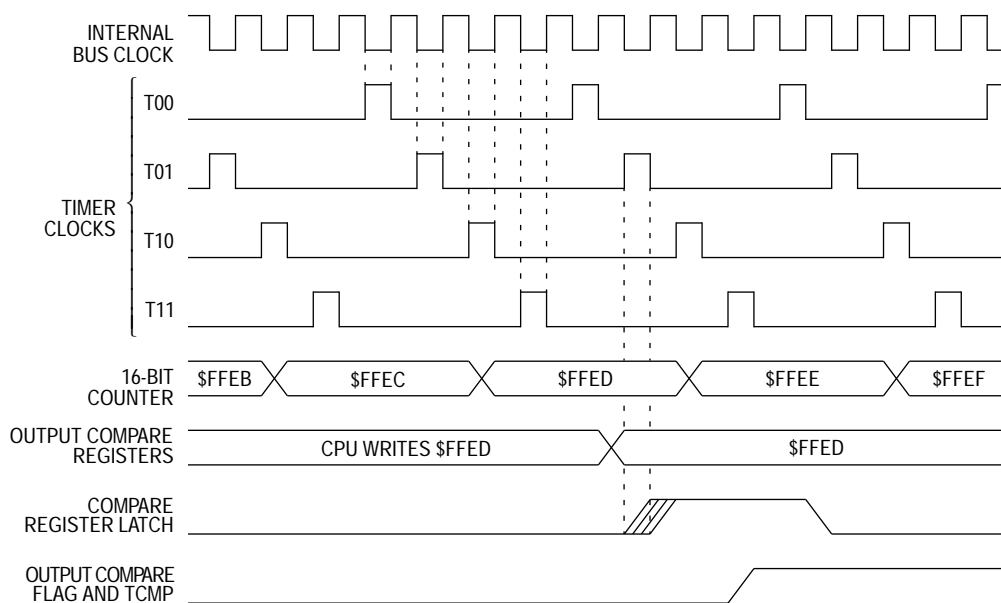
Figure 6. Timer Reset Timing



NOTE:

If the input capture edge occurs in the shaded area between T10 states, then the input capture flag becomes set during the next T11 state.

Figure 7. Input Capture Timing



NOTES:

1. A write to the output compare registers may occur at any time, but a compare only occurs at timer state T01. Therefore, the compare may follow the write by up to four cycles.
2. The output compare flag is set at the timer state T11 that follows the comparison latch.

Figure 8. Output Compare Timing

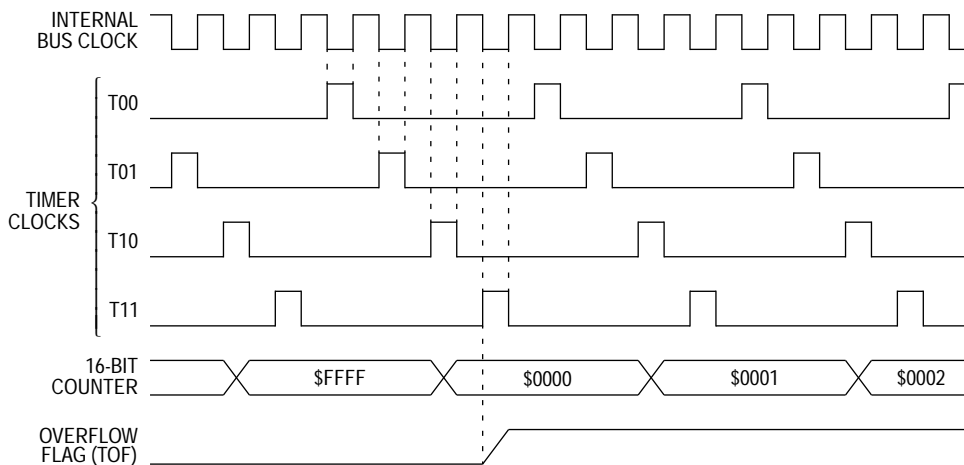


Figure 9. Timer Overflow Timing

Interrupts

The following timer sources can generate interrupts:

- Input capture flag (ICF) — The ICF bit is set when an edge of the selected polarity occurs on the input capture pin. The input capture interrupt enable bit, ICIE, enables ICF interrupt requests.
- Output compare flag (OCF) — The OCF bit is set when the counter value matches the value written in the output compare registers. The output compare interrupt enable bit, OCIE, enables OCF interrupt requests.
- Timer overflow flag (TOF) — The TOF bit is set when the counter value rolls over from \$FFFF to \$0000. The timer overflow enable bit (TOIE) enables timer overflow interrupt requests.

Table 3 summarizes the timer interrupt sources.

Table 3. Timer Interrupt Sources

| Source | Local Mask | Global Mask | Priority (1 = Highest) |
|-------------------------------|----------------------------------|-------------|------------------------|
| ICF Bit OCF Bit TOF Bit | ICIE Bit OCIE Bit TOIE Bit | I Bit | 3 |

I/O Registers

The following registers control and monitor the operation of the timer:

- Timer control register (TCR)
- Timer status register (TSR)
- Timer registers (TRH and TRL)
- Alternate timer registers (ATRH and ATRL)
- Input capture registers (ICRH and ICRL)
- Output compare registers (OCRH and OCRL)

Timer Control Register

The timer control register (TCR) performs the following functions:

- Enables input capture interrupts
- Enables output compare interrupts
- Enables timer overflow interrupts
- Controls the active edge polarity of the TCAP signal
- Controls the active level of the TCMP output

| \$0012 | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|------|------|---|---|---|------|-------|
| Read: | ICIE | OCIE | TOIE | 0 | 0 | 0 | IEDG | OLVL |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | U | 0 |

U = Unaffected

Figure 10. Timer Control Register (TCR)

ICIE — Input Capture Interrupt Enable

This read/write bit enables interrupts caused by an active signal on the PD7/TCAP pin. Reset clears the ICIE bit.

- 1 = Input capture interrupts enabled
- 0 = Input capture interrupts disabled

OCIE — Output Compare Interrupt Enable

This read/write bit enables interrupts caused by an active signal on the TCMP pin. Reset clears the OCIE bit.

- 1 = Output compare interrupts enabled
- 0 = Output compare interrupts disabled

TOIE — Timer Overflow Interrupt Enable

This read/write bit enables interrupts caused by a timer overflow. Reset clears the TOIE bit.

- 1 = Timer overflow interrupts enabled
- 0 = Timer overflow interrupts disabled

Bits 4–2 — Unused

These are read/write bits that always read as logic zeros.

IEDG — Input Edge

The state of this read/write bit determines whether a positive or negative transition on the PD7/TCAP pin triggers a transfer of the contents of the timer registers to the input capture registers. Reset has no effect on the IEDG bit.

- 1 = Positive edge (low-to-high transition) triggers input capture
- 0 = Negative edge (high-to-low transition) triggers input capture

OLVL — Output Level

The state of this read/write bit determines whether a logic one or a logic zero appears on the TCMP pin when a successful output compare occurs. Reset clears the OLVL bit.

- 1 = TCMP goes high on output compare
- 0 = TCMP goes low on output compare

Timer Status Register

The timer status register (TSR) contains flags for the following events:

- An active signal on the PD7/TCAP pin, transferring the contents of the timer registers to the input capture registers
- A match between the 16-bit counter and the output compare registers, transferring the OLVL bit to the TCMP pin
- A timer rollover from \$FFFF to \$0000

| \$0013 | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-----|-----|---|---|---|---|-------|
| Read: | ICF | OCF | TOF | 0 | 0 | 0 | 0 | 0 |
| Write: | | | | | | | | |
| Reset: | U | U | U | 0 | 0 | 0 | 0 | 0 |

= Unimplemented U = Unaffected

Figure 11. Timer Status Register (TSR)

ICF — Input Capture Flag

The ICF bit is automatically set when an edge of the selected polarity occurs on the PD7/TCAP pin. Clear the ICF bit by reading the timer status register with ICF set, and then reading the low byte of the input capture registers. Reset has no effect on ICF.

- 1 = Input capture
- 0 = No input capture

OCF — Output Compare Flag

The OCF bit is automatically set when the value of the timer registers matches the contents of the output compare registers. Clear the OCF bit by reading the timer status register with OCF set, and then reading the low byte of the output compare registers. Reset has no effect on OCF.

- 1 = Output compare
- 0 = No output compare

TOF — Timer Overflow Flag

The TOF bit is automatically set when the 16-bit counter rolls over from \$FFFF to \$0000. Clear the TOF bit by reading the timer status register with TOF set, and then reading the low byte of the timer registers. Reset has no effect on TOF.

- 1 = Timer overflow
- 0 = No timer overflow

Timer Registers

The read-only timer registers (TRH and TRL) contain the current high and low bytes of the 16-bit counter. Reading TRH before reading TRL causes TRL to be latched until TRL is read. Reading TRL after reading the timer status register clears the timer overflow flag (TOF). Writing to the timer registers has no effect.

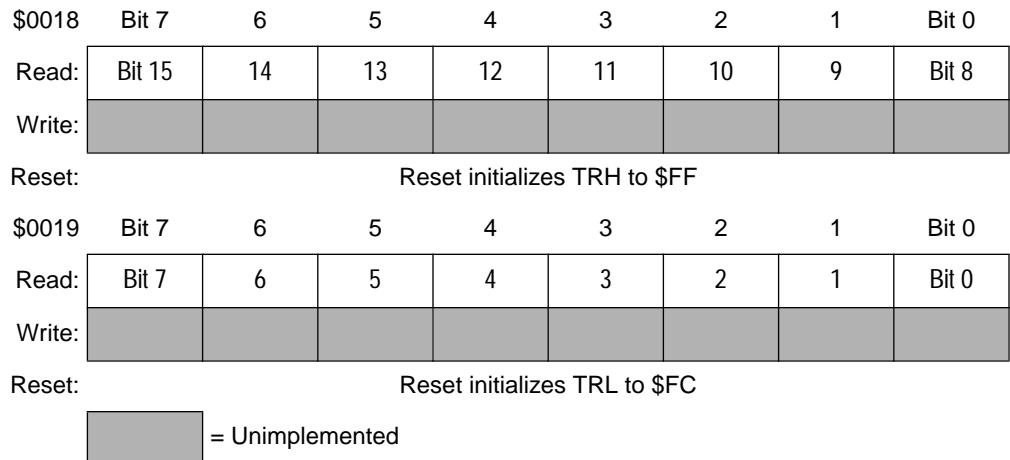


Figure 12. Timer Registers (TRH and TRL)

Reading TRH returns the current value of the high byte of the counter and causes the low byte to be latched into a buffer. The buffer value remains fixed even if the high byte is read more than once. Reading TRL reads the transparent low byte buffer and completes the read sequence of the timer registers.

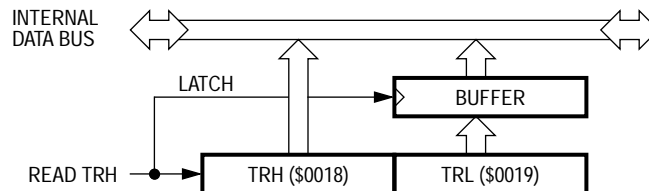


Figure 13. Timer Register Reads

NOTE: To prevent interrupts from occurring between readings of TRH and TRL, set the interrupt mask (I bit) in the condition code register before reading TRH, and clear the mask after reading TRL.

Alternate Timer Registers

The read-only alternate timer registers (ATRH and ATRL) contain the current high and low bytes of the 16-bit counter. Reading ATRH before reading ATRL causes ATRL to be latched until ATRL is read. Reading does not affect the timer overflow flag (TOF). Writing to the alternate timer registers has no effect.

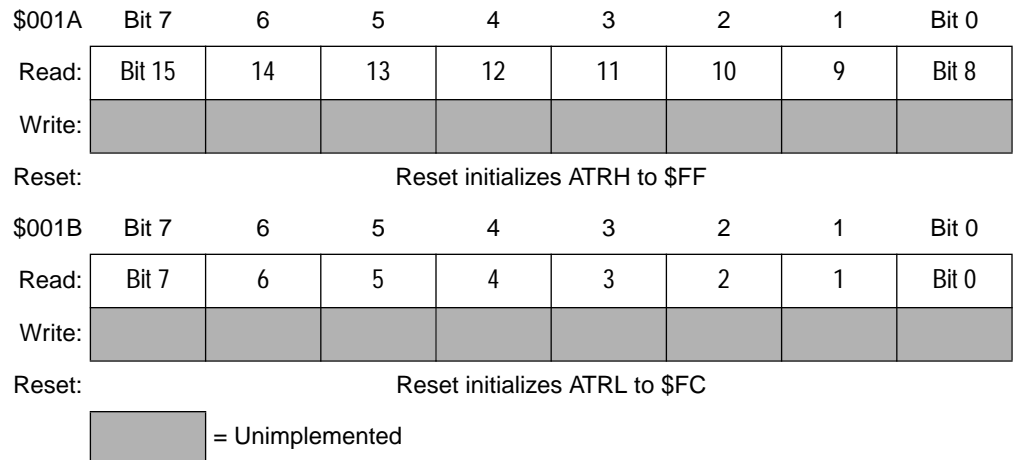


Figure 14. Alternate Timer Registers (ATRH and ATRL)

Reading ATRH returns the current value of the high byte of the counter and causes the low byte to be latched into a buffer.

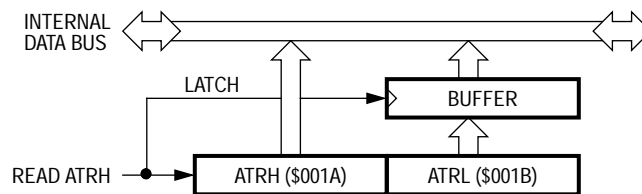


Figure 15. Alternate Timer Register Reads

NOTE: To prevent interrupts between readings of ATRH and ATRL, set the interrupt mask (I bit) in the condition code register before reading ATRH, and clear the mask after reading ATRL.

Input Capture Registers

When a selected edge occurs on the TCAP pin, the current high and low bytes of the 16-bit counter are latched into the read-only input capture registers (ICRH and ICRL). Reading ICRH before reading ICRL inhibits further captures until ICRL is read. Reading ICRL after reading the timer status register clears the input capture flag (ICF). Writing to the input capture registers has no effect.

| | | | | | | | | |
|--------|---------------------|----|----|----|----|----|---|-------|
| \$0014 | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Read: | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit 8 |
| Write: | | | | | | | | |
| Reset: | Unaffected by reset | | | | | | | |
| \$0015 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Read: | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Write: | | | | | | | | |
| Reset: | Unaffected by reset | | | | | | | |

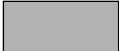
 = Unimplemented

Figure 16. Input Capture Registers (ICRH and ICRL)

NOTE: *To prevent interrupts between readings of ICRH and ICRL, set the interrupt mask (I bit) in the condition code register before reading ICRH, and clear the mask after reading ICRL.*

Output Compare Registers

When the value of the 16-bit counter matches the value in the read/write output compare registers (OCRH and OCRL), the planned TCMP pin action takes place. Writing to OCRH before writing to OCRL inhibits timer compares until OCRL is written. Reading or writing to OCRL after reading the timer status register clears the output compare flag (OCF).

| | | | | | | | | |
|--------|---------------------|----|----|----|----|----|---|-------|
| \$0016 | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Read: | Bit 15 | 14 | 13 | 12 | 11 | 10 | 9 | Bit 8 |
| Write: | | | | | | | | |
| Reset: | Unaffected by reset | | | | | | | |
| \$0017 | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Read: | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Write: | | | | | | | | |
| Reset: | Unaffected by reset | | | | | | | |

Figure 17. Output Compare Registers (OCRH and OCRL)

To prevent OCF from being set between the time it is read and the time the output compare registers are updated, use the following procedure:

1. Disable interrupts by setting the I bit in the condition code register.
2. Write to OCRH. Compares are now inhibited until OCRL is written.
3. Clear bit OCF by reading the timer status register (TSR).
4. Enable the output compare function by writing to OCRL.
5. Enable interrupts by clearing the I bit in the condition code register.

Low-Power Modes

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

Stop Mode

The STOP instruction suspends the timer counter. Upon exit from stop mode by external reset:

- The timer counter resumes counting from \$FFFC.
- An input capture edge during stop mode does not affect the ICF bit or the input capture registers.

Upon exit from stop mode by external interrupt:

- The counter resumes counting from the suspended value.
- An input capture edge during stop mode sets the ICF bit and transfers the suspended timer counter value to the input capture registers.

Wait Mode

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

Serial Input/Output Port SIOP

Contents

| | |
|-----------------------------|-----|
| Features | 108 |
| Introduction | 108 |
| Operation | 110 |
| Pin Functions | 111 |
| PB7/SCK | 111 |
| PB5/SDO | 112 |
| PB6/SDI | 113 |
| Data Movement | 113 |
| Timing | 114 |
| Interrupts | 115 |
| I/O Registers | 116 |
| SIOP Control Register | 116 |
| SIOP Status Register | 118 |
| SIOP Data Register | 119 |
| Low-Power Modes | 120 |
| Stop Mode | 120 |
| Wait Mode | 120 |

Features

- Master or Slave Operation
- Programmable MSB-First or LSB-First Operation
- Interrupt-Driven Operation with Transfer Complete Flag
- Data Collision Flag
- Master Mode Frequency = Bus Frequency \div 4
- Maximum Slave Mode Frequency = Bus Frequency \div 4
- No Minimum Slave Mode Frequency

Introduction

The serial input/output port (SIOP) is a 3-wire master/slave communication port with serial clock, data input, and data output connections. The SIOP enables high-speed synchronous serial data transfer between the MCU and peripheral devices. Shift registers used with the SIOP can increase the number of parallel I/O pins controlled by the MCU. More powerful peripherals such as analog-to-digital converters and real-time clocks are also compatible with the SIOP.

Figure 1 shows the structure of the SIOP module.

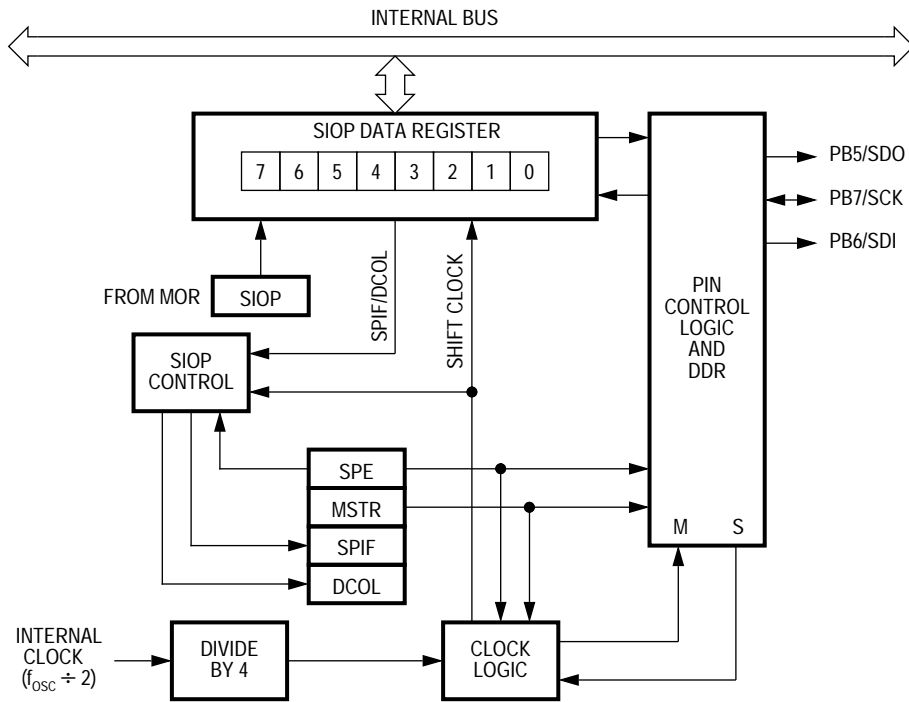


Figure 1. SIOP Block Diagram

| Addr. | Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-----------------------------|--------|---------------------|------|---|------|---|---|---|-------|
| \$000A | SIOP Control Register (SCR) | Read: | 0 | SPE | 0 | MSTR | 0 | 0 | 0 | 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$000B | SIOP Status Register (SSR) | Read: | SPIF | DCOL | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$000C | SIOP Data Register (SDR) | Read: | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |

= Unimplemented

Figure 2. SIOP I/O Register Summary

Operation

The master MCU initiates and controls the transfer of data to and from one or more slave peripheral devices. In master mode, a transmission is initiated by writing to the SIOP data register (SDR). Data written to the SDR is parallel-loaded and shifted out serially to the slave device(s).

Many simple slave devices are designed to only receive data from a master or to only supply data to a master. For example, when a serial-to-parallel shift register is used as an 8-bit port, the master MCU initiates transfers of 8-bit data values to the shift register. Since the serial-to-parallel shift register does not send any data to the master, the MCU ignores whatever it receives as a result of the transmission.

The SIOP is simpler than the serial peripheral interface (SPI) on some other Motorola MCUs in that:

- The polarity of the serial clock is fixed.
- There is no slave select pin.
- The direction of serial data does not automatically switch as on the SPI because the SIOP is not intended for use in multimaster systems. Most applications use one MCU as the master to initiate and control data transfer between one or more slave peripheral devices.

A programmable option allows the SIOP to transfer data MSB first or LSB first.

Pin Functions The SIOPI uses three pins and shares them with port B:

- PB7/SCK
- PB6/SDI
- PB5/SDO

NOTE: *Do not use the PB7/SCK, PB6/SDI, or PB5/SDO pins for general-purpose I/O while the SIOPI is enabled.*

When bit 6 (SPE) of the SIOPI control register (SCR) is set, the SIOPI is enabled and the PB7/SCK, PB5/SDO, and PB6/SDI pins are dedicated to SIOPI functions. Clearing SPE disables the SIOPI and the SIOPI pins become standard I/O port pins.

NOTE: *Enabling and then disabling the SIOPI configures the data direction register bits associated with the SIOPI pins for SIOPI operation and can also change the associated port data register. After disabling the SIOPI, initialize the data direction register and the port data register as the application requires.*

PB7/SCK

The PB7/SCK pin synchronizes the movement of data into and out of the MCU through the PB6/SDI and PB5/SDO pins. In master mode, the PB7/SCK pin is an output. The serial clock frequency in master mode is one-fourth the internal clock frequency.

In slave mode, the PB7/SCK pin is an input. The maximum serial clock frequency in slave mode is one-fourth the internal clock rate. Slave mode has no minimum serial clock frequency.

Figure 3 shows the timing relationships among the serial clock, data input, and data output. The state of the serial clock between transmissions is a logic one. The first falling edge on the PB7/SCK pin signals the beginning of a transmission, and data appears at the PB5/SDO pin. Data is captured at the PB6/SDI pin on the rising edge of the serial clock, and the transmission ends on the eighth rising edge of the serial clock.

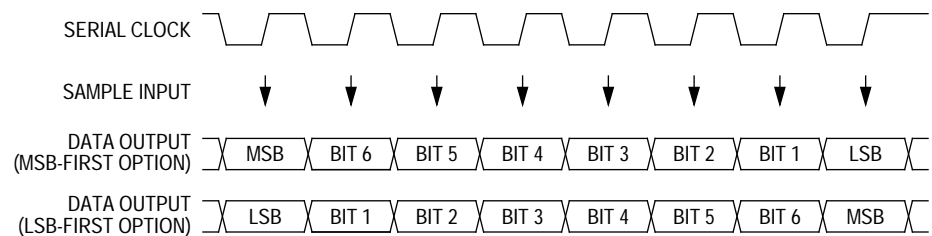


Figure 3. SIOP Data/Clock Timing

The first falling edge on PB7/SCK begins a transmission. At this time the first bit of received data is accepted at the PB6/SDI pin and the first bit of transmitted data is presented at the PB5/SDO pin.

PB5/SDO

The PB5/SDO pin is the SIOP data output. Between transfers, the state of the PB5/SDO pin reflects the value of the last bit shifted out on the previous transmission, if there was one. To preset the beginning state, write to the corresponding port data bit before enabling the SIOP. On the first falling edge on the PB7/SCK pin, the first data bit to be shifted out appears at the PB5/SDO pin.

After SPE is set, the PB5/SDO output driver can be disabled by writing a zero to the corresponding data direction register bit of the port, thereby configuring PB5/SDO as a high-impedance input.

PB6/SDI

The PB6/SDI pin is the SIOPI data input. Valid SDI data must be present for an SDI setup time, t_{s} , before the rising edge of the serial clock and must remain valid for an SDI hold time, t_{H} , after the rising edge of the serial clock. (See [Table 1](#) and [Table 2](#).)

Data Movement

Connecting the SIOPI data register of a master MCU with the SIOPI of a slave MCU forms a 16-bit circular shift register. During an SIOPI transfer, the master shifts out the contents of its SIOPI data register on its PB5/SDO pin. At the same time, the slave MCU shifts out the contents of its SIOPI data register on its PB5/SDO pin. [Figure 4](#) shows how the master and slave exchange the contents of their data registers.

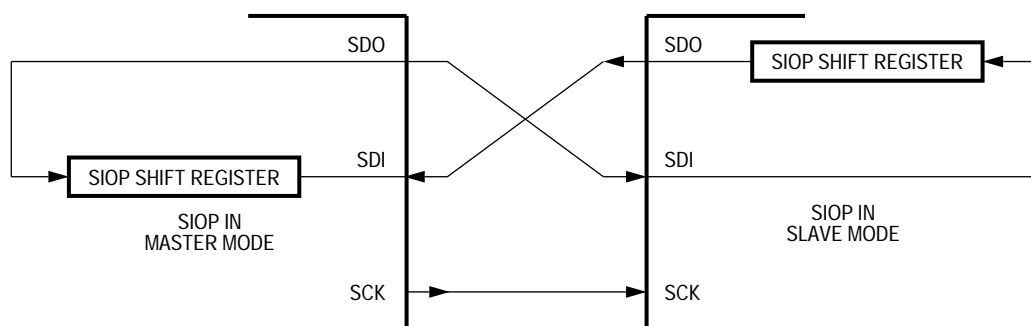


Figure 4. Master/Slave SIOPI Shift Register Operation

Timing

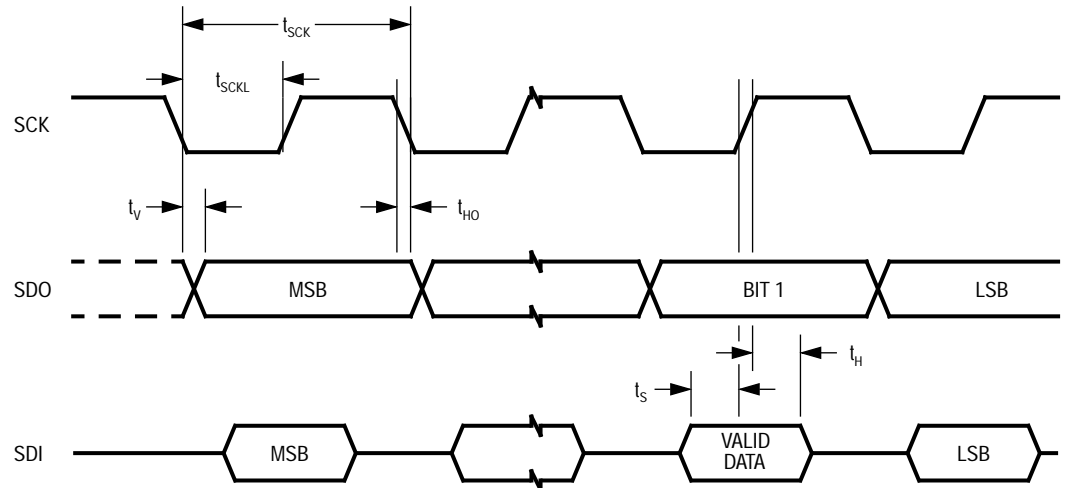


Figure 5. SIOP Timing

Table 1. SIOP Timing ($V_{DD} = 5.0 \text{ Vdc}$)⁽¹⁾

| Characteristic | Symbol | Min | Max | Unit |
|---|--------------------------------|--------------------|--------------------|-----------------------|
| Frequency of Operation Master Slave | $f_{SIOP(M)}$ $f_{SIOP(S)}$ | $f_{OSC}/64$ dc | $f_{OSC}/8$ 525 | MHz kHz |
| Cycle Time Master Slave | $t_{SCK(M)}$ $t_{SCK(S)}$ | 4.0 — | 4.0 1920 | $t_{CYC}^{(2)}$ ns |
| Clock (SCK) Low Time ($f_{OP} = 2.1 \text{ MHz}$) ⁽³⁾⁽⁴⁾ | t_{SCKL} | 932 | — | ns |
| SDO Data Valid Time | t_v | — | 200 | ns |
| SDO Hold Time | t_{HO} | 0 | — | ns |
| SDI Setup Time | t_s | 100 | — | ns |
| SDI Hold Time | t_H | 100 | — | ns |

1. $V_{DD} = 5.0 \text{ Vdc} \pm 10\%$, $V_{SS} = 0 \text{ Vdc}$, $T_A = T_L$ to T_H unless otherwise noted.
2. $t_{CYC} = 1 \div f_{OP}$
3. f_{OSC} = crystal frequency; $f_{OP} = f_{OSC} \div 2 = 2.1 \text{ MHz}$ maximum
4. In master mode, the frequency of SCK is $f_{OP} \div 4$.

Table 2. SIOPI Timing ($V_{DD} = 3.3 \text{ Vdc}$)⁽¹⁾

| Characteristic | Symbol | Min | Max | Unit |
|---|--|---------------------------|---------------------------|---------------------------------|
| Frequency of Operation Master Slave | $f_{\text{SIOPI(M)}}$ $f_{\text{SIOPI(S)}}$ | $f_{\text{OSC}}/64$ dc | $f_{\text{OSC}}/8$ 250 | MHz kHz |
| Cycle Time Master Slave | $t_{\text{SCK(M)}}$ $t_{\text{SCK(S)}}$ | 4.0 — | 4.0 4000 | t_{CYC} ⁽²⁾ |
| Clock (SCK) Low Time ($f_{\text{OP}} = 1.0 \text{ MHz}$) ^{(3) (4)} | t_{SCKL} | 1980 | — | ns |
| SDO Data Valid Time | t_{V} | — | 400 | ns |
| SDO Hold Time | t_{HO} | 0 | — | ns |
| SDI Setup Time | t_{S} | 200 | — | ns |
| SDI Hold Time | t_{H} | 200 | — | ns |

1. $V_{DD} = 3.3 \text{ Vdc} \pm 10\%$, $V_{SS} = 0 \text{ Vdc}$, $T_A = T_L$ to T_H unless otherwise noted

2. $t_{\text{CYC}} = 1 \div f_{\text{OP}}$

3. f_{OSC} = crystal frequency; $f_{\text{OP}} = f_{\text{OSC}} \div 2 = 1.0 \text{ MHz}$ maximum

4. In master mode, the frequency of SCK is $f_{\text{OP}} \div 4$.

Interrupts

The SIOPI does not generate interrupt requests.

I/O Registers

The following registers control and monitor SIOP operation:

- SIOP control register (SCR)
- SIOP status register (SSR)
- SIOP data register (SDR)

SIOP Control Register

The read/write SIOP control register (SCR) contains two bits. One bit enables the SIOP, and the other configures the SIOP for master mode or for slave mode.

| \$000A | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-----|---|------|---|---|---|-------|
| Read: | 0 | SPE | 0 | MSTR | 0 | 0 | 0 | 0 |
| Write: | 0 | SPE | 0 | MSTR | 0 | 0 | 0 | 0 |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 6. SIOP Control Register (SCR)

SPE — SIOP Enable

This read/write bit enables the SIOP. Setting SPE initializes the data direction register as follows:

- The PB6/SDI pin is an input.
- The PB5/SDO pin is an output.
- The PB7/SCK pin is an input in slave mode and an output in master mode.

Clearing SPE disables the SIOP and returns the port to its normal I/O functions. The data direction register and the port data register remain in their SIOP-initialized state.

NOTE: *After clearing SPE, be sure to initialize the port for its intended I/O use.*

Clearing SPE during a transmission aborts the transmission, resets the bit counter, and returns the port to its normal I/O function. Reset clears SPE.

- 1 = SIOP enabled
- 0 = SIOP disabled

MSTR — Master Mode Select

This read/write bit configures the SIOP for master mode. Setting MSTR initializes the PB7/SCK pin as the serial clock output. Clearing MSTR initializes the PB7/SCK pin as the serial clock input. MSTR can be set at any time regardless of the state of SPE. Reset clears MSTR.

- 1 = Master mode selected
- 0 = Slave mode selected

SIOP Status Register

The read-only SIOP status register (SSR) contains two bits. One bit indicates that a SIOP transfer is complete, and the other indicates that an invalid access of the SIOP data register occurred while a transfer was in progress.

| \$000B | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|------|---|---|---|---|---|-------|
| Read: | SPIF | DCOL | 0 | 0 | 0 | 0 | 0 | 0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 7. SIOP Status Register (SSR)

SPIF — Serial Peripheral Interface Flag

This clearable, read-only bit is set automatically on the eighth rising edge on the PB7/SCK pin and indicates that a data transmission took place. SPIF does not inhibit further transmissions. Clear SPIF by reading the SIOP status register while SPIF is set and then reading or writing the SIOP data register. Reset clears SPIF.

- 1 = Transmission complete
- 0 = Transmission not complete

DCOL — Data Collision Flag

This clearable, read-only bit is automatically set if the SIOP data register is accessed while a data transfer is in progress. Reading or writing the SIOP data register while a transmission is in progress causes invalid data to be transmitted or read. Clear DCOL by reading the SIOP status register with SPIF set and then accessing the SIOP data register. Because the clearing sequence accesses the SIOP data register, the sequence has to be completed before another transmission starts or DCOL is set again.

To clear DCOL when SPIF is not set, turn off the SIOP by writing a zero to SPE and then turn it back on by writing a one to SPE. Reset clears DCOL.

- 1 = Invalid access of SDR
- 0 = Valid access of SDR

SIOP Data Register

The SIOP data register (SDR) is both the transmit data register and the receive data register. To read or write the SIOP data register, the SPE bit in the SIOP control register must be set.



Figure 8. SIOP Data Register (SDR)

With the SIOP configured for master mode, writing to the SIOP data register initiates a serial transfer. This register is not buffered. Writing to the SIOP data register overwrites the previous contents. Reading or writing to the SIOP data register while a transmission is in progress can cause invalid data to be transmitted or received.

Low-Power Modes

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

Stop Mode

The STOP instruction suspends the clock to the SIOPI. When the MCU exits stop mode, processing resumes after the internal oscillator stabilization delay of 4064 oscillator cycles.

A STOP instruction in a master SIOPI does not suspend the clock to slave SIOPIs.

Wait Mode

The WAIT instruction suspends the clock to the SIOPI. When the MCU exits wait mode, processing resumes immediately.

A WAIT instruction in a master SIOPI does not suspend the clock to slave SIOPIs.

Analog-to-Digital Converter

ADC

Contents

| | |
|---|-----|
| Features | 121 |
| Introduction | 122 |
| Operation | 123 |
| Pin Functions | 123 |
| PC7/V _{RH} | 123 |
| PC6/AN0–PC3/AN3 | 124 |
| Interrupts | 124 |
| Timing and Electrical Characteristics | 125 |
| I/O Registers | 126 |
| ADC Status and Control Register | 126 |
| ADC Data Register | 129 |
| Low-Power Modes | 130 |
| Stop Mode | 130 |
| Wait Mode | 130 |

Features

- 8-Bit Conversions with ± 1.5 -LSB Precision
- Four External and Three Internal Analog Input Channels
- Wait Mode Operation

Introduction

The ADC consists of a single successive-approximation A/D converter, an input multiplexer to select one of four external or two internal channels, and control circuitry. **Figure 1** shows the structure of the ADC module.

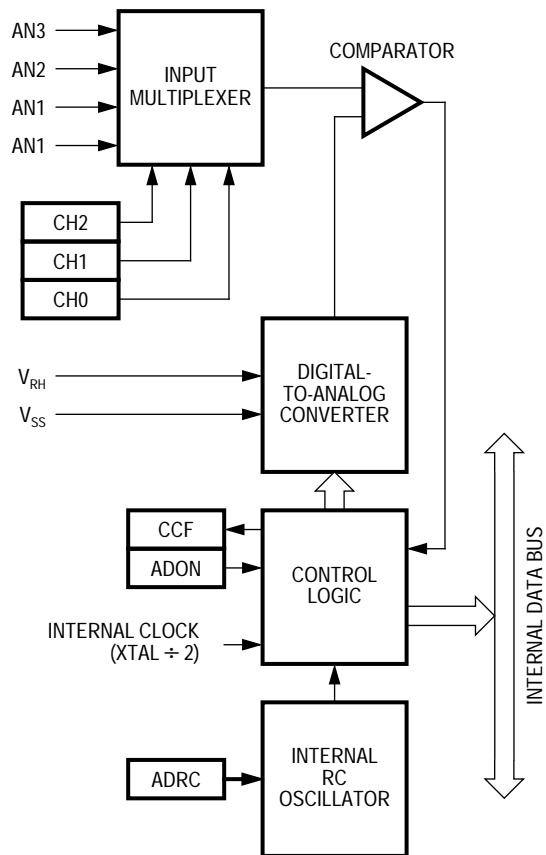



Figure 1. ADC Block Diagram

Table 1. ADC I/O Register Summary

| Addr. | Name | R/W | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------------------------------------|--------|---------------------|------|------|---|---|-----|-----|-------|
| \$001D | ADC Data Register (ADDR) | Read: | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| \$001E | ADC Status/Control Register (ADSCR) | Read: | CCF | ADRC | ADON | 0 | 0 | CH2 | CH1 | CH0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

 = Unimplemented

Operation

The A/D conversion process is ratiometric, using two reference voltages, V_{RH} and V_{SS} . Conversion accuracy is guaranteed only if V_{RH} is equal to V_{DD} .

Pin Functions

The ADC uses five pins and shares them with port C:

- $PC7/V_{RH}$
- $PC6/AN0$, $PC5/AN1$, $PC4/AN2$, and $PC3/AN3$

$PC7/V_{RH}$

The voltage reference high pin ($PC7/V_{RH}$) supplies the high reference voltage for the ratiometric conversion process. For ratiometric conversion, the supply voltage of the analog source should be the same as V_{RH} and be referenced to V_{SS} .

PC6/AN0–
PC3/AN3

The multiplexer can select one of four external analog input channels (AN0, AN1, AN2, or AN3) for sampling. The conversion takes 32 cycles. The first 12 cycles sample the voltage on the selected input pin by charging an internal capacitor. In the last 20 cycles, a comparator successively compares the output of an internal D/A converter to the sampled analog input. Control logic changes the D/A converter input one bit at a time, starting with the MSB, until the D/A converter output matches the sampled analog input. The conversion is monotonic and has no missing codes. At the end of the conversion, the conversion complete flag (CCF) becomes set, and the CPU takes two cycles to move the result to the ADC data register.

NOTE: *To prevent excess power dissipation, do not simultaneously use an I/O port pin as a digital input and an analog input.*

While the ADC is on, the selected analog input reads as logic zero. The port C pins that are not selected read normally.

An analog input voltage equal to V_{RH} converts to digital \$FF; an input voltage greater than V_{RH} converts to \$FF with no overflow. An analog input voltage equal to V_{SS} converts to digital \$00. For ratiometric conversion, the source of each analog input should use V_{RH} as the supply voltage and be referenced to V_{SS} .

The clock frequency must be equal to or greater than 1 MHz. If the internal clock frequency is less than 1 MHz, the internal RC oscillator (nominally 1.5 MHz) must be used for the ADC conversion clock. Make this selection by setting the ADRC bit to logic one in the ADC status and control register.

Interrupts

The ADC cannot generate interrupt requests.

Timing and Electrical Characteristics

Table 2. ADC Characteristics ($V_{DD} = 5.0 \text{ Vdc}$)⁽¹⁾

| Characteristic | Min | Max | Unit |
|--|-------------------------------|-----------|-------------------------|
| Resolution | 8 | 8 | Bit |
| Absolute Accuracy ($4.0 > V_{RH} > V_{DD}$) ⁽²⁾ | — | ± 1.5 | LSB |
| Conversion Range ($PC7/V_{RH}$) | V_{SS} | V_{DD} | V |
| Conversion Time (Includes Sampling Time) | | | |
| External Clock | 32 | 32 | t_{AD} ⁽³⁾ |
| Internal RC Oscillator (ADRC = 1) | 32 | 32 | μs |
| Monotonicity | Inherent (Within Total Error) | | |
| Zero Input Reading ($V_{IN} = 0 \text{ V}$) | 00 | 01 | Hex |
| Full-Scale Reading ($V_{IN} = V_{RH}$) | FF | FF | Hex |
| Sample Acquisition Time ⁽⁴⁾ | | | |
| External Clock | 12 | 12 | t_{AD} ⁽⁵⁾ |
| Internal RC Oscillator (ADRC = 1) | — | 12 | μs |
| Input Capacitance PC6/AN0, PC5/AN1, PC4/AN2, PC3/AN3 | — | 12 | pF |
| Analog Input Voltage | V_{SS} | V_{RH} | V |
| Input Leakage ⁽⁶⁾ | | | |
| PC6/AN0, PC5/AN1, PC4/AN2, PC3/AN3 | — | ± 1 | μA |
| PC7/ V_{RH} | — | ± 1 | μA |
| ADC On Current Stabilization Time | — | 100 | μs |

- $V_{DD} = 5.0 \text{ Vdc} \pm 10\%$, $V_{SS} = 0 \text{ Vdc}$
- ADC accuracy may decrease proportionately as V_{RH} is reduced below 4.0 V.
- t_{AD} = cycle time of the A/D converter
- Source impedances more than 10 k Ω adversely affect internal RC charging time during input sampling.
- $t_{AD} = t_{CYC} (1 \div f_{OP})$ if MCU clock is clock source
- External system error caused by input leakage approximately equals R source times input current.

I/O Registers

The following registers control and monitor operation of the ADC:

- ADC status and control register (ADSCR)
- ADC data register (ADDR)

ADC Status and Control Register

The ADC status and control register (ADSCR) contains a conversion complete flag and four writable control bits. Writing to ADSCR clears the conversion complete flag and starts a new conversion sequence.

| \$001E | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|------|------|---|---|-----|-----|-------|
| Read: | CCF | ADRC | ADON | 0 | 0 | CH2 | CH1 | CH0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |


 = Unimplemented

Figure 2. ADC Status and Control Register (ADSCR)

CCF — Conversion Complete Flag

This read-only bit is automatically set when an analog-to-digital conversion is complete, and a new result can be read from the ADC data register. Clear the CCF bit by writing to the ADC status and control register or by reading the ADC data register. Resets clear the CCF bit.

- 1 = Conversion complete
- 0 = Conversion not complete

ADRC — ADC RC (Oscillator)

This read/write bit turns on the internal RC oscillator to drive the ADC. If the internal clock frequency (f_{OP}) is less than 1 MHz, ADRC must be set. When the RC oscillator is turned on, it requires a time, t_{ADRC} , to stabilize, and results can be inaccurate during this time. Resets clear the ADRC bit.

1 = Internal RC oscillator drives ADC

0 = Internal clock drives ADC

When the internal RC oscillator is being used as the ADC clock, two limitations apply:

- Because of the frequency tolerance of the RC oscillator and its asynchronism with the internal clock, the conversion complete flag must be used to determine when a conversion sequence is complete.
- The conversion process runs at the nominal 1.5-MHz rate, but the conversion results must be transferred to the ADC data register synchronously with the internal clock; therefore, the conversion process is limited to a maximum of one channel every internal clock cycle.

ADON — ADC On

This read/write bit turns on the ADC. When the ADC is on, it requires a time, t_{ADON} , for the current sources to stabilize. During this time, results can be inaccurate. Resets clear the ADON bit.

1 = ADC turned on

0 = ADC turned off

Bits 4–2 — Not used

Bits 4–2 always read as logic zeros.

CH[2:0] — Channel Select Bits

These read/write bits select one of eight ADC input channels as shown in [Table 3](#). Channels 0–3 are the input pins, PC3/AN3, PC4/AN2, PC5/AN1, and PC6/AN0. Channels 4–6 can be used for reference measurements. Channel 7 is reserved for factory testing.

Table 3. ADC Input Channel Selection

| CH[2:1:0] | Channel | Signal |
|-----------|---------|-------------------------|
| 000 | 0 | AN0 |
| 001 | 1 | AN1 |
| 010 | 2 | AN2 |
| 011 | 3 | AN3 |
| 100 | 4 | V_{RH} |
| 101 | 5 | $(V_{RH} + V_{SS}) / 2$ |
| 110 | 6 | V_{SS} |
| 111 | 7 | Reserved |

To prevent excess power dissipation, do not use an ADC pin as an analog input and a digital input at the same time.

Using one of the port pins as the ADC input does not affect the ability to use the remaining port pins as digital inputs.

Reading a port pin that is selected as an analog input returns a logic zero.

ADC Data Register The ADC data register (ADDR) is a read-only register that contains the result of the most recent analog-to-digital conversion.

| | | | | | | | | |
|--------|---------------------|---|---|---|---|---|---|-------|
| \$001D | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Read: | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Write: | | | | | | | | |
| Reset: | Unaffected by reset | | | | | | | |

Figure 3. ADC Data Register (ADDR)

Low-Power Modes

- Stop Mode** The STOP instruction turns off the ADC and aborts any current and pending conversions.
- Wait Mode** The ADC continues to operate normally after the WAIT instruction. To reduce power consumption in wait mode:
- If the ADC is not being used, clear both the ADON and ADRC bits before entering wait mode.
 - If the ADC is being used and the internal clock rate is above 1 MHz, clear the ADRC bit before entering wait mode.

Contents

| | |
|---|-----|
| Maximum Ratings | 132 |
| Operating Temperature Range | 133 |
| Thermal Characteristics | 133 |
| Power Considerations | 134 |
| 5.0 V DC Electrical Characteristics | 135 |
| 3.3 V DC Electrical Characteristics | 136 |
| Driver Characteristics | 137 |
| Typical Supply Current vs. Internal Clock Frequency | 138 |
| Maximum Supply Current vs. Internal Clock Frequency | 139 |
| 5.0 V Control Timing | 140 |
| 3.3 V Control Timing | 141 |
| Test Load | 142 |
| Mechanical Specifications | 142 |
| 28-Pin PDIP — Case #710 | 143 |
| 28-Pin Cerdip — Case #733 | 143 |
| 28-Pin SOIC — Case #751F | 144 |

Maximum Ratings

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

The MCU contains circuitry to protect the inputs against damage from high static voltages; however, do not apply voltages higher than those shown in [Table 1](#). Keep V_{IN} and V_{OUT} within the range $V_{SS} \leq (V_{IN} \text{ or } V_{OUT}) \leq V_{DD}$. Connect unused inputs to the appropriate voltage level, either V_{SS} or V_{DD} .

Table 1. Maximum Ratings

| Rating | Symbol | Value | Unit |
|---|-----------|----------------------------------|------|
| Supply Voltage | V_{DD} | -0.3 to +7.0 | V |
| Current Drain per Pin (Excluding V_{DD} and V_{SS}) | I | 25 | mA |
| Input Voltage | V_{IN} | $V_{SS} - 0.3$ to $V_{DD} + 0.3$ | V |
| EPROM Programming Voltage | V_{PP} | 16.75 | V |
| Storage Temperature Range | T_{STG} | -65 to +150 | °C |

NOTE: *This device is not guaranteed to operate properly at the maximum ratings. Refer to [5.0 V DC Electrical Characteristics](#) on page 135 and [3.3 V DC Electrical Characteristics](#) on page 136 for guaranteed operating conditions.*

Operating Temperature Range

Table 2. Operating Temperature Range

| Package Type | Symbol | Value | Unit |
|--|--------|---|------|
| MC68HC705P9P ⁽¹⁾ , DW ⁽²⁾ , S ⁽³⁾ (Standard) MC68HC705P9C ⁽⁴⁾ P, CDW, CS (Extended) MC68HC705P9V ⁽⁵⁾ P, VDW, VS (Automotive) MC68HC705P9M ⁽⁶⁾ P, MDW, MS (Automotive) | T_A | T_L to T_H 0 to 70 -40 to +85 -40 to +105 -40 to +125 | °C |

1. P = Plastic dual in-line package (PDIP)
2. DW = Small outline integrated circuit (SOIC)
3. S = Ceramic dual in-line package (Cerdip)
4. C = Extended temperature range (-40 to +85 °C)
5. V = Automotive temperature range (-40 to +105 °C)
6. M = Automotive temperature range (-40 to +125 °C)

Thermal Characteristics

Table 3. Thermal Characteristics

| Characteristic | Symbol | Value | Unit |
|---|---------------|----------------|------|
| Thermal Resistance Plastic Dual In-Line Package (PDIP) Small Outline Integrated Circuit (SOIC) Ceramic Dual In-Line Package (Cerdip) | θ_{JA} | 60 60 60 | °C/W |

Power Considerations

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

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

where:

T_A = ambient temperature in °C

θ_{JA} = package thermal resistance, junction to ambient in °C/W

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

$P_{INT} = I_{CC} \times V_{CC}$ = chip internal power dissipation

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

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

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

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

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

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

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

5.0 V DC Electrical Characteristics

Table 4. DC Electrical Characteristics ($V_{DD} = 5.0 \text{ Vdc}$)⁽¹⁾

| Characteristic | Symbol | Min | Typ ⁽²⁾ | Max | Unit |
|--|-----------------------|---------------------------------|---------------------------------------|--------------------------------------|---|
| Output Voltage $I_{LOAD} = 10.0 \mu\text{A}$ $I_{LOAD} = -10.0 \mu\text{A}$ | V_{OL} V_{OH} | — $V_{DD} - 0.1$ | — — | 0.1 — | V |
| Output High Voltage ($I_{LOAD} = -0.8 \text{ mA}$) PA7–PA0, PB7/SCK–PB5/SDO, PC7/ V_{RH} –PC0, PD5, TCMP | V_{OH} | $V_{DD} - 0.8$ | — | — | V |
| Output Low Voltage ($I_{LOAD} = 1.6 \text{ mA}$) PA7–PA0, PB7/SCK–PB5/SDO, PC7/ V_{RH} –PC0, PD5, TCMP | V_{OL} | — | — | 0.4 | V |
| Input High Voltage PA7–PA0, PB7/SCK–PB5/SDO, PC7/ V_{RH} –PC0, PD5, PD7/TCAP, $\overline{\text{IRQ}}/V_{PP}$, $\overline{\text{RESET}}$, OSC1 | V_{IH} | $0.7 \times V_{DD}$ | — | V_{DD} | V |
| Input Low Voltage PA7–PA0, PB7/SCK–PB5/SDO, PC7/ V_{RH} –PC0, PD5, PD7/TCAP, $\overline{\text{IRQ}}/V_{PP}$, $\overline{\text{RESET}}$, OSC1 | V_{IL} | V_{SS} | — | $0.2 \times V_{DD}$ | V |
| Supply Current ^{(3) (4) (5) (6)} Run Mode Wait Mode (ADC On) Wait Mode (ADC Off) Stop Mode 25 °C 0 to 70 °C (Standard) –40 to 125 °C | I_{DD} | — — — — — — — | 4.7 2.1 1.3 2 — — — | 6.5 2.9 1.9 30 50 100 | mA mA mA μA μA μA |
| I/O Ports Hi-Z Leakage Current PA7–PA0, PB7/SCK–PB5/SDO, PC7/ V_{RH} –PC0, PD5 | I_{IL} | — | — | ± 10 | μA |
| ADC Ports Hi-Z Leakage Current | I_{OZ} | — | — | ± 1 | μA |
| Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}/V_{PP}$, OSC1, PD7/TCAP | I_{IN} | — | — | ± 1 | μA |
| Capacitance Ports (As Inputs or Outputs) $\overline{\text{RESET}}$, $\overline{\text{IRQ}}/V_{PP}$ | C_{OUT} C_{IN} | — — | — — | 12 8 | pF |
| Programming Voltage | V_{PP} | 16.25 | 16.5 | 16.75 | V |
| Programming Current | I_{PP} | — | 5 | 10 | mA |
| Programming Time per Byte | t_{EPGM} | 4 | — | — | ms |

- $V_{DD} = 5.0 \text{ Vdc} \pm 10\%$, $V_{SS} = 0 \text{ Vdc}$, $T_A = T_L$ to T_H unless otherwise noted
- Typical values at midpoint of voltage range, 25 °C only
- Run mode and wait mode I_{DD} measured using external square wave clock source ($f_{OSC} = 4.2 \text{ MHz}$); all inputs 0.2 V from rail; no dc loads; less than 50 pF on all outputs; $C_L = 20 \text{ pF}$ on OSC2
- Wait mode and stop mode I_{DD} measured with all ports configured as inputs; $V_{IL} = 0.2 \text{ V}$; $V_{IH} = V_{DD} - 0.2 \text{ V}$
- Stop mode I_{DD} measured with $OSC1 = V_{SS}$
- Wait mode I_{DD} affected linearly by OSC2 capacitance

3.3 V DC Electrical Characteristics

Table 5. DC Electrical Characteristics ($V_{DD} = 3.3 \text{ Vdc}$)⁽¹⁾

| Characteristic | Symbol | Min | Typ ⁽²⁾ | Max | Unit |
|---|----------------------|---------------------|--------------------|---------------------|---------------|
| Output Voltage ($I_{LOAD} \leq 10.0 \mu\text{A}$) | V_{OL} V_{OH} | — $V_{DD} - 0.1$ | — — | 0.1 — | V |
| Output High Voltage ($I_{LOAD} = -0.2 \text{ mA}$) PA7-PA0, PB7/SCK-PB5/SDO, PC7/ V_{RH} -PC0, PD5, TCMP | V_{OH} | $V_{DD} - 0.3$ | — | — | V |
| Output Low Voltage ($I_{LOAD} = 0.4 \text{ mA}$) PA7-PA0, PB7/SCK-PB5/SDO, PC7/ V_{RH} -PC0, PD5, TCMP | V_{OL} | — | — | 0.3 | V |
| Input High Voltage PA7-PA0, PB7/SCK-PB5/SDO, PC7/ V_{RH} -PC0, PD5, PD7/TCAP, $\overline{\text{IRQ}}/V_{PP}$, $\overline{\text{RESET}}$, OSC1 | V_{IH} | $0.7 \times V_{DD}$ | — | V_{DD} | V |
| Input Low Voltage PA7-PA0, PB7/SCK-PB5/SDO, PC7/ V_{RH} -PC0, PD5, PD7/TCAP, $\overline{\text{IRQ}}/V_{PP}$, $\overline{\text{RESET}}$, OSC1 | V_{IL} | V_{SS} | — | $0.2 \times V_{DD}$ | V |
| Data-Retention Mode Supply Voltage | V_{RM} | 2.0 | — | — | V |
| Supply Current ^{(3) (4) (5) (6)} | | | | | |
| Run Mode | | — | 1.6 | 2.3 | mA |
| Wait Mode (ADC On) | | — | 0.9 | 1.3 | mA |
| Wait Mode (ADC Off) | | — | 0.4 | 0.6 | mA |
| Stop Mode | I_{DD} | | | | |
| 25 °C | | — | 1.0 | 20 | μA |
| 0 to 70 °C (Standard) | | — | — | 40 | μA |
| -40 to 125 °C | | — | — | 50 | μA |
| I/O Ports Hi-Z Leakage Current PA7-PA0, PB7/SCK-PB5/SDO, PC7/ V_{RH} -PC0, PD5 | I_{IL} | — | — | ± 10 | μA |
| Input Current $\overline{\text{RESET}}$, $\overline{\text{IRQ}}/V_{PP}$, OSC1, PD7/TCAP | I_{IN} | — | — | ± 1 | μA |
| Capacitance | | | | | |
| Ports (As Inputs or Outputs) | C_{OUT} | — | — | 12 | pF |
| $\overline{\text{RESET}}$, $\overline{\text{IRQ}}/V_{PP}$ | C_{IN} | — | — | 8 | pF |

1. $V_{DD} = 3.3 \text{ Vdc} \pm 10\%$, $V_{SS} = 0 \text{ Vdc}$, $T_A = T_L$ to T_H unless otherwise noted

2. Typical values at midpoint of voltage range, 25 °C only

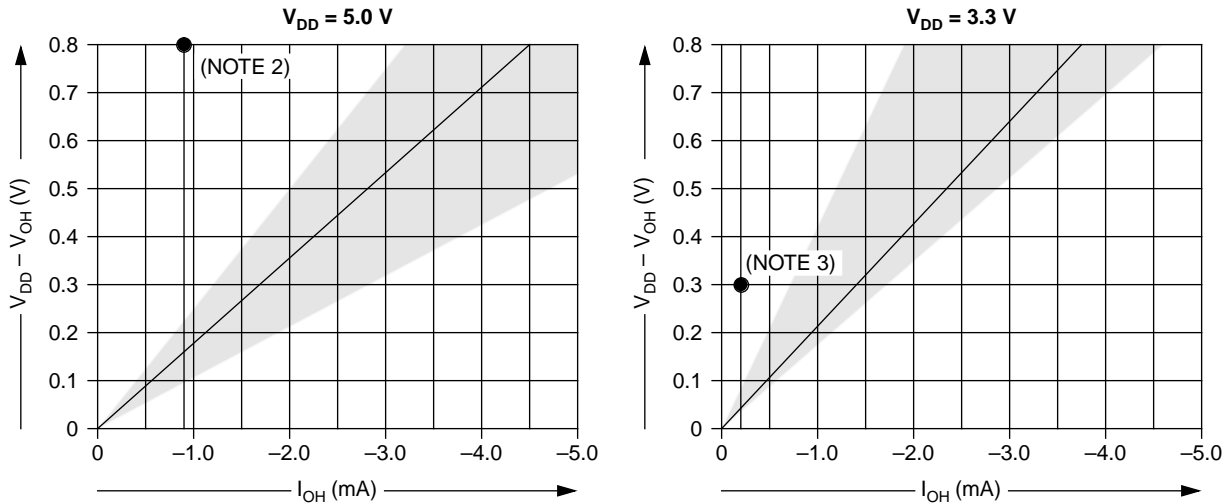
3. Run mode and wait mode I_{DD} measured using external square wave clock source ($f_{OSC} = 2.1 \text{ MHz}$); all inputs 0.2 V from rail; no dc loads; less than 50 pF on all outputs; $C_L = 20 \text{ pF}$ on OSC2

4. Wait mode and stop mode I_{DD} measured with all ports configured as inputs; $V_{IL} = 0.2 \text{ V}$; $V_{IH} = V_{DD} - 0.2 \text{ V}$

5. Stop mode I_{DD} measured with $\text{OSC1} = V_{SS}$

6. Wait mode I_{DD} affected linearly by OSC2 capacitance

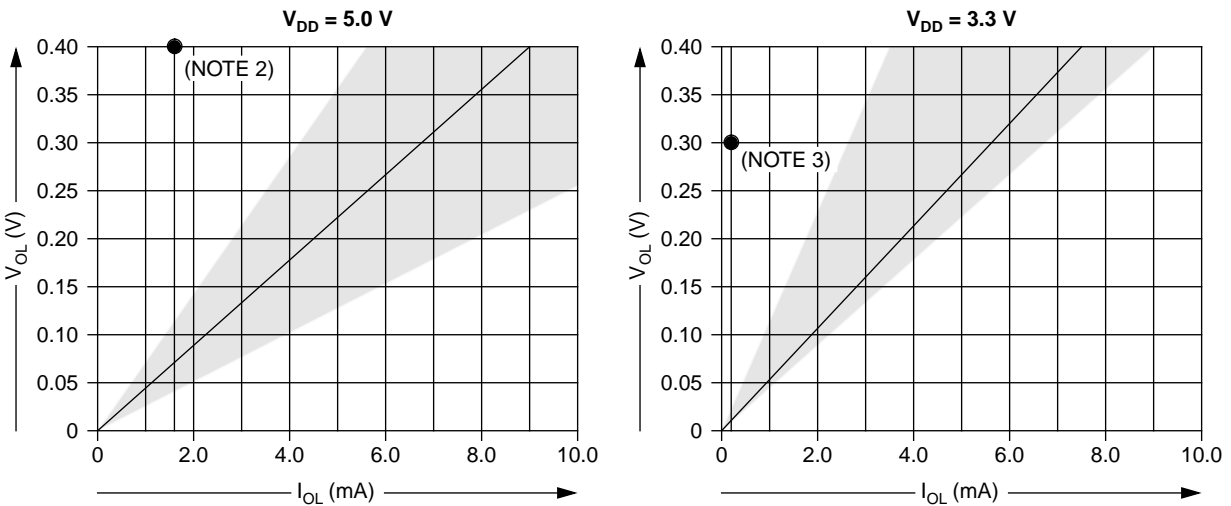
Driver Characteristics



NOTES:

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

Figure 1. Typical High-Side Driver Characteristics



NOTES:

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

Figure 2. Typical Low-Side Driver Characteristics

Typical Supply Current vs. Internal Clock Frequency

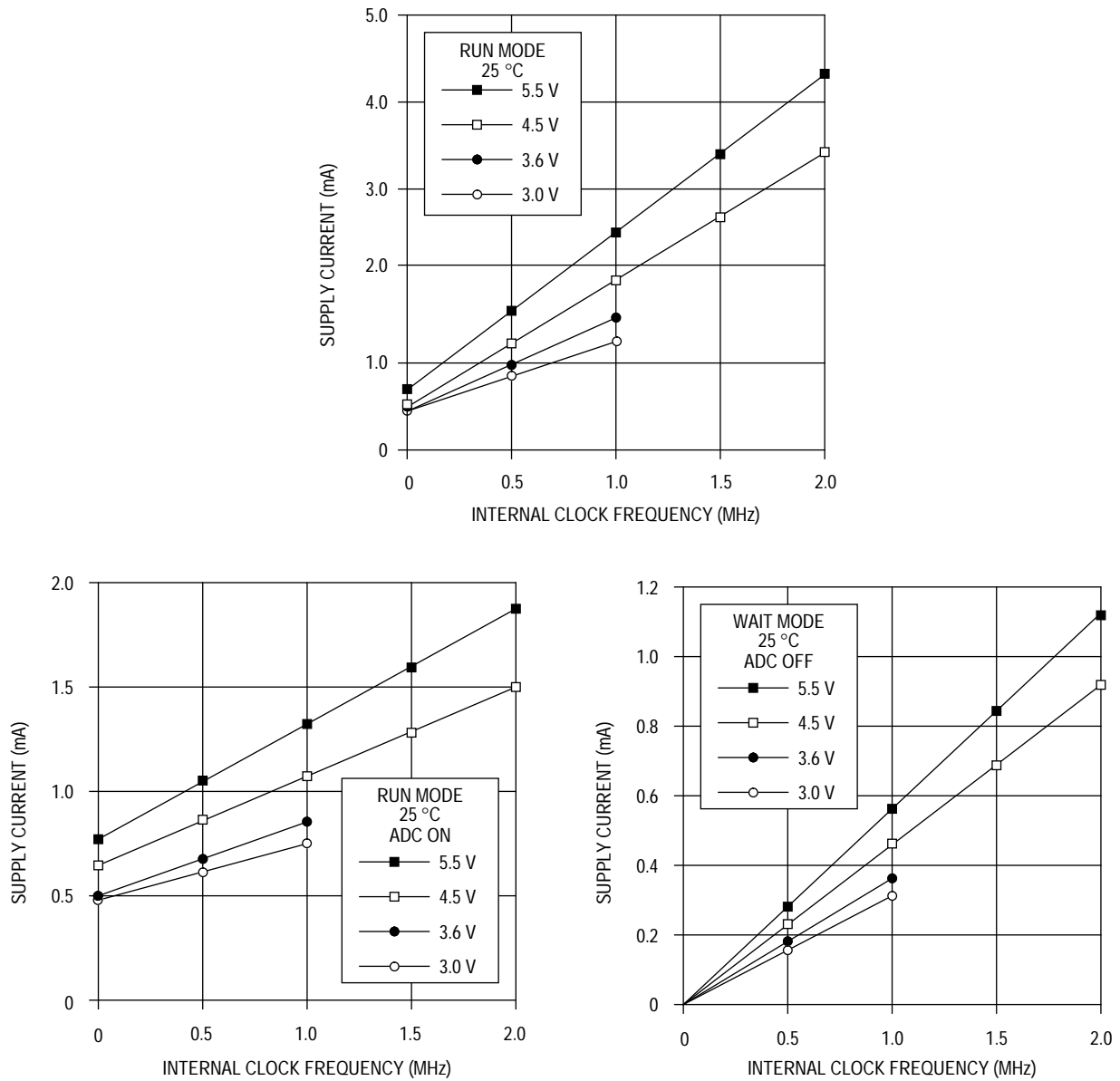


Figure 3. Typical Supply Current vs. Internal Clock Frequency

Maximum Supply Current vs. Internal Clock Frequency

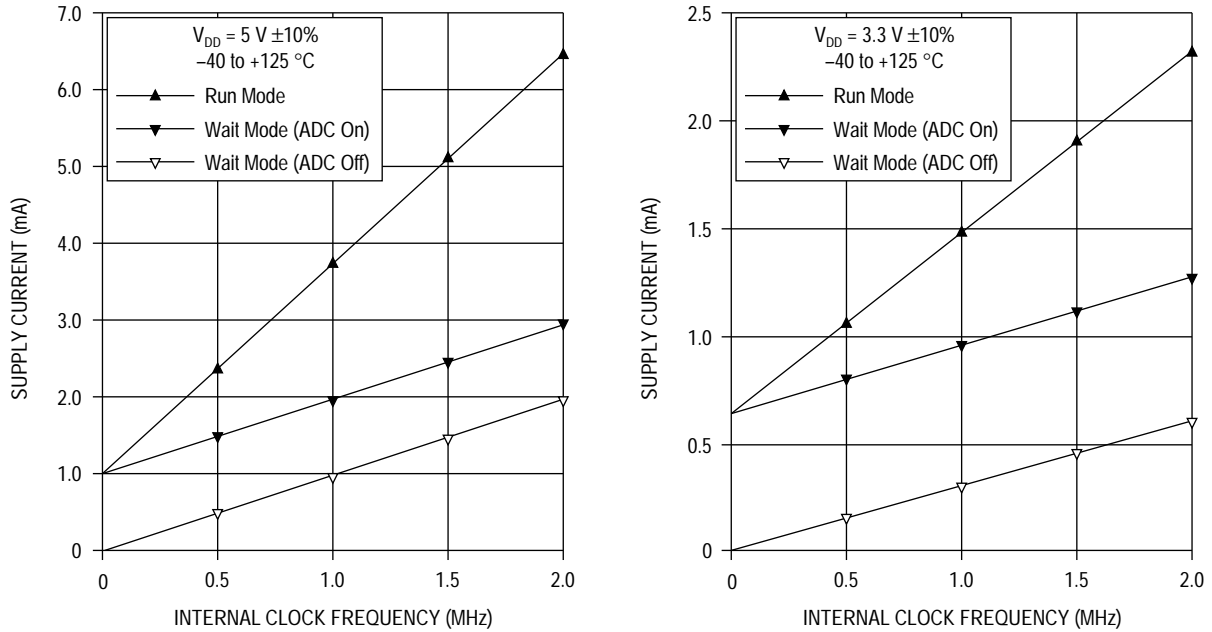


Figure 4. Maximum Supply Current vs. Internal Clock Frequency

5.0 V Control Timing

Table 6. Control Timing ($V_{DD} = 5.0 \text{ Vdc}$)⁽¹⁾

| Characteristic | Symbol | Min | Max | Unit |
|---|--|-----------------------------------|-------------|------------------------------|
| Oscillator Frequency Crystal External Clock | f_{OSC} | — dc | 4.2 4.2 | MHz |
| Internal Operating Frequency ($f_{OSC} \div 2$) Crystal External Clock | f_{OP} | — dc | 2.1 2.1 | MHz |
| Cycle Time ($1 \div f_{OP}$) | t_{CYC} | 480 | — | ns |
| Crystal Oscillator Startup Time | t_{OXOV} | — | 100 | ms |
| Stop Recovery Startup Time (Crystal Oscillator) | t_{ILCH} | — | 100 | ms |
| RESET Pulse Width | t_{RL} | 1.5 | — | t_{CYC} |
| Timer Resolution ⁽²⁾ Input Capture Pulse Width Input Capture Pulse Period | t_{RESL} t_H, t_L t_{TLTL} | 4.0 125 Note ⁽³⁾ | — — — | t_{CYC} ns t_{CYC} |
| Interrupt Pulse Width Low (Edge-Triggered) | t_{ILIH} | 125 | — | ns |
| Interrupt Pulse Period | t_{ILIL} | Note ⁽⁴⁾ | — | t_{CYC} |
| OSC1 Pulse Width | t_{OH}, t_{OL} | 90 | — | ns |
| RC Oscillator Stabilization Time | t_{RCON} | — | 5 | μs |
| ADC On Current Stabilization Time | t_{ADON} | — | 100 | μs |

- $V_{DD} = 5.0 \text{ Vdc} \pm 10\%$, $V_{SS} = 0 \text{ Vdc}$, $T_A = T_L$ to T_H unless otherwise noted
- A 2-bit prescaler in the timer is the limiting factor as it counts 4 t_{CYC}
- The minimum t_{TLTL} should not be less than the number of interrupt service routine cycles plus 19 t_{CYC}
- The minimum t_{ILIL} should not be less than the number of interrupt service routine cycles plus 19 t_{CYC}

3.3 V Control Timing

Table 7. Control Timing ($V_{DD} = 3.3 \text{ Vdc}$)⁽¹⁾

| Characteristic | Symbol | Min | Max | Unit |
|---|--|-----------------------------------|-------------|------------------------------|
| Oscillator Frequency Crystal External Clock | f_{OSC} | — dc | 2.0 2.0 | MHz |
| Internal Operating Frequency ($f_{OSC} \div 2$) Crystal External Clock | f_{OP} | — dc | 1.0 1.0 | MHz |
| Cycle Time ($1 \div f_{OP}$) | t_{CYC} | 1 | — | ms |
| Crystal Oscillator Startup Time | t_{OXOV} | — | 100 | ms |
| Stop Recovery Startup Time (Crystal Oscillator) | t_{LCH} | — | 100 | ms |
| RESET Pulse Width | t_{RL} | 1.5 | — | t_{CYC} |
| Timer Resolution ⁽²⁾ Input Capture Pulse Width Input Capture Pulse Period | t_{RESL} t_H, t_L t_{TLTL} | 4.0 250 Note ⁽³⁾ | — — — | t_{CYC} ns t_{CYC} |
| Interrupt Pulse Width Low (Edge-Triggered) | t_{LIH} | 250 | — | ns |
| Interrupt Pulse Period | t_{LIL} | Note ⁽⁴⁾ | — | t_{CYC} |
| OSC1 Pulse Width | t_{OH}, t_{OL} | 200 | — | ns |

1. $V_{DD} = 3.3 \text{ Vdc} \pm 10\%$, $V_{SS} = 0 \text{ Vdc}$, $T_A = T_L$ to T_H unless otherwise noted
2. A 2-bit prescaler in the timer is the limiting factor as it counts 4 t_{CYC}
3. The minimum t_{TLTL} should not be less than the number of interrupt service routine cycles plus 19 t_{CYC}
4. The minimum t_{LIL} should not be less than the number of interrupt service routine cycles plus 19 t_{CYC}

Test Load

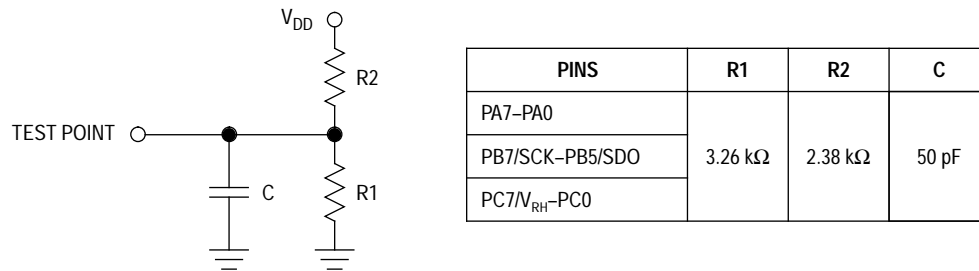


Figure 5. Test Load

Mechanical Specifications

The MC68HC705P9 is available in the following packages:

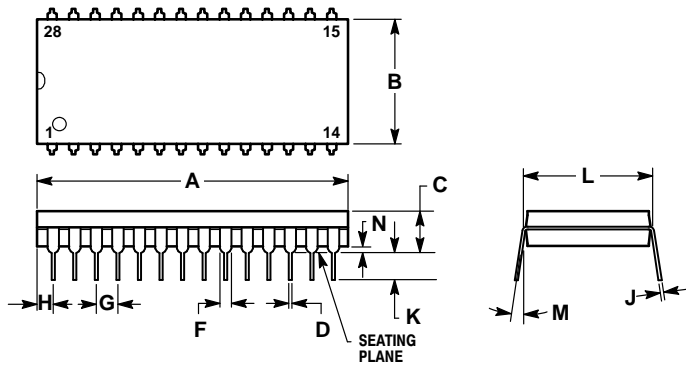
- 710 — Plastic dual in-line package (PDIP)
- 733 — Ceramic dual in-line package (Cerdip)
- 751F — Small outline integrated circuit (SOIC)

The following figures show the latest packages at the time of this publication. To make sure that you have the latest package specifications, contact one of the following:

- Local Motorola Sales Office
- Motorola Mfax
 - Phone 602-244-6609
 - EMAIL rmfax0@email.sps.mot.com
- Worldwide Web (wwweb) at <http://design-net.com>

Follow Mfax or wwweb on-line instructions to retrieve the current mechanical specifications.

28-Pin PDIP —
Case #710

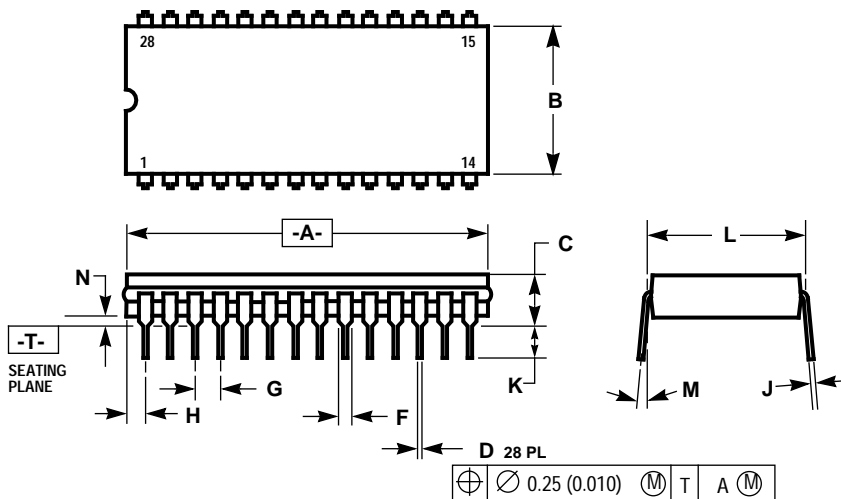


NOTES:

1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

| DIM | MILLIMETERS | | INCHES | |
|-----|-------------|-------|-----------|-------|
| | MIN | MAX | MIN | MAX |
| A | 36.45 | 37.21 | 1.435 | 1.465 |
| B | 13.72 | 14.22 | 0.540 | 0.560 |
| C | 3.94 | 5.08 | 0.155 | 0.200 |
| D | 0.36 | 0.56 | 0.014 | 0.022 |
| F | 1.02 | 1.52 | 0.040 | 0.060 |
| G | 2.54 BSC | | 0.100 BSC | |
| H | 1.65 | 2.16 | 0.065 | 0.085 |
| J | 0.20 | 0.38 | 0.008 | 0.015 |
| K | 2.92 | 3.43 | 0.115 | 0.135 |
| L | 15.24 BSC | | 0.600 BSC | |
| M | 0° | 15° | 0° | 15° |
| N | 0.51 | 1.02 | 0.020 | 0.040 |

28-Pin Cerdip —
Case #733



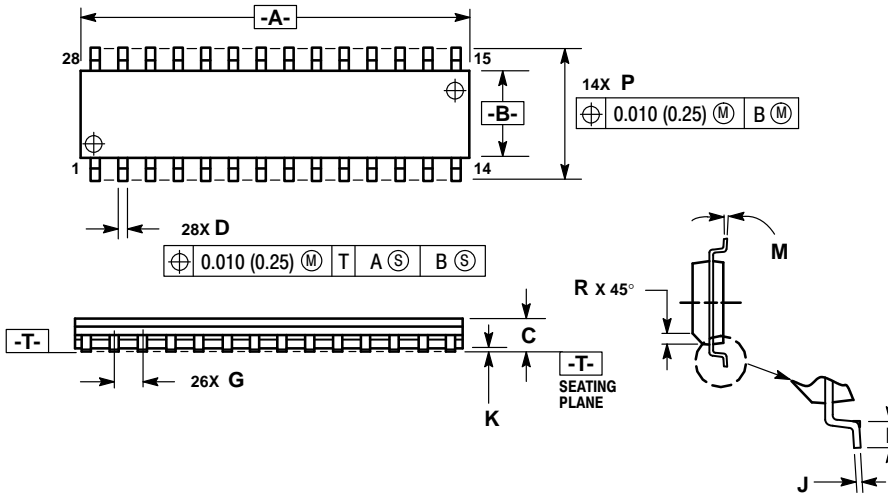
NOTES:

1. DIMENSIONS A AND B INCLUDES MENISCUS.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
4. CONTROLLING DIMENSION: INCH.

| DIM | INCHES | | MILLIMETERS | |
|-----|-----------|-------|-------------|-------|
| | MIN | MAX | MIN | MAX |
| A | 1.435 | 1.490 | 36.45 | 37.84 |
| B | 0.500 | 0.605 | 12.70 | 15.36 |
| C | 0.160 | 0.230 | 4.06 | 5.84 |
| D | 0.015 | 0.022 | 0.38 | 0.55 |
| F | 0.050 | 0.065 | 1.27 | 1.65 |
| G | 0.100 BSC | | 2.54 BSC | |
| J | 0.008 | 0.012 | 0.20 | 0.30 |
| K | 0.125 | 0.160 | 3.18 | 4.06 |
| L | 0.600 BSC | | 15.24 BSC | |
| M | 0° | 15° | 0° | 15° |
| N | 0.020 | 0.050 | 0.51 | 1.27 |

13-mc68hc705p9

28-Pin SOIC —
Case #751F



NOTES:

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

| DIM | MILLIMETERS | | INCHES | |
|-----|-------------|-------|-----------|-------|
| | MIN | MAX | MIN | MAX |
| A | 17.80 | 18.05 | 0.701 | 0.711 |
| 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 | 0.41 | 0.90 | 0.016 | 0.035 |
| G | 1.27 BSC | | 0.050 BSC | |
| J | 0.23 | 0.32 | 0.009 | 0.013 |
| K | 0.13 | 0.29 | 0.005 | 0.011 |
| M | 0° | 8° | 0° | 8° |
| P | 10.05 | 10.55 | 0.395 | 0.415 |
| R | 0.25 | 0.75 | 0.010 | 0.029 |

| A | C |
|--|-------------------------------------|
| accumulator (A) | C bit |
| ADC | case outlines |
| ADC (analog-to-digital converter) | CCF bit |
| block diagram | central processor unit |
| features | ceramic resonator circuit |
| I/O register summary | CH[2:0] bits |
| I/O registers | computer operating properly |
| low-power modes | watchdog |
| ADC data register | condition code register |
| (ADDR) | (CCR) |
| ADC status and control register | COP |
| (ADSCR) | COP register (COPR) |
| addressing modes | COP watchdog |
| ADON bit | COP in stop mode |
| ADRC bit | COP in wait mode |
| alternate timer registers (ATRH/L) | COP register (COPR) |
| ALU | enabling and disabling |
| AN[3:0] bits | features |
| analog-to-digital converter | operation |
| arithmetic/logic unit (ALU) | timeout period |
| | COP watchdog reset |
| B | COPC bit |
| bootloader ROM | COPE bit |
| bootload procedure | CPU |
| bootloader circuit | block diagram |
| location | control unit |
| brownout | features |
| bypass capacitors | instruction set summary |

| | |
|--|--|
| <ul style="list-style-type: none"> instruction types 43 instructions set 40 opcode map 54 registers 36 CPU registers 24, 40, 43, 47, 62 <ul style="list-style-type: none"> accumulator (A) 40, 43 condition code register (CCR) .36, 46, 59, 61–62, 102–105 index register (X) 40–43 program counter (PC) 42, 45, 56, 62 stack pointer (SP) 24 crystal <ul style="list-style-type: none"> AT-cut 16 strip 16 tuning fork 16 crystal oscillator circuit 16 | <ul style="list-style-type: none"> test load 142 thermal characteristics 133 electrostatic damage 72 EPGM bit 26 EPROM erasure 25, 30 EPROM programming register (EPROG) 26 EPROM/OTPROM <ul style="list-style-type: none"> bootloader circuit 28 EPROM erasing 30 locations 25 programming 26 external interrupt 17, 59–60, 88, 106 external interrupt vector 62 external reset 57, 88, 106 |
| D | F |
| <ul style="list-style-type: none"> data direction registers 111–112, 116 <ul style="list-style-type: none"> data direction register A (DDRA) ..17, 74 data direction register B (DDRB) ..17, 77 data direction register C (DDRC) ..17, 80 data direction register D (DDRD) ..18, 83 data-retention mode 70 DCOL bit 119 DDRA[7:0] bits 74 DDRB[7:5] bits 77 DDRC[7:0] bits 80 DDRD5 bit 83 | <ul style="list-style-type: none"> features 10 |
| E | I |
| <ul style="list-style-type: none"> electrical specifications 131 <ul style="list-style-type: none"> control timing 140–141 current versus internal clock <ul style="list-style-type: none"> frequency 138–139 DC electrical characteristics 135–136 driver characteristics 137 maximum ratings 132 operating temperature range 133 power considerations 134 | <ul style="list-style-type: none"> I bit 59, 61–62, 102–105 I/O bits <ul style="list-style-type: none"> ADON bit 127, 130 ADRC bit 124, 127, 130 AN[3:0] bits 79 C bit 46 CCF bit 124, 126 CH[2:0] bits 128 COPC bit 58, 87–88 COPE bit 32 DCOL bit 119 DDRA[7:0] bits 74 DDRB[7:5] bits 77 DDRC[7:0] bits 80 DDRD5 bit 83 EPGM bit 26 I bit 59, 61–62, 102–105 ICF bit 61, 98, 101, 104, 106 ICIE bit 61, 98–99 IEDG bit 100 |

- IRQ bit31
- LATCH bit26
- MSTR bit117
- OCF bit61, 98, 101, 105
- OCIE bit61, 98–99
- OLVL bit94, 100
- PA[7:0] bits73
- PB[7:5] bits76
- PC[7:0] bits79
- PD5 bit82
- PD7 bit82
- SIOP bit31
- SPE bit111–112, 116, 119
- SPIF bit118
- TOF bit62, 98, 101–103
- TOIE bit62, 98–99
- I/O pins
 - IRQ/V_{PP} pin12, 17, 26–27, 31, 59
 - OSC1 pin15
 - OSC2 pin15
 - PB5/SDO pin77, 111–112, 116
 - PB6/SDI pin77, 111, 113, 116
 - PB7/SCK pin76, 111–112, 116
 - PC3/AN0 pin128
 - PC3/AN3 pin79, 123
 - PC4/AN2 pin79, 123, 128
 - PC5/AN1 pin79, 123, 128
 - PC6/AN0 pin79, 123, 128
 - PC7/V_{RH} pin79, 123
 - PD7/TCAP pin27, 82, 93, 100–101
 - RESET pin17, 27, 56–58
 - TCMP pin18, 93–94, 100, 105
 - V_{SS} pin15
- I/O port pin termination72
- I/O registers
 - ADC data register (ADDR)124, 126–127, 129
 - ADC status and control register (ADSCR)124, 126
 - alternate timer registers (ATRH/L) ...103
 - COP register (COPR)58, 87
 - data direction register A (DDRA)74
 - data direction register B (DDRB)77
 - data direction register C (DDRC)80
 - data direction register D (DDRD)83
 - EPROM programming register (EPROG)26
 - input capture registers (ICRH/ICRL)93
 - input capture registers (ICRH/L)100–101, 104, 106
 - mask option register (MOR)12, 25, 31, 86
 - output compare registers (OCRH/L)100–101, 105
 - output compare registers (OCRH/OCRL)94
 - port A data register (PORTA)73
 - port B data register (PORTB)76
 - port C data register (PORTC)79
 - port D data register (PORTD)82
 - SIOP control register (SCR)111, 116, 119
 - SIOP data register (SDR)118–119
 - SIOP status register (SSR)118
 - timer control register (TCR) ...61–62, 99
 - timer registers (TRH/L)100–102
 - timer status register (TSR)61–62, 100, 104–105
- ICF bit61, 98, 101, 104, 106
- ICIE bit61, 98–99
- IEDG bit100
- index register (X)40–43
- input capture interrupt90
- input capture registers (ICRH/L)93, 100–101, 104, 106
- instruction set40
 - addressing modes40
 - instruction set summary48
 - instruction types43
 - opcode map54
- internal clock86
 - frequency15

- internal RC oscillator124, 127
- interrupts55, 98
 - external interrupt59
 - external interrupt logic60
 - external interrupt timing60–61
 - interrupt flowchart64
 - interrupt processing62
 - interrupt sources59
 - interrupt stacking order63
 - reset/interrupt vector addresses63
 - software interrupt59
 - timer interrupts61
- introduction, MC68HC705P99
- IRQ bit31
- IRQ latch59
- $\overline{\text{IRQ}}/\text{V}_{\text{PP}}$ pin12, 17, 26–27, 31, 59

- J**
- junction temperature134

- L**
- LATCH bit26
- literature updates151
- low voltage protection58
- low-power modes65
 - ADC in stop and wait modes130
 - COP in stop and wait modes88
 - data-retention mode70
 - S1OP in stop and wait modes120
 - STOP instruction flowchart67
 - stop mode65
 - stop recovery timing66
 - STOP/WAIT clock logic70
 - timer in stop and wait modes106
 - WAIT instruction flowchart69
 - wait mode68

- M**
- mask option register (MOR) ...12, 25, 31, 86
- mechanical specifications
 - packages142
- memory19
 - EPROM/OTPROM25
 - features19
 - parallel I/O register summary21
 - RAM24
- MSTR bit117

- N**
- noise15, 17

- O**
- OCF bit61, 98, 101, 105
- OCIE bit61, 98–99
- OLVL bit94, 100
- on-chip oscillator15
 - frequency15
 - stabilization delay56, 88, 120
- opcode map54
- operating temperature12
- operating temperature range133
- options
 - programmable12
- order numbers12
- OSC1 pin15
- OSC2 pin15
- output compare interrupt90, 94
- output compare registers
 - (OCRH/L)94, 100–101, 105

- P**
- PA[7:0] bits73
- package dimensions
 - Cerdip143
 - PDIP143
 - SOIC144
- package types12
- parallel I/O ports71
- PB[7:5] bits76
- PB5/SDO pin77, 111–112, 116

- PB6/SDI pin77, 111, 113, 116
 - PB7/SCK pin76, 111–112, 116
 - PB7SCK pin111
 - PC[7:0] bits79
 - PC3/AN3 pin79, 123, 128
 - PC4/AN2 pin79, 123, 128
 - PC5/AN1 pin79, 123, 128
 - PC6/AN0 pin79, 123, 128
 - PC7/V_{RH} pin79, 123
 - PD5 bit82
 - PD7 bit82
 - PD7/TCAP pin27, 82, 93, 100–101
 - pin assignments14
 - pin descriptions13
 - pin functions15
 - port A17, 73
 - data direction register A (DDRA)74
 - port A data register (PORTA)73
 - port B17, 76
 - data direction register B (DDRB)77
 - port B data register (PORTB)76
 - port C17, 79
 - data direction register C (DDRC)80
 - port C data register (PORTC)79
 - port D18, 82
 - data direction register D (DDRD)83
 - port D data register (PORTD)82
 - ports72
 - parallel I/O port register summary72
 - port A73
 - port B76
 - port C79
 - port D82
 - power dissipation128, 134
 - power supply (V_{DD})15, 30
 - power supply (V_{PP})29
 - power-on reset56
 - program counter (PC)42, 45, 56, 62
 - programmable options
 - COP watchdog enable/disable12, 86
 - external interrupt pin triggering12, 31
 - SIOP data format12, 110
- Q**
- quartz window25
- R**
- RAM
 - locations24
 - stack24
 - Reading102
 - registers
 - ADC I/O register summary123
 - CPU registers36
 - parallel I/O port register summary72
 - parallel I/O register summary21
 - SIOP I/O register summary109
 - timer I/O register summary92
 - RESET pin17, 27, 56–58
 - reset sources
 - COP watchdog56
 - power-on56
 - RESET pin56
 - reset vector56
 - resets55–56
 - COP watchdog reset58, 86
 - COP watchdog reset operation86
 - external reset57
 - external reset timing57
 - low-voltage protection reset58
 - power-on reset (POR)56
 - power-on reset (POR) timing57
 - reset sources56
 - reset/interrupt vector addresses63
 - resets and interrupts55
- S**
- serial input/output port107
 - SIOP107

- SIOPI (serial input/output port)
 - block diagram109
 - description108
 - features108
 - I/O register summary109
 - I/O registers116
 - low-power modes120
 - operation110
 - timing114–115
 - SIOPI bit31
 - SIOPI control register (SCR) ...111, 116, 119
 - SIOPI data register (SDR)118–119
 - SIOPI status register (SSR)118
 - software failure86
 - software interrupt vector62
 - SPE bit111–112, 116, 119
 - specifications131
 - See "electrical specifications."132
 - See "mechanical specifications."142
 - SPIFI bit118
 - stack pointer (SP)24
 - stack RAM24, 62
 - stop mode65
 - effect on ADC130
 - effect on capture/compare timer106
 - effect on COP watchdog88
 - effect on SIOPI120
 - STOP instruction flowchart67
 - stop recovery timing66
 - STOP/WAIT clock logic70
 - supply voltage (V_{DD})86, 132
- T**
- TCMP pin18, 93–94, 100, 105
 - thermal resistance133–134
 - timer89
 - block diagram91
 - features90
 - I/O register summary92
 - I/O registers98
 - interrupts61, 98
 - low-power modes106
 - operation93
 - reading93, 103
 - timing95–97
 - timer control register (TCR)61–62, 99
 - timer interrupt vector62
 - timer registers (TRH/L)100–102
 - timer resolution93
 - timer status register (TSR)61–62, 100, 104–105
 - TOFI bit62, 98, 101–103
 - TOIE bit62, 98–99
- V**
- V_{DD} power supply15, 30
 - V_{PP} power supply29
 - V_{SS} pin15
- W**
- wait mode68
 - effect on ADC130
 - effect on capture/compare timer106
 - effect on COP watchdog88
 - effect on SIOPI120
 - STOP/WAIT clock logic70
 - WAIT instruction flowchart69

This document contains the latest data available at publication time. For updates, contact one of the following:

Literature Distribution Centers

Call or order literature by mail.

USA/Europe: Motorola Literature Distribution
P.O. Box 20912
Phoenix, Arizona 85036
Phone 1 800 441-2447 or 602 303-5454

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

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

Mfax

Call to access this worldwide faxing service.

RMFAX0@email.sps.mot.com

TOUCH-TONE 602 244-6609

Or, on the <http://Design-NET.com> home page, select the Mfax icon. Obtain a fax of complete, easy-to-use Mfax instructions by entering your FAX number and then pressing the 1 key.

Motorola SPS World Marketing World Wide Web Server

Use the Internet to access this World Wide Web Server. Use the following URL:

<http://Design-net.com>

CSIC Microcontroller Division's Web Site

Directly access the CSIC Microcontroller Division's web site with the following URL:

http://Design-net.com/csic/CSIC_home.html



Please help us to continue improving the quality and usefulness of our data books by filling out this form and sending your comments to us. You can return the form by mail, or FAX it to 512-891-3236.

Thank you for your help and continued support!

1. How do you rate the quality of this data book?

| | High | | | Low | |
|--------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Organization | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Readability | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Accuracy | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Figures | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

| | High | | | Low | |
|--------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Tables | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Table of contents | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Page size/binding | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Overall impression | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

2. Are you able to find the information you need easily?

| Yes | No | | | |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

3. What are your recommendations for making this data book more useful?

4. What additional information would you like to see included in future data books?

PLEASE
PASTE
POSTAGE
HERE

Motorola
6501 William Cannon Drive West
Mail Stop OE17
Austin, Texas 78735-8598
USA

Attention: CSIC Publications Department



Second: fold back along this line

First: cut along this line to remove

Please supply the following information (optional).

Name: _____

Company Name: _____


Title: _____

Address: _____

City: _____ State: _____ Zip: _____

Phone Number: _____



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

How to reach us:

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

INTERNET: <http://Design-net.com>

USA/EUROPE: Motorola Literature Distribution; P.O. Box 20912; Phoenix, Arizona 85036. 1-800-441-2447

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

HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park, 51 Ting Tok Road, Tai Po, N.T., Hong Kong. 852-26629298