

Freescale Semiconductor

HCO5

MC68HC05P4

TECHNICAL
DATA





Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**



Freescale Semiconductor, Inc.

MC68HC05P4
HCMOS MICROCONTROLLER UNIT

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**



Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**

TABLE OF CONTENTS

| Paragraph Number | Title | Page Number |
|------------------|-------|-------------|
|------------------|-------|-------------|

**Section 1
Introduction**

**Section 2
Pin Descriptions**

| | | |
|-------|---|-----|
| 2.1 | V _{DD} and V _{SS} | 2-1 |
| 2.2 | Oscillator Inputs (OSC1, OSC2) | 2-2 |
| 2.2.1 | Crystal (XTAL)..... | 2-2 |
| 2.2.2 | Ceramic Resonator..... | 2-2 |
| 2.2.3 | External Clock..... | 2-3 |
| 2.2.4 | RC Oscillator..... | 2-4 |
| 2.3 | $\overline{\text{RESET}}$ | 2-4 |
| 2.4 | Interrupt Request ($\overline{\text{IRQ}}$) | 2-4 |
| 2.5 | I/O Port Function..... | 2-5 |
| 2.6 | Port A..... | 2-6 |
| 2.7 | Port B and Serial I/O Port (SIOP)..... | 2-7 |
| 2.8 | Port C | 2-7 |
| 2.9 | Port D and Timer Capture (TCAP) | 2-8 |
| 2.10 | Timer Compare (TCMP) | 2-8 |

**Section 3
Central Processor Unit**

| | | |
|---------|------------------------------------|-----|
| 3.1 | CPU Registers..... | 3-1 |
| 3.1.1 | Accumulator (A)..... | 3-2 |
| 3.1.2 | Index Register (X)..... | 3-2 |
| 3.1.3 | Stack Pointer (SP)..... | 3-3 |
| 3.1.4 | Program Counter (PC)..... | 3-4 |
| 3.1.5 | Condition Code Register (CCR)..... | 3-4 |
| 3.1.5.1 | Half-Carry Bit (H-Bit)..... | 3-5 |

| Paragraph Number | Title | Page Number |
|---------------------|--|----------------|
| 3.1.5.2 | Interrupt Mask (I-Bit)..... | 3-5 |
| 3.1.5.3 | Negative Bit (N-Bit) | 3-5 |
| 3.1.5.4 | Zero Bit (Z-Bit)..... | 3-5 |
| 3.1.5.5 | Carry/Borrow Bit (C-Bit)..... | 3-6 |
| 3.2 | Arithmetic/Logic Unit (ALU) and CPU Control..... | 3-6 |
| 3.3 | Addressing Modes..... | 3-6 |
| 3.3.1 | Inherent..... | 3-7 |
| 3.3.2 | Immediate..... | 3-8 |
| 3.3.3 | Direct..... | 3-8 |
| 3.3.4 | Extended | 3-10 |
| 3.3.5 | Indexed, No Offset | 3-11 |
| 3.3.6 | Indexed, 8-Bit Offset | 3-11 |
| 3.3.7 | Indexed, 16-Bit Offset..... | 3-11 |
| 3.3.8 | Relative..... | 3-12 |
| 3.4 | Instruction Set..... | 3-13 |
| 3.4.1 | Register/Memory Instructions | 3-14 |
| 3.4.2 | Read-Modify-Write Instructions | 3-15 |
| 3.4.3 | Jump/Branch Instructions..... | 3-15 |
| 3.4.4 | Bit Manipulation Instructions..... | 3-17 |
| 3.4.5 | Control Instructions..... | 3-17 |
| 3.4.6 | Instruction Set Summary..... | 3-18 |
| 3.4.7 | Opcode Map | 3-23 |
| 3.5 | Low-Power Modes..... | 3-24 |
| 3.5.1 | STOP Mode | 3-24 |
| 3.5.2 | WAIT Mode..... | 3-25 |

**Section 4
Resets and Interrupts**

| | | |
|-------|---|-----|
| 4.1 | Resets | 4-1 |
| 4.1.1 | Power-On Reset (POR)..... | 4-2 |
| 4.1.2 | External $\overline{\text{RESET}}$ Input..... | 4-2 |
| 4.1.3 | Computer Operating Properly (COP) Reset..... | 4-2 |
| 4.2 | Interrupts..... | 4-3 |
| 4.2.1 | External Interrupt..... | 4-4 |
| 4.2.2 | Software Interrupt (SWI)..... | 4-6 |
| 4.2.3 | Capture/Compare Timer Interrupt..... | 4-6 |

| Paragraph Number | Title | Page Number |
|---------------------|-------|----------------|
|---------------------|-------|----------------|

Section 5
Memory

| | | |
|-----|--------------------------|-----|
| 5.1 | Memory Map..... | 5-1 |
| 5.2 | Data-Retention Mode..... | 5-4 |

Section 6
Capture/Compare Timer

| | | |
|-----|-----------------------------------|------|
| 6.1 | Input Capture Operation..... | 6-3 |
| 6.2 | Output Compare Operation..... | 6-3 |
| 6.3 | Timer Counter..... | 6-4 |
| 6.4 | Output Compare Register..... | 6-6 |
| 6.5 | Input Capture Register..... | 6-7 |
| 6.6 | Timer Status Register (TSR)..... | 6-8 |
| 6.7 | Timer Control Register (TCR)..... | 6-9 |
| 6.8 | Timer During WAIT Mode..... | 6-10 |
| 6.9 | Timer During STOP Mode..... | 6-10 |

Section 7
Serial I/O Port System (SIOP)

| | | |
|-------|----------------------------------|-----|
| 7.1 | SIOP Pin Descriptions..... | 7-3 |
| 7.1.1 | SIOP Clock..... | 7-3 |
| 7.1.2 | SIOP Data Input..... | 7-4 |
| 7.1.3 | SIOP Data Output..... | 7-4 |
| 7.2 | SIOP Control Register (SCR)..... | 7-5 |
| 7.3 | SIOP Status Register (SSR)..... | 7-5 |
| 7.4 | SIOP Data Register (SDR)..... | 7-6 |

Section 8
Self-Check Mode

| | | |
|-----|-------------------------|-----|
| 8.1 | Self-Check Circuit..... | 8-1 |
| 8.2 | Self-Check Results..... | 8-1 |

| Paragraph Number | Title | Page Number |
|---------------------|-------|----------------|
|---------------------|-------|----------------|

**Section 9
Electrical Specifications**

| | | |
|-----|--|------|
| 9.1 | Maximum Ratings | 9-1 |
| 9.2 | Thermal Characteristics..... | 9-1 |
| 9.3 | Power Considerations | 9-2 |
| 9.4 | DC Electrical Characteristics ($V_{DD} = 5.0$ Vdc)..... | 9-3 |
| 9.5 | DC Electrical Characteristics ($V_{DD} = 3.3$ Vdc)..... | 9-4 |
| 9.6 | Control Timing ($V_{DD} = 5.0$ Vdc) | 9-7 |
| 9.7 | Control Timing ($V_{DD} = 3.3$ Vdc)..... | 9-9 |
| 9.8 | SIOPI Timing ($V_{DD} = 5.0$ Vdc)..... | 9-11 |
| 9.9 | SIOPI Timing ($V_{DD} = 3.3$ Vdc)..... | 9-12 |

**Section 10
Mechanical Specifications**

| | | |
|------|---|------|
| 10.1 | Dual In-Line Package (DIP)..... | 10-1 |
| 10.2 | Small Outline Integrated Circuit (SOIC) | 10-2 |

**Section 11
Ordering Information**

| | | |
|--------|------------------------------------|------|
| 11.1 | ROM Pattern Media | 11-1 |
| 11.1.1 | Flexible Disks | 11-1 |
| 11.1.2 | EPROMs | 11-2 |
| 11.2 | ROM Pattern Verification | 11-2 |
| 11.2.1 | Verification Media..... | 11-2 |
| 11.2.2 | ROM Verification Units (RVUs)..... | 11-2 |
| 11.3 | MC Order Numbers | 11-2 |

LIST OF FIGURES

| Figure Number | Title | Page Number |
|---------------|--|-------------|
| 1-1 | MC68HC05P4 Block Diagram | 1-2 |
| 2-1 | MC68HC05P4 Pin Assignments..... | 2-1 |
| 2-2 | Crystal/Ceramic Resonator Connections..... | 2-3 |
| 2-3 | External Clock Source Connections..... | 2-3 |
| 2-4 | RC Oscillator Connections..... | 2-4 |
| 2-5 | Parallel I/O Port Circuit..... | 2-6 |
| 2-6 | Port A Data Register and DDRA..... | 2-6 |
| 2-7 | Port B Data Register and DDRB..... | 2-7 |
| 2-8 | Port C Data Register and DDRC | 2-8 |
| 2-9 | Port D Data Register and DDRD | 2-8 |
| 3-1 | CPU Block Diagram | 3-1 |
| 3-2 | Programming Model..... | 3-2 |
| 3-3 | Accumulator..... | 3-2 |
| 3-4 | Index Register..... | 3-3 |
| 3-5 | Stack Pointer | 3-3 |
| 3-6 | Program Counter | 3-4 |
| 3-7 | Condition Code Register..... | 3-4 |
| 3-8 | STOP Function Flowchart | 3-24 |
| 3-9 | WAIT Function Flowchart..... | 3-26 |
| 4-1 | COP Control Register..... | 4-3 |
| 4-2 | Interrupt Stacking Order | 4-4 |
| 4-3 | Reset and Interrupt Flowchart..... | 4-5 |
| 4-4 | IRQ Mask Option Logic | 4-6 |
| 5-1 | MC68HC05P4 Memory Map | 5-2 |
| 5-2 | I/O and Control Register Summary..... | 5-3 |
| 6-1 | Capture/Compare Timer Block Diagram..... | 6-2 |
| 6-2 | Input Capture Operation..... | 6-3 |
| 6-3 | Output Compare Operation..... | 6-4 |

| Figure Number | Title | Page Number |
|----------------------|---|--------------------|
| 6-4 | Counter Register/Counter Alternate Register | 6-5 |
| 6-5 | 16-Bit Counter Reads..... | 6-5 |
| 6-6 | Output Compare Register..... | 6-6 |
| 6-7 | Input Capture Register | 6-7 |
| 6-8 | Timer Status Register..... | 6-8 |
| 6-9 | Timer Control Register | 6-9 |
| 7-1 | Serial I/O Port (SIOP) Block Diagram | 7-1 |
| 7-2 | SIOP Shift Register Operation..... | 7-2 |
| 7-3 | SIOP Data/Clock Timing Diagram..... | 7-4 |
| 7-4 | SIOP Control Register..... | 7-5 |
| 7-5 | SIOP Status Register | 7-5 |
| 7-6 | SIOP Data Register | 7-6 |
| 8-1 | Self-Check Circuit Schematic | 8-2 |
| 9-1 | Test Load..... | 9-2 |
| 9-2 | Typical High-Side Driver Characteristics..... | 9-5 |
| 9-3 | Typical Low-Side Driver Characteristics..... | 9-5 |
| 9-4 | Typical Supply Current vs Internal Clock Frequency..... | 9-6 |
| 9-5 | Maximum Supply Current vs Internal Clock Frequency..... | 9-6 |
| 9-6 | TCAP Timing..... | 9-7 |
| 9-7 | STOP Recovery Timing | 9-8 |
| 9-8 | External Interrupt Timing | 9-8 |
| 9-9 | Power-On Reset Timing..... | 9-10 |
| 9-10 | External Reset Timing..... | 9-10 |
| 9-11 | SIOP Timing..... | 9-11 |
| 10-1 | Case 710-02 Dimensions | 10-1 |
| 10-2 | Case 751F-02 Dimensions..... | 10-2 |

LIST OF TABLES

| Table Number | Title | Page Number |
|-------------------------|--|------------------------|
| 2-1 | I/O Pin Functions..... | 2-6 |
| 3-1 | Inherent Addressing Instructions | 3-7 |
| 3-2 | Immediate Addressing Instructions | 3-8 |
| 3-3 | Direct Addressing Instructions..... | 3-9 |
| 3-4 | Extended Addressing Instructions..... | 3-10 |
| 3-5 | Indexed Addressing Instructions..... | 3-12 |
| 3-6 | Relative Addressing Instructions..... | 3-13 |
| 3-7 | Register/Memory Instructions | 3-14 |
| 3-8 | Read-Modify-Write Instructions | 3-15 |
| 3-9 | Jump and Branch Instructions..... | 3-16 |
| 3-10 | Bit Manipulation Instructions..... | 3-17 |
| 3-11 | Control Instructions..... | 3-17 |
| 3-12 | Instruction Set..... | 3-19 |
| 3-13 | Opcode Map | 3-23 |
| 8-1 | Self-Check Results..... | 8-1 |
| 9-1 | Maximum Ratings | 9-1 |
| 9-2 | Thermal Characteristics..... | 9-1 |
| 9-3 | DC Electrical Characteristics ($V_{DD} = 5.0$ Vdc)..... | 9-3 |
| 9-4 | DC Electrical Characteristics ($V_{DD} = 3.3$ Vdc)..... | 9-4 |
| 9-5 | Control Timing ($V_{DD} = 5.0$ Vdc)..... | 9-7 |
| 9-6 | Control Timing ($V_{DD} = 3.3$ Vdc)..... | 9-9 |
| 9-7 | SIOp Timing ($V_{DD} = 5.0$ Vdc)..... | 9-11 |
| 9-8 | SIOp Timing ($V_{DD} = 3.3$ Vdc)..... | 9-12 |
| 11-1 | MC Order Numbers | 11-2 |



Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

MOTOROLA
x

MC68HC05P4

**For More Information On This Product,
Go to: www.freescale.com**

SECTION 1 INTRODUCTION

The MC68HC05P4 high-density complementary metal-oxide semiconductor (HCMOS) microcontroller unit (MCU) is a member of the popular M68HC05 Family of microcontrollers. This high-performance, low-cost MCU is a complete system on a single chip. The MCU features include the following:

- Memory-Mapped Input/Output (I/O)
- 4160 Bytes of On-Chip ROM
- 240 Bytes of On-Chip Self-Check ROM
- 176 Bytes of On-Chip RAM (Contents Saved in Data-Retention Mode)
- 20 Bidirectional I/O Lines plus One Fixed Input and One Timer Output
- Synchronous Serial I/O Port (SIOP) Subsystem
- Fully Static Operation (No Minimum Clock Speed)
- 16-Bit Capture/Compare Timer Subsystem
- STOP, WAIT, and Data-Retention Modes
- Most Significant Bit (MSB) First or Least Significant Bit (LSB) First SIOP Data Format (Selected by Mask Option)
- Computer Operating Properly (COP) Watchdog (Enabled by Mask Option)
- On-chip Oscillator with Crystal and Resistor/Capacitor (RC) Mask Options
- External Interrupt Sensitivity (Selected by Mask Option)
- Single 3.0-Volt to 5.5-Volt Supply
- 8 × 8 Unsigned Multiply Instruction

Figure 1-1 shows the structure of the MC68HC05P4 MCU.

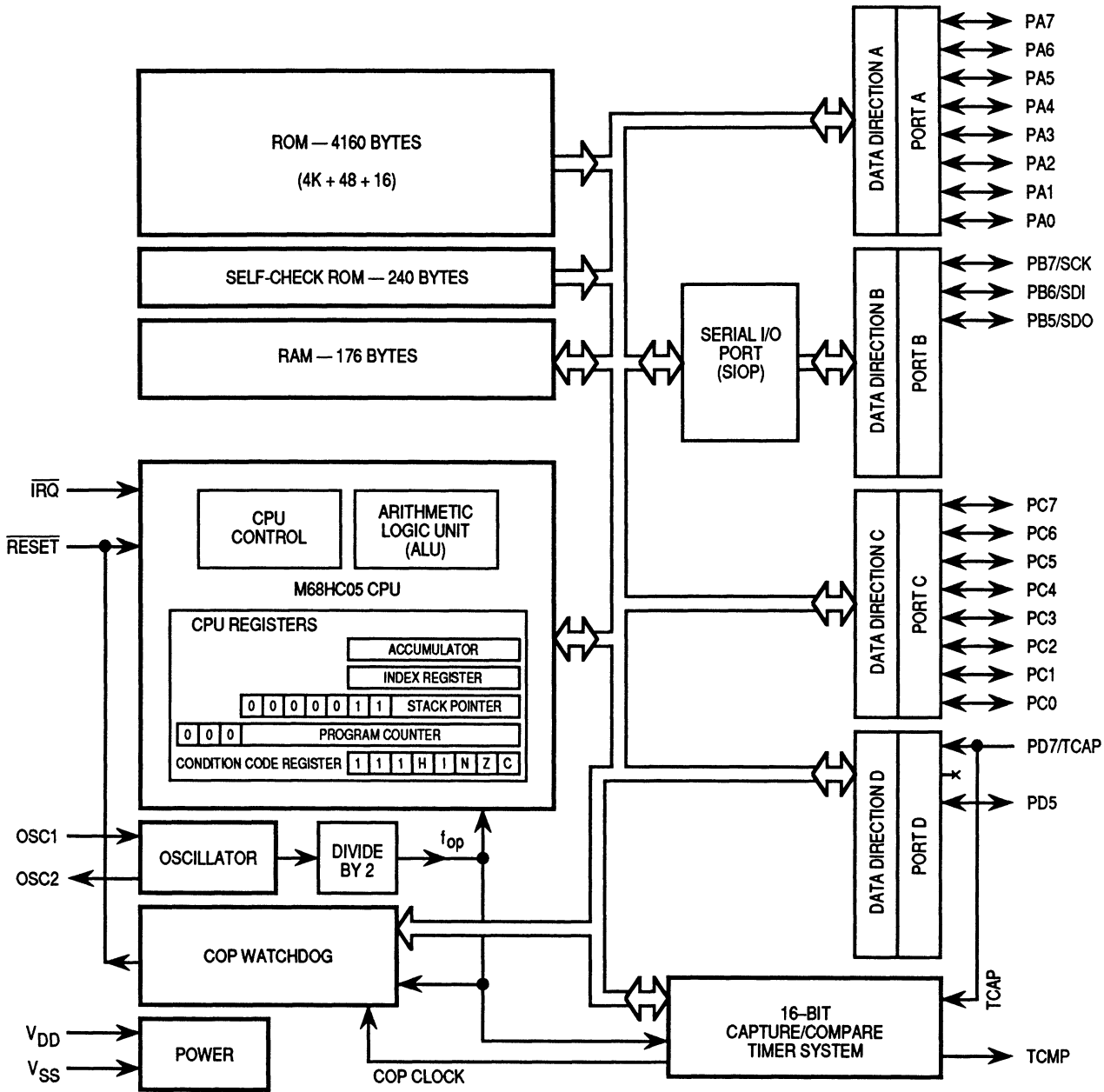


Figure 1-1. MC68HC05P4 Block Diagram

SECTION 2 PIN DESCRIPTIONS

This section shows the MC68HC05P4 pin assignments and describes the function of each pin.

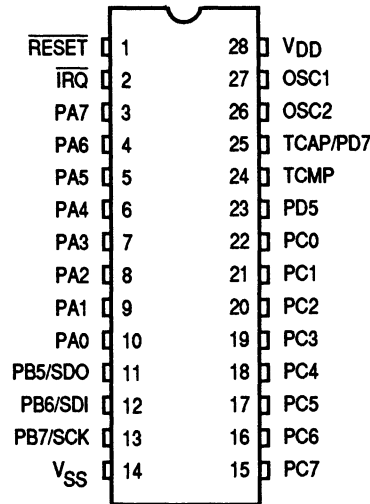


Figure 2-1. MC68HC05P4 Pin Assignments

2.1 VDD and VSS

Power is supplied to the MCU through VDD and VSS. VDD is the power supply, and VSS is ground. The MCU operates from a single 5-volt (nominal) power supply.

Very fast signal transitions occur on the MCU pins. The short rise and fall times place very high short-duration current demands on the power supply. To prevent noise problems, special care must be taken to provide good power supply bypassing at the MCU. Bypass capacitors should have good high-frequency characteristics and be as close to the MCU as possible. Bypassing requirements vary, depending on how heavily the MCU pins are loaded.

Oscillator Inputs (OSC1, OSC2)

OSC1 and OSC2 are the control connections for the internal clock circuit. There are three ways to control the frequency of the internal clock. The OSC1 and OSC2 pins can accept the following:

- A crystal or ceramic resonator (see Figure 2-2)
- An external clock signal connected to OSC1 (see Figure 2-3)
- A resistor between OSC1 and OSC2 to form an RC circuit with an internal capacitor (see Figure 2-4)

A factory-set mask option selects either a crystal/ceramic resonator or a resistor as the frequency-determining element. The frequency (f_{osc}) of the oscillator connected to OSC1 and OSC2 is divided by two to produce the internal operating frequency, f_{op} .

2.2.1 Crystal (XTAL)

The circuit in Figure 2-2 shows a typical crystal oscillator circuit for an AT-cut, parallel resonant crystal. The crystal supplier's recommendations should be followed, since the crystal parameters determine the external component values required to provide maximum stability and reliable starting. The load capacitance values used in the oscillator circuit design should include all stray layout capacitances.

2.2.2 Ceramic Resonator

A ceramic resonator can be used in place of the crystal in cost-sensitive applications. The circuit in Figure 2-2 can be used for a ceramic resonator. The resonator manufacturer's recommendations should be followed, since the resonator parameters determine the external component values required to provide maximum stability and reliable starting. The load capacitance values used in the oscillator circuit design should include all stray layout capacitances.

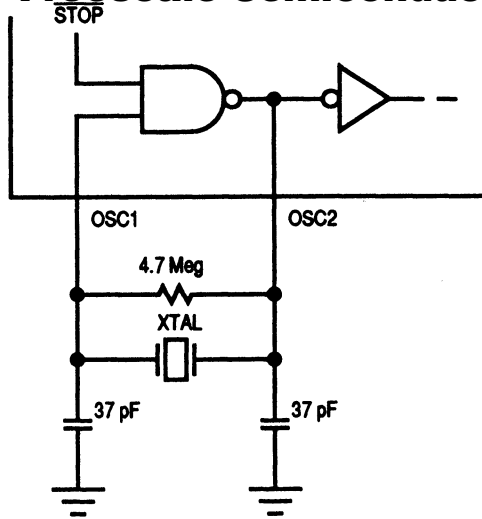


Figure 2-2. Crystal/Ceramic Resonator Connections

Freescale Semiconductor, Inc.

2.2.3 External Clock

An external clock from another CMOS-compatible device can be connected to the OSC1 input, with the OSC2 input not connected, as shown in Figure 2-3. To use an external clock, the crystal/ceramic resonator mask option should be specified when ordering the MCU.

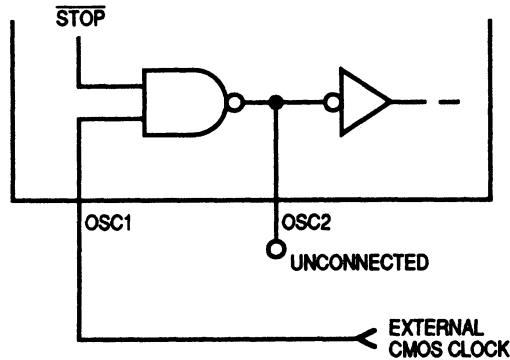


Figure 2-3. External Clock Source Connections

2.2.3 Oscillator

With this option, a resistor is connected to the oscillator pins as shown in Figure 2-4. Since the accuracy of the RC oscillator is $\pm 50\%$, the nominal design frequency must be limited to 66% of the maximum frequency to ensure that the operating frequency remains below the upper limit of operating frequency. This 50% tolerance only allows for the MCU variation, and additional allowance must be made for the tolerances of external components. Operation with a crystal or ceramic resonator is preferred.

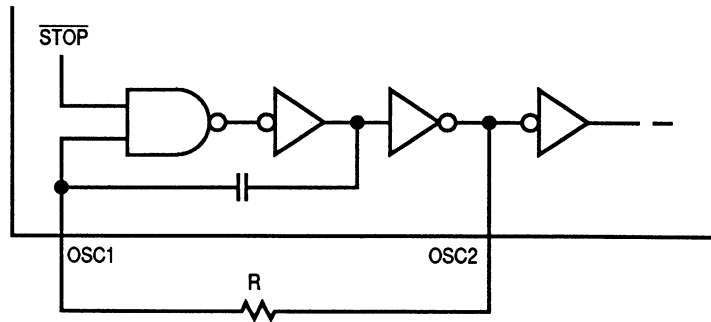


Figure 2-4. RC Oscillator Connections

2.3 $\overline{\text{RESET}}$

$\overline{\text{RESET}}$ is an input-only pin that forces the MCU to a known startup condition. Holding $\overline{\text{RESET}}$ at logical zero for $1.5 t_{\text{cyc}}$ (internal clock cycles) or longer forces the CPU to assume a set of initial conditions. When $\overline{\text{RESET}}$ returns to a logical one, the CPU begins executing instructions from a predetermined starting address.

2.4 Interrupt Request ($\overline{\text{IRQ}}$)

The $\overline{\text{IRQ}}$ pin provides two different choices of interrupt-triggering sensitivity. The factory-set mask options are the following:

- Negative edge-sensitive triggering only
- Both negative edge-sensitive triggering and level-sensitive triggering

In the latter case, either a negative edge or a logical zero level input to the $\overline{\text{IRQ}}$ pin produces an interrupt. The CPU completes the current instruction before it responds to the interrupt request. When the $\overline{\text{IRQ}}$ pin goes to a logical zero level, a small synchronization delay occurs, and a logical one is latched internally to signify that an interrupt is requested. When the CPU completes its current instruction, the interrupt latch is tested. If the interrupt latch contains a logical one, and the interrupt mask (I-bit) in the condition code register is a logical zero, the CPU then begins the interrupt sequence.

(See **4.2 Interrupts** for more detail about interrupts.)

2.5 I/O Port Function

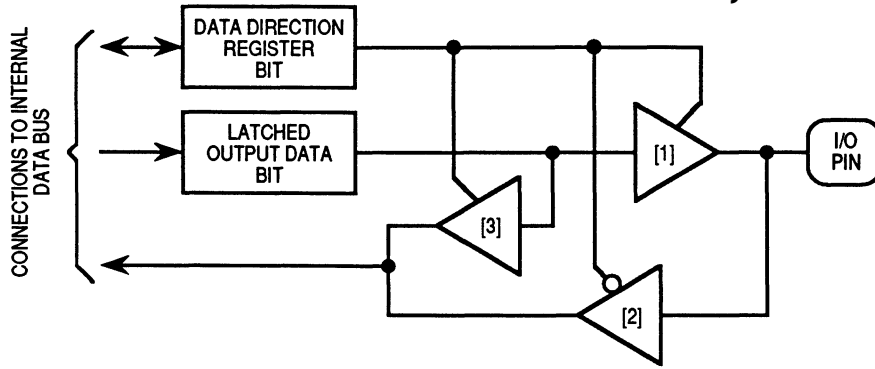
The MCU has 20 I/O pins that form four I/O ports. Each I/O pin is programmable as an input or as an output. Data direction is determined by the contents of the data direction register (DDR) for the port. Writing a logical one to a DDR bit enables the output buffer for that pin; a logical zero disables the output buffer. On reset, all implemented DDR bits are initialized to logical zero to put the pins in the input mode.

NOTE

Any unused inputs and I/O pins should be connected to an appropriate logical level (e.g., either V_{DD} or V_{SS}). Although the I/O ports do not require termination for proper operation, termination is recommended to reduce the possibility of electrostatic damage.

A reset does not initialize the four port data registers. The data registers for ports A, B, C, and D are at addresses \$00, \$01, \$02, and \$03, respectively. To avoid undefined levels, the data registers should be written before writing the DDR bits.

When a pin is programmed to be an output, reading the associated port bit actually reads the value of the output data latch and not the voltage on the pin itself. When a pin is programmed as an input, reading the port bit reads the voltage level on the I/O pin. The output data latch can always be written, regardless of the state of its DDR bit. (See Figure 2-5 for typical port circuitry, and Table 2-1 for a summary of I/O pin functions.)



- [1] Output buffer. Enables latched output to drive pin when DDR bit is 1 (output mode).
- [2] Input buffer. Enabled when DDR bit is 0 (input mode).
- [3] Input buffer. Enabled when DDR bit is 1 (output mode).

Figure 2-5. Parallel I/O Port Circuit

Table 2-1. I/O Pin Functions

| R/ \bar{W} | DDR | I/O Pin Functions |
|--------------|-----|---|
| 0 | 0 | The I/O pin is in input mode. Data is written into the output data latch. |
| 0 | 1 | Data is written into the output data latch, which drives the I/O pin. |
| 1 | 0 | The state of the I/O pin is read. |
| 1 | 1 | The I/O pin is in output mode. The output data latch is read. |

NOTE: \bar{R}/\bar{W} is an internal signal.

2.6 Port A

PA7–PA0 form an 8-bit general-purpose bidirectional I/O port. The contents of data direction register A (DDRA) determine whether each pin is an input or an output. Figure 2-6 shows the port A data register and DDRA.

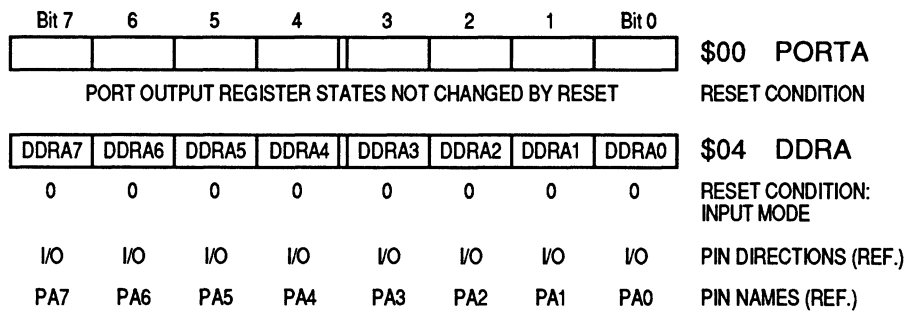


Figure 2-6. Port A Data Register and DDRA

PB7/SCK (serial clock), PB6/SDI (serial data input), and PB5/SDO (serial data output) form a 3-bit shared-function I/O port. Port B can be either the SIO or a general-purpose I/O port. Figure 2-7 shows the port B data register and data direction register B (DDRB). Bits 4–0 of these registers are not implemented.

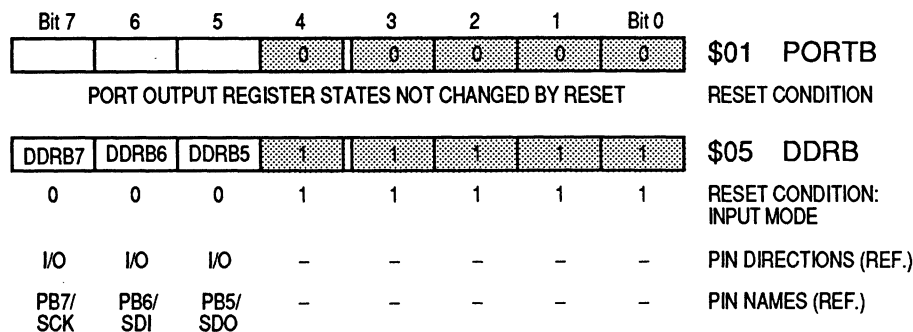


Figure 2-7. Port B Data Register and DDRB

The SIO is a three-wire master/slave system. When the SIO is enabled, PB7 (SCK) serves as a clock output in master mode or as a clock input in slave mode. PB6 is the SDI, and PB5 is the SDO. User software can change the settings of DDRB7–5 to override these defaults if necessary. The SIO data format is selectable as MSB first or LSB first by a factory-set mask option.

These same pins may be used as a general-purpose I/O port when the SIO system is disabled. DDRB7–5 determine the data direction of PB7, PB6, and PB5 (input or output).

(See **SECTION 7 SERIAL I/O PORT SYSTEM (SIO)** for more details.)

2.8 Port C

PC7–PC0 form an 8-bit general-purpose bidirectional I/O port. The contents of data direction register C (DDRC) determine whether each pin is an input or an output. Figure 2-8 shows the port C data register and DDRC.

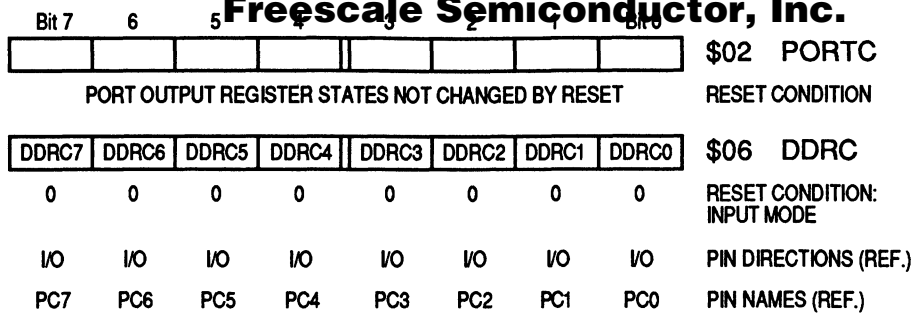


Figure 2-8. Port C Data Register and DDRC

2.9 Port D and Timer Capture (TCAP)

PD7/TCAP and PD5 form a 2-bit special-function I/O port. The PD7/TCAP pin serves both as the edge-detecting input capture line for the capture/compare timer and as a general-purpose digital input. PD7/TCAP can be used as a digital input even when the timer is using it as the input capture pin. There is no output driver associated with the PD7 pin. PD5 is a general-purpose digital I/O pin whose direction is controlled by bit 5 of DDRD. Figure 2-9 shows the port D data register and DDRD.

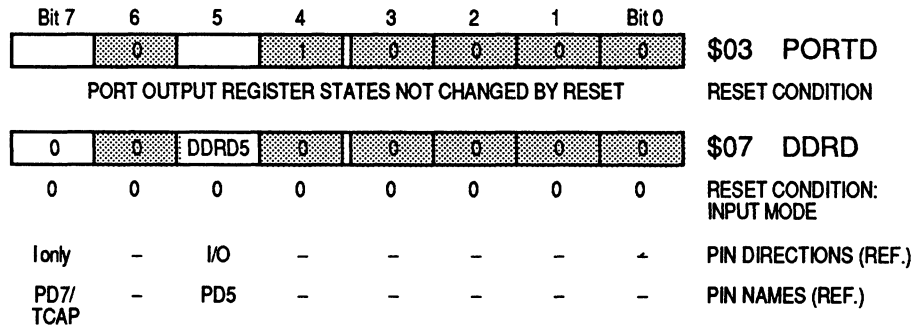


Figure 2-9. Port D Data Register and DDRD

The TCAP pin controls the input capture feature of the capture/compare timer. (See **SECTION 6 CAPTURE/COMPARE TIMER** for more information.)

2.10 Timer Compare (TCMP)

The TCMP pin is the output pin for the output compare feature of the capture/compare timer. (See **SECTION 6 CAPTURE/COMPARE TIMER** for more information.)

SECTION 3 CENTRAL PROCESSOR UNIT

This section describes the registers, instruction set, and addressing modes of the M68HC05 CPU. The STOP and WAIT modes are initiated by software instructions and are also described in this section.

The M68HC05 CPU executes all instructions of the earlier M6805 and M146805 instruction sets and is upgraded to include an 8 × 8 bit unsigned multiply instruction.

Freescale Semiconductor, Inc.

3.1 CPU Registers

The CPU contains the following five registers:

- Accumulator (A)
- Index register (X)
- Stack pointer (SP)
- Program counter (PC)
- Condition code register (CCR)

The CPU registers are hard-wired within the CPU and are not part of the memory map. Figure 3-1 is a block diagram of the MC68HC05 CPU.

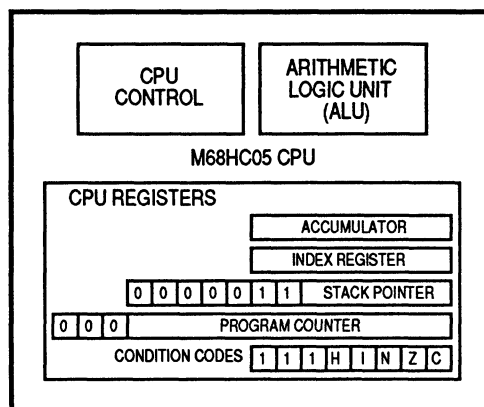


Figure 3-1. CPU Block Diagram

Figure 3-2 shows the five CPU registers.

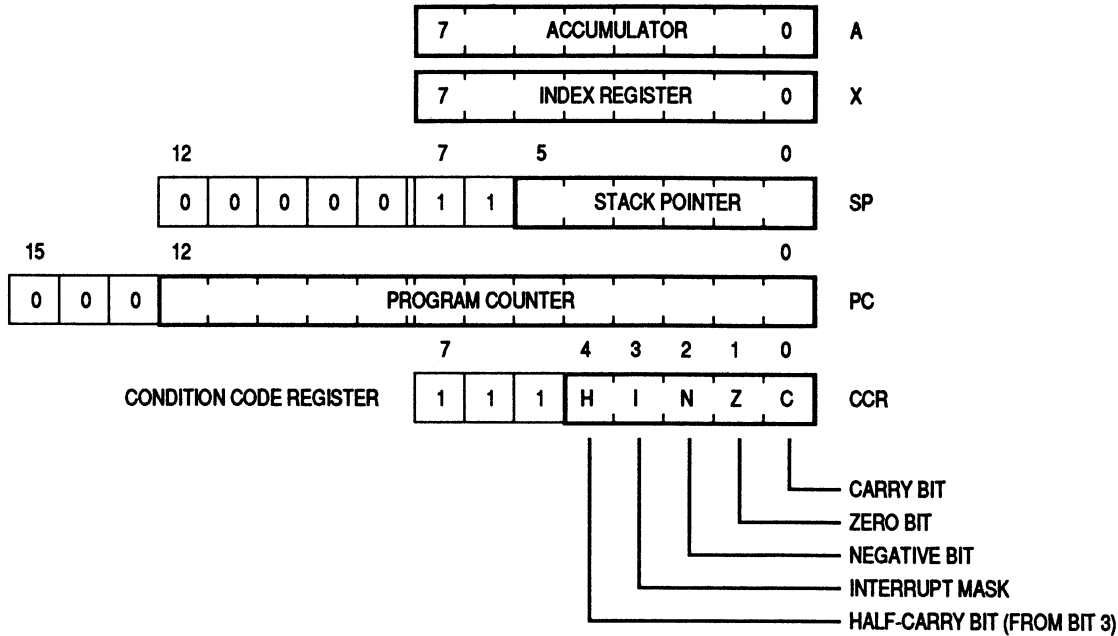


Figure 3-2. Programming Model

3.1.1 Accumulator (A)

The accumulator is a general-purpose 8-bit register. The CPU uses the accumulator to hold operands and results of arithmetic and nonarithmetic operations. (See Figure 3-3.)



Figure 3-3. Accumulator

3.1.2 Index Register (X)

The index register is an 8-bit register that can perform two functions:

- Indexed addressing
- Temporary storage

Freescale Semiconductor, Inc.

in indexed addressing with no offset, the index register contains the low byte of the operand address, and the high byte is assumed to be \$00. In indexed addressing with an 8-bit offset, the CPU finds the operand address by adding the index register contents to an 8-bit immediate value. In indexed addressing with a 16-bit offset, the CPU finds the operand address by adding the index register contents to a 16-bit immediate value. (See **3.3 Addressing Modes**.)

The index register can also serve as an auxiliary accumulator for temporary storage. (See Figure 3-4.)

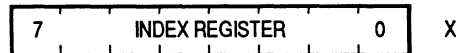


Figure 3-4. Index Register

3.1.3 Stack Pointer (SP)

The stack pointer is a 13-bit register that contains the address of the next free location on the stack. During a reset or after the reset stack pointer (RSP) instruction, the stack pointer contents are set to \$FF. The address in the stack pointer is decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the seven most significant bits of the stack pointer are permanently set to 000011. (See Figure 3-5.) These seven bits are appended to the six least significant register bits to produce an address within the range of \$FF-\$C0. Subroutines and interrupts may use up to 64 stack locations. If 64 locations are exceeded, the stack pointer wraps around and writes over the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.

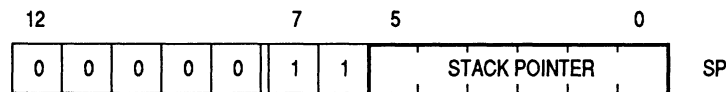


Figure 3-5. Stack Pointer

3.1.4 Program Counter (PC)

The program counter is a 13-bit register that contains the address of the next instruction or operand to be fetched. Since addresses are often 16-bit values, the program counter may be thought of as having three additional upper bits that are always zeros. (See Figure 3-6.)

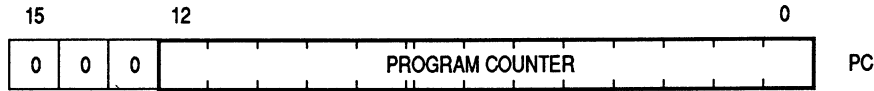


Figure 3-6. Program Counter

Normally, the address in the program counter 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.

3.1.5 Condition Code Register (CCR)

The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. A fifth bit is the interrupt mask. (See Figure 3-7.) These bits can be individually tested by a program, and specific actions can be taken as a result of their states. The condition code register should be thought of as having three additional upper bits that are always ones.

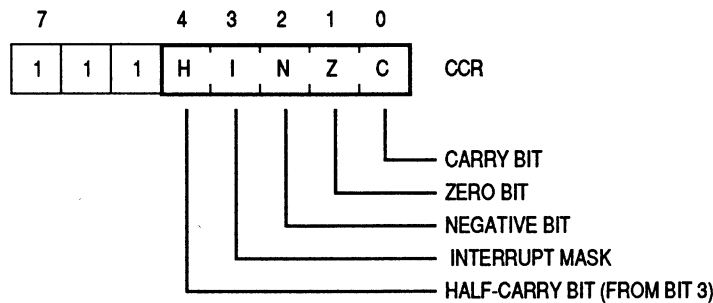


Figure 3-7. Condition Code Register

The following paragraphs explain the functions of the lower five bits of the condition code register.

3.1.5.1 Half-Carry Bit (H-Bit)

When the half-carry bit is set, it means that a carry occurred between bits 3 and 4 of the accumulator during the last ADD or ADC (add with carry) operation. The half-carry bit is required for binary-coded decimal (BCD) arithmetic operations.

3.1.5.2 Interrupt Mask (I-Bit)

When the interrupt mask is set, capture/compare timer interrupts and external interrupts are disabled. Interrupts are enabled when the interrupt mask is cleared. When an interrupt occurs, the interrupt mask is automatically set after the CPU registers are saved on the stack, but before the interrupt vector is fetched. If an interrupt occurs while the interrupt mask is set, the interrupt is latched. Normally, the interrupt is processed as soon as the interrupt mask is cleared.

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 only be cleared by a software instruction.

3.1.5.3 Negative Bit (N-Bit)

The negative is set when the result of the last arithmetic operation, logical operation, or data manipulation was negative. (Bit 7 of the result was a logical one.)

The negative bit can also be used to check an often-tested flag by assigning the flag to bit 7 of a register or memory location. Loading the accumulator with the contents of that register or location then sets or clears the negative bit according to the state of the flag.

3.1.5.4 Zero Bit (Z-Bit)

The zero bit is set when the result of the last arithmetic operation, logical operation, or data manipulation was zero.

Carry/Borrow Bit (C-Bit)

The carry/borrow bit is set when a carry out of bit 7 of the accumulator occurred during the last arithmetic operation, logical operation, or data manipulation. The carry/borrow bit is also set or cleared during bit test and branch instructions and during shifts and rotates.

3.2 Arithmetic/Logic Unit (ALU) and CPU Control

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

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

The CPU control circuitry sequences the logic elements of the ALU to carry out the required operations.

3.3 Addressing Modes

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

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

3.3.1 Inherent

The inherent addressing mode is used for instructions with no operand (e.g., STOP) and for some of the instructions that act on data in the CPU registers (e.g., CLRA). No memory address is required for inherent instructions. Inherent instructions are one byte long. Table 3-1 lists the instructions that can be used in the inherent addressing mode.

Table 3-1. Inherent Addressing Instructions

| Instruction | Mnemonic |
|---|-----------------|
| Arithmetic Shift Left | ASLA, ASLX |
| Arithmetic Shift Right | ASRA, ASRX |
| Clear Carry Bit | CLC |
| Clear Interrupt Mask | CLI |
| Clear | CLRA, CLRX |
| Complement | COMA, COMX |
| Decrement | DECA, DECX |
| Increment | INCA, INCX |
| Logical Shift Left | LSLA, LSLX |
| Logical Shift Right | LSRA, LSRX |
| Multiply | MUL |
| Negate | NEGA, NEGX |
| No Operation | NOP |
| Rotate Left through Carry Bit | ROLA, ROLX |
| Rotate Right through Carry Bit | RORA, RORX |
| Reset Stack Pointer | RSP |
| Return from Interrupt | RTI |
| Return from Subroutine | RTS |
| Set Carry Bit | SEC |
| Set Interrupt Mask | SEI |
| Enable \overline{IRQ} and Stop Oscillator | STOP |
| Software Interrupt | SWI |
| Transfer Accumulator to Index Register | TAX |
| Test for Negative or Zero | TSTA, TSTX |
| Transfer Index Register to Accumulator | TXA |
| Enable Interrupt and Halt Processor | WAIT |

3.3.2 Immediate

The immediate addressing mode is used for instructions that contain a value to be used in an operation with the value in the accumulator or index register. No memory address is required for immediate instructions. The operand is contained in the byte immediately following the opcode. These are two-byte instructions, one for the opcode and one for the immediate data byte. Table 3-2 lists the instructions that can be used in the immediate addressing mode.

Table 3-2. Immediate Addressing Instructions

| Instruction | Mnemonic |
|--------------------------------------|----------|
| Add with Carry | ADC |
| Add | ADD |
| Logical AND | AND |
| Bit Test Memory with Accumulator | BIT |
| Compare Accumulator with Memory | CMP |
| Compare Index Register with Memory | CPX |
| Exclusive OR Memory with Accumulator | EOR |
| Load Accumulator from Memory | LDA |
| Load Index Register from Memory | LDX |
| Inclusive OR | ORA |
| Subtract with Carry | SBC |
| Subtract | SUB |

3.3.3 Direct

The direct addressing mode is used to access data within the first 256 bytes of memory with a single two-byte instruction. In the direct addressing mode, the low byte of the operand's address is contained in the byte following the opcode. The high byte of the address is assumed to be \$00. Most direct instructions take two bytes, one for the opcode and one for the operand's address. BRSET and BRCLR are three-byte instructions that use direct addressing to access the operand and relative addressing to specify a branch destination. Table 3-3 lists the instructions that can be used in the direct addressing mode.

Table 3-3. Direct Addressing Instructions

| Instruction | Mnemonic |
|--------------------------------------|----------|
| Add with Carry | ADC |
| Add | ADD |
| Logical AND | AND |
| Arithmetic Shift Left | ASL |
| Arithmetic Shift Right | ASR |
| Clear Bit in Memory | BCLR |
| Bit Test Memory with Accumulator | BIT |
| Branch if Bit n Is Clear | BRCLR |
| Branch if Bit n Is Set | BRSET |
| Set Bit in Memory | BSET |
| Clear | CLR |
| Compare Accumulator with Memory | CMP |
| Complement | COM |
| Compare Index Register with Memory | CPX |
| Decrement | DEC |
| Exclusive OR Memory with Accumulator | EOR |
| Increment | INC |
| Jump | JMP |
| Jump to Subroutine | JSR |
| Load Accumulator from Memory | LDA |
| Load Index Register from Memory | LDX |
| Logical Shift Left | LSL |
| Logical Shift Right | LSR |
| Negate | NEG |
| Inclusive OR | ORA |
| Rotate Left through Carry Bit | ROL |
| Rotate Right through Carry Bit | ROR |
| Subtract with Carry | SBC |
| Store Accumulator in Memory | STA |
| Store Index Register in Memory | STX |
| Subtract | SUB |
| Test for Negative or Zero | TST |

NOTE: ASL = LSL

3.3.4 Extended

The extended addressing mode is used to access data in any memory location with a single three-byte instruction. In the extended addressing mode, the high and low bytes of the operand's address are contained in the two bytes following the opcode. Extended instructions take three bytes, one for the opcode and two for the operand's address.

When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction. Table 3-4 lists the instructions that can be used in the extended addressing mode.

Table 3-4. Extended Addressing Instructions

| Instruction | Mnemonic |
|--------------------------------------|-----------------|
| Add with Carry | ADC |
| Add | ADD |
| Logical AND | AND |
| Bit Test Memory with Accumulator | BIT |
| Compare Accumulator with Memory | CMP |
| Compare Index Register with Memory | CPX |
| Exclusive OR Memory with Accumulator | EOR |
| Jump | JMP |
| Jump to Subroutine | JSR |
| Load Accumulator from Memory | LDA |
| Load Index Register from Memory | LDX |
| Inclusive OR | ORA |
| Subtract with Carry | SBC |
| Store Accumulator in Memory | STA |
| Store Index Register in Memory | STX |
| Subtract | SUB |

Indexed, No Offset

The indexed, no offset addressing mode is used to access data with variable addresses within the first 256 memory locations. The CPU finds the low byte of the operand's conditional address by reading the contents of the index register. The high byte is assumed to be \$00. These instructions are only one byte long. The indexed, no offset mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location. Table 3-5 lists the instructions that can be used in the indexed, no offset addressing mode.

3.3.6 Indexed, 8-Bit Offset

The indexed, 8-bit offset addressing mode is used to access data with variable addresses within the first 511 memory locations. The CPU finds the operand's conditional address by adding the unsigned contents of the index register to the unsigned byte following the opcode. This addressing mode is useful for selecting the kth element in an n-element table. The table may begin anywhere within the first 256 memory locations and could extend as far as location 510 (\$01FE). With this two-byte instruction, k typically would be in the index register, and the address of the beginning of the table would be in the byte following the opcode. Table 3-5 lists the instructions that can be used in the indexed, 8-bit offset addressing mode.

3.3.7 Indexed, 16-Bit Offset

The indexed, 16-bit offset addressing mode is used to access data with variable addresses at any location in memory. The CPU finds the operand's conditional address by adding the unsigned contents of the 8-bit index register to the 16-bit unsigned word formed by the two bytes following the opcode. The first byte after the opcode is the high byte of the 16-bit offset; the second byte is the low byte. This addressing mode can be used in a manner similar to indexed, 8-bit offset, but this three-byte instruction allows tables to be anywhere in memory.

As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing. Table 3-5 lists the instructions that can be used in the indexed, 16-bit offset addressing mode.

Freescale Semiconductor, Inc.
Table 3-5. Indexed Addressing Instructions

| Instruction | Mnemonic | No Offset | 8-Bit Offset | 16-Bit Offset |
|--------------------------------------|----------|-----------|--------------|---------------|
| Add with Carry | ADC | √ | √ | √ |
| Add | ADD | √ | √ | √ |
| Logical AND | AND | √ | √ | √ |
| Arithmetic Shift Left | ASL | √ | √ | |
| Arithmetic Shift Right | ASR | √ | √ | |
| Bit Test Memory with Accumulator | BIT | √ | √ | √ |
| Clear | CLR | √ | √ | |
| Compare Accumulator with Memory | CMP | √ | √ | √ |
| Complement | COM | √ | √ | |
| Compare Index Register with Memory | CPX | √ | √ | √ |
| Decrement | DEC | √ | √ | |
| Exclusive OR Memory with Accumulator | EOR | √ | √ | √ |
| Increment | INC | √ | √ | |
| Jump | JMP | √ | √ | √ |
| Jump to Subroutine | JSR | √ | √ | √ |
| Load Accumulator from Memory | LDA | √ | √ | √ |
| Load Index Register from Memory | LDX | √ | √ | √ |
| Logical Shift Left | LSL | √ | √ | |
| Logical Shift Right | LSR | √ | √ | |
| Negate | NEG | √ | √ | |
| Inclusive OR | ORA | √ | √ | √ |
| Rotate Left through Carry Bit | ROL | √ | √ | |
| Rotate Right through Carry Bit | ROR | √ | √ | |
| Subtract with Carry | SBC | √ | √ | √ |
| Store Accumulator in Memory | STA | √ | √ | √ |
| Store Index Register in Memory | STX | √ | √ | √ |
| Subtract | SUB | √ | √ | √ |
| Test for Negative or Zero | TST | √ | √ | |

3.3.8 Relative

The relative addressing mode is used only for branch instructions. The CPU finds the conditional branch destination by adding the signed byte following the opcode to the contents of the program counter if the branch condition is true. If the branch condition is not true, the program counter goes to the next instruction. To branch either forward or backward, the offset is a signed, two's complement byte that gives a branching range of -127 to +128 bytes from the address of the next location after the branch instruction.

The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch. Table 3-6 lists the instructions that can use the relative addressing mode.

Table 3-6. Relative Addressing 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 Interrupt Line is High | BIH |
| Branch if Interrupt Line Is Low | BIL |
| Branch if Lower | BLO |
| Branch if Lower or Same | BLS |
| Branch if Interrupt Mask is Clear | BMC |
| Branch if Minus | BMI |
| Branch if Interrupt Mask is Set | BMS |
| Branch if Not Equal | BNE |
| Branch if Plus | BPL |
| Branch Always | BRA |
| Branch if Bit n Is Clear | BRCLR |
| Branch if Bit n Is Set | BRSET |
| Branch Never | BRN |
| Branch to Subroutine | BSR |

Freescale Semiconductor, Inc.

3.4 Instruction Set

This MCU uses all the instructions available in the M146805 CMOS Family plus the unsigned multiply (MUL) instruction. The MUL instruction allows unsigned multiplication of the contents of the accumulator and the index register. The high-order product is then stored in the index register, and the low-order product is stored in the accumulator.

The MCU instructions can be divided into five basic types:

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

3.4.1 Register/Memory Instructions

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. Most register/memory instructions can be used in the following addressing modes:

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

Table 3-7 lists the register/memory instructions.

Table 3-7. Register/Memory Instructions

| Instruction | Mnemonic |
|--|----------|
| Load Accumulator from Memory | LDA |
| Load Index Register from Memory | LDX |
| Store Accumulator in Memory | STA |
| Store Index Register in Memory | STX |
| Add Memory to Accumulator | ADD |
| Add Memory and Carry Bit to Accumulator | ADC |
| Subtract Memory | SUB |
| Subtract Memory from Accumulator with Borrow | SBC |
| AND Memory with Accumulator | AND |
| OR Memory with Accumulator | ORA |
| Exclusive OR Memory with Accumulator | EOR |
| Arithmetic Compare Accumulator with Memory | CMP |
| Arithmetic Compare Index Register with Memory | CPX |
| Bit Test Memory with Accumulator (Logical Compare) | BIT |
| Multiply | MUL |

Read-Modify-Write Instructions

These instructions read a memory location or a register, modify its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not write a replacement value. Read-modify-write instructions can be used in the following addressing modes:

- Inherent
- Direct
- Indexed, no offset
- Indexed, 8-bit offset

Table 3-8 lists the read-modify-write instructions.

Table 3-8. Read-Modify-Write Instructions

| Instruction | Mnemonic |
|--------------------------------|----------|
| Increment | INC |
| Decrement | DEC |
| Clear | CLR |
| Complement | COM |
| Negate (Twos Complement) | NEG |
| Rotate Left through Carry Bit | ROL |
| Rotate Right through Carry Bit | ROR |
| Logical Shift Left | LSL |
| Logical Shift Right | LSR |
| Arithmetic Shift Right | ASR |
| Test for Negative or Zero | TST |

Freescale Semiconductor, Inc.

3.4.3 Jump/Branch Instructions

Jump instructions allow the CPU to interrupt the normal sequence of the program counter. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Jump instructions can be used in the following addressing modes:

- Direct
- Extended
- Indexed, no offset
- Indexed, 8-bit offset
- Indexed, 16-bit offset

Branch instructions allow the CPU to interrupt the normal sequence of the program counter when a test condition is met. If the test condition is not met, the branch is not performed. All branch instructions are used in the relative addressing mode.

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

Table 3-9 lists the jump and branch instructions.

Table 3-9. Jump and Branch Instructions

| Instruction | Mnemonic |
|--------------------------------|----------|
| Branch Always | BRA |
| Branch Never | BRN |
| Branch if Bit n of M = 0 | BRCLR |
| Branch if Bit n of M = 1 | BRSET |
| Branch if Higher | BHI |
| Branch if Lower or Same | BLS |
| Branch if Carry Bit Clear | BCC |
| Branch if Higher or Same | BHS |
| Branch if Carry Bit Set | BCS |
| Branch if Lower | BLO |
| Branch if Not Equal | BNE |
| Branch if Equal | BEQ |
| Branch if Half-Carry Bit Clear | BHCC |
| Branch if Half-Carry Bit Set | BHCS |
| Branch if Plus | BPL |
| Branch if Minus | BMI |
| Branch if Interrupt Mask Clear | BMC |
| Branch if Interrupt Mask Set | BMS |
| Branch if Interrupt Line Low | BIL |
| Branch if Interrupt Line High | BIH |
| Branch to Subroutine | BSR |
| Jump Unconditional | JMP |
| Jump to Subroutine | JSR |

3.4.4 Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory. Port data registers, port data direction registers, control/status registers for on-chip subsystems, and on-chip RAM locations are in the first 256 bytes of memory. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations. Bit manipulation instructions are used in the direct addressing mode. Table 3-10 lists the bit manipulation instructions.

Table 3-10. Bit Manipulation Instructions

| Instruction | Mnemonic |
|--------------------------|------------------------|
| Set Bit n | BSET n (n = 0 . . . 7) |
| Clear Bit n | BCLR n (n = 0 . . . 7) |
| Branch if Bit n of M = 0 | BRCLR |
| Branch if Bit n of M = 1 | BRSET |

3.4.5 Control Instructions

Control instructions are register reference instructions that control CPU operation during program execution. Control instructions are used in the inherent addressing mode. Table 3-11 lists the control instructions.

Table 3-11. Control Instructions

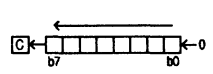
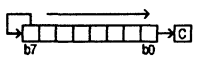
| Instruction | Mnemonic |
|--|----------|
| Transfer Accumulator to Index Register | TAX |
| Transfer Index Register to Accumulator | TXA |
| Set Carry Bit | SEC |
| Clear Carry Bit | CLC |
| Set Interrupt Mask | SEI |
| Clear Interrupt Mask | CLI |
| Software Interrupt | SWI |
| Return from Subroutine | RTS |
| Return from Interrupt | RTI |
| Reset Stack Pointer | RSP |
| No Operation | NOP |
| Stop | STOP |
| Wait | WAIT |

3.4.0 Instruction Set Summary

Table 3-12 shows all MC68HC05P4 instructions in all possible addressing modes. For each instruction, the operand construction is shown as well as the execution time in internal clock cycles (t_{cyc}). One internal clock cycle equals two oscillator input cycles. The following legend summarizes the symbols and abbreviations used in Table 3-12.

Abbreviations and Symbols

| | | | | |
|-------|---|--|-------|------------------------------|
| A | – | Accumulator | | |
| C | – | Carry/borrow bit in condition code register | | |
| CCR | – | Condition code register | | |
| dd | – | Address of operand in direct addressing mode (1 byte) | | |
| dd rr | – | Address (dd) of operand and offset (rr) of branch instruction for bit test instructions | | |
| DIR | – | Direct addressing mode | | |
| ee ff | – | High (ee) and low (ff) bytes of offset in indexed, 16-bit offset addressing mode (2 bytes) | | |
| EXT | – | Extended addressing mode | | |
| ff | – | Offset byte in indexed, 8-bit offset addressing mode (1 byte) | | |
| H | – | Half carry flag in condition code register | | |
| hh ll | – | High (hh) and low (ll) bytes of operand address in extended addressing mode (2 bytes) | | |
| I | – | Interrupt mask in condition code register | | |
| ii | – | Operand byte for immediate addressing mode | | |
| IMM | – | Immediate addressing mode | | |
| INH | – | Inherent addressing mode | | |
| IX | – | Indexed, no offset addressing mode | | |
| IX1 | – | Indexed, 8-bit offset addressing mode | | |
| IX2 | – | Indexed, 16-bit offset addressing mode | | |
| M | – | Any memory location (1 byte) | | |
| N | – | Negative flag in condition code register | | |
| n | – | Any bit (7,6,5 . . . 0) | | |
| opr | – | Operand byte | | |
| PC | – | Program counter | | |
| PCH | – | Program counter high byte | | |
| PCL | – | Program counter low byte | | |
| REL | – | Relative addressing mode | | |
| rel | – | Offset byte for relative addressing mode | | |
| rr | – | Offset byte of branch instruction | | |
| SP | – | Stack pointer | | |
| X | – | Index register | | |
| Z | – | Zero flag in condition code register | | |
| ↕ | – | Set if true; clear if not true | – () | – Negation (twos complement) |
| – | – | Not affected | + | – Inclusive OR |
| ? | – | If | ⊕ | – Exclusive OR |
| 0 | – | Cleared | — | – NOT |
| 1 | – | Set | × | – Multiplication |
| () | – | Contents of | + | – Addition |
| ← | – | Is loaded with | – | – Subtraction |
| • | – | AND | : | – Concatenated with |

| Source Form(s) | Operation | Description | Addressing Mode for Operand | Machine Coding (hexadecimal) | | Cycles | Condition Code | | | | |
|---|--|---|--|--|--|--------------------------------------|----------------|---|---|---|---|
| | | | | Opcode | Operand | | H | I | N | Z | C |
| ADC opr | Add with carry | $A \leftarrow (A) + (M) + C$ | IMM DIR EXT IX2 IX1 IX | A9 B9 C9 D9 E9 F9 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 | ↓ | - | ↓ | ↓ | ↓ |
| ADD opr | Add without carry | $A \leftarrow (A) + (M)$ | IMM DIR EXT IX2 IX1 IX | AB BB CB DB EB FB | ii dd hh ll ee ff ff | 2 3 4 5 4 3 | ↓ | - | ↓ | ↓ | ↓ |
| AND opr | Logical AND | $A \leftarrow (A) \bullet (M)$ | IMM DIR EXT IX2 IX1 IX | A4 B4 C4 D4 E4 F4 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 | - | - | ↓ | ↓ | - |
| ASL opr ASLA ASLX ASL opr ASL opr | Arithmetic shift left |  | DIR INH INH IX1 IX | 38 48 58 68 78 | dd | 5 3 3 6 5 | - | - | ↓ | ↓ | ↓ |
| ASR opr ASRA ASRX ASR opr ASR opr | Arithmetic shift right |  | DIR INH INH IX1 IX | 37 47 57 67 77 | dd | 5 3 3 6 5 | - | - | ↓ | ↓ | ↓ |
| BCC rel | Branch if carry bit clear | ? 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 | ? C = 1 | REL | 25 | rr | 3 | - | - | - | - | - |
| BEQ rel | Branch if equal | ? Z = 1 | REL | 27 | rr | 3 | - | - | - | - | - |
| BHCC rel | Branch if half carry bit clear | ? H = 0 | REL | 28 | rr | 3 | - | - | - | - | - |
| BHCS rel | Branch if half carry bit set | ? H = 1 | REL | 29 | rr | 3 | - | - | - | - | - |
| BHI rel | Branch if higher | ? C + Z = 0 | REL | 22 | rr | 3 | - | - | - | - | - |
| BHS rel | Branch if higher or same | ? C = 0 | REL | 24 | rr | 3 | - | - | - | - | - |
| BIH rel | Branch if IRQ pin high | ? IRQ = 1 | REL | 2F | rr | 3 | - | - | - | - | - |
| BIL rel | Branch if IRQ pin low | ? IRQ = 0 | REL | 2E | rr | 3 | - | - | - | - | - |
| BIT rel | Bit test accumulator contents with memory contents | $(A) \bullet (M)$ | IMM DIR EXT IX2 IX1 IX | A5 B5 C5 D5 E5 F5 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 | - | - | ↓ | ↓ | - |
| BLO rel | Branch if lower | ? C = 1 | REL | 25 | rr | 3 | - | - | - | - | - |
| BLS rel | Branch if lower or same | ? C + Z = 1 | REL | 23 | rr | 3 | - | - | - | - | - |
| BMC rel | Branch if interrupt mask clear | ? I = 0 | REL | 2C | rr | 3 | - | - | - | - | - |
| BMI rel | Branch if minus | ? N = 1 | REL | 2B | rr | 3 | - | - | - | - | - |
| BMS rel | Branch if interrupt mask set | ? I = 0 | REL | 2D | rr | 3 | - | - | - | - | - |
| BNE rel | Branch if not equal | ? Z = 0 | REL | 26 | rr | 3 | - | - | - | - | - |
| BPL rel | Branch if plus | ? N = 0 | REL | 2A | rr | 3 | - | - | - | - | - |

| Source Form(s) | Operation | Description | Addressing Mode for Operand | Machine Coding (hexadecimal) | | Cycles | Condition Code | | | | |
|--|--|--|-----------------------------|------------------------------|---------|--------|----------------|---|---|---|---|
| | | | | Opcode | Operand | | H | I | N | Z | C |
| BRA rel | Branch always | ? 1 = 1 | REL | 20 | rr | 3 | - | - | - | - | - |
| BRCLR n opr rel | Branch if bit n clear | ? Mn = 0 | 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 rel | Branch never | ? 1 = 0 | REL | 21 | rr | 3 | - | - | - | - | |
| BRSET n opr rel | Branch if bit n set | ? Mn = 1 | 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 n opr | Set bit n | Mn ← 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 | - | - | - | - | - |
| BSR rel | Branch to subroutine | PC ← (PC) + 2; push (PCL) SP ← (SP) - 1; push (PCH) SP ← (SP) - 1 PC ← (PC) + rel | REL | AD | rr | 6 | - | - | - | - | |
| CLC | Clear carry bit | C ← 0 | INH | 98 | | 2 | - | - | - | 0 | |
| CLI | Clear interrupt mask | I ← 0 | INH | 9A | | 2 | - | 0 | - | - | |
| CLR opr CLRA CLR X CLR opr CLR opr | Clear register | M ← \$00 A ← \$00 X ← \$00 M ← \$00 M ← \$00 | DIR | 3F | dd | 5 | - | - | 0 | 1 | - |
| | | | INH | 4F | | 3 | - | - | - | - | - |
| | | | INH | 5F | | 3 | - | - | - | - | - |
| | | | IX1 | 6F | ff | 6 | - | - | - | - | - |
| | | | IX | 7F | | 5 | - | - | - | - | - |
| CMP opr | Compare accumulator contents with memory contents | (A) - (M) | IMM | A1 | ii | 2 | - | - | ‡ | ‡ | ‡ |
| | | | DIR | B1 | dd | 3 | - | - | - | - | - |
| | | | EXT | C1 | hh ll | 4 | - | - | - | - | - |
| | | | IX2 | D1 | ee ff | 5 | - | - | - | - | - |
| | | | IX1 | E1 | ff | 4 | - | - | - | - | - |
| | | | IX | F1 | | 3 | - | - | - | - | - |
| COM opr COMA COM X COM opr COM opr | Complement register contents (ones complement) | M ← M = \$FF - (M) A ← A = \$FF - (A) X ← X = \$FF - (X) M ← M = \$FF - (M) M ← M = \$FF - (M) | DIR | 33 | dd | 5 | - | - | ‡ | ‡ | 1 |
| | | | INH | 43 | | 3 | - | - | - | - | - |
| | | | INH | 53 | | 3 | - | - | - | - | - |
| | | | IX1 | 63 | ff | 6 | - | - | - | - | - |
| | | | IX | 73 | | 5 | - | - | - | - | - |
| CPX opr | Compare index register contents with memory contents | (X) - (M) | IMM | A3 | ii | 2 | - | - | ‡ | ‡ | ‡ |
| | | | DIR | B3 | dd | 3 | - | - | - | - | - |
| | | | EXT | C3 | hh ll | 4 | - | - | - | - | - |
| | | | IX2 | D3 | ee ff | 5 | - | - | - | - | - |
| | | | IX1 | E3 | ff | 4 | - | - | - | - | - |
| | | | IX | F3 | | 3 | - | - | - | - | - |
| DEC opr DECA DEC X DEC opr DEC opr | Decrement register contents | M ← (M) - 1 A ← (A) - 1 X ← (X) - 1 M ← (M) - 1 M ← (M) - 1 | DIR | 3A | dd | 5 | - | - | ‡ | ‡ | - |
| | | | INH | 4A | | 3 | - | - | - | - | - |
| | | | INH | 5A | | 3 | - | - | - | - | - |
| | | | IX1 | 6A | ff | 6 | - | - | - | - | - |
| | | | IX | 7A | | 5 | - | - | - | - | - |

| Source Form(s) | Operation | Description | Addressing Mode for Operand | Machine Coding (hexadecimal) | | Cycles | Condition Code | | | | |
|---|--|---|-----------------------------|------------------------------|---------|--------|----------------|---|---|---|---|
| | | | | Opcode | Operand | | H | I | N | Z | C |
| EOR opr | Exclusive OR accumulator contents with memory contents | $A \leftarrow (A) \oplus (M)$ | IMM | A8 | ii | 2 | - | - | ↓ | ↓ | - |
| | | | DIR | B8 | dd | 3 | - | - | ↓ | ↓ | - |
| | | | EXT | C8 | hh ll | 4 | - | - | ↓ | ↓ | - |
| | | | IX2 | D8 | ee ff | 5 | - | - | ↓ | ↓ | - |
| | | | IX1 | E8 | ff | 4 | - | - | ↓ | ↓ | - |
| IX | F8 | | 3 | - | - | ↓ | ↓ | - | - | | |
| INC opr INCA INCX INC opr INC opr | Increment memory or register contents | $M \leftarrow (M) + 1$ $A \leftarrow (A) + 1$ $X \leftarrow (X) + 1$ $M \leftarrow (M) + 1$ $M \leftarrow (M) + 1$ | DIR | 3C | dd | 5 | - | - | ↓ | ↓ | - |
| | | | INH | 4C | | 3 | - | - | ↓ | ↓ | - |
| | | | INH | 5C | | 3 | - | - | ↓ | ↓ | - |
| | | | IX1 | 6C | ff | 6 | - | - | ↓ | ↓ | - |
| | | | IX | 7C | | 5 | - | - | ↓ | ↓ | - |
| JMP opr | Unconditional jump | $PC \leftarrow \text{jump address}$ | DIR | BC | dd | 2 | - | - | - | - | - |
| | | | EXT | CC | hh ll | 3 | - | - | - | - | - |
| | | | IX2 | DC | ee ff | 4 | - | - | - | - | - |
| | | | IX1 | EC | ff | 3 | - | - | - | - | - |
| | | | IX | FC | | 2 | - | - | - | - | - |
| JSR opr | Jump to subroutine | $PC \leftarrow (PC) + n$ ($n = 1, 2, \text{ or } 3$) Push (PCL); $SP \leftarrow (SP) - 1$ Push (PCH); $SP \leftarrow (SP) - 1$ $PC \leftarrow \text{conditional address}$ | DIR | BD | dd | 5 | - | - | - | - | - |
| | | | EXT | CD | hh ll | 6 | - | - | - | - | - |
| | | | IX2 | DD | ee ff | 7 | - | - | - | - | - |
| | | | IX1 | ED | ff | 6 | - | - | - | - | - |
| | | | IX | FD | | 5 | - | - | - | - | - |
| LDA opr | Load accumulator with memory contents | $A \leftarrow (M)$ | 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 opr | Load index register with memory contents | $X \leftarrow (M)$ | 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 opr LSLA LSLX LSL opr LSL opr | Logical shift left | | DIR | 38 | dd | 5 | - | - | ↓ | ↓ | ↓ |
| | | | INH | 48 | | 3 | - | - | ↓ | ↓ | ↓ |
| | | | INH | 58 | | 3 | - | - | ↓ | ↓ | ↓ |
| | | | IX1 | 68 | ff | 6 | - | - | ↓ | ↓ | ↓ |
| | | | IX | 78 | | 5 | - | - | ↓ | ↓ | ↓ |
| LSR opr LSRA LSRX LSR opr LSR opr | Logical shift right | | DIR | 34 | dd | 5 | - | - | 0 | ↓ | ↓ |
| | | | INH | 44 | | 3 | - | - | 0 | ↓ | ↓ |
| | | | INH | 54 | | 3 | - | - | 0 | ↓ | ↓ |
| | | | IX1 | 64 | ff | 6 | - | - | 0 | ↓ | ↓ |
| | | | IX | 74 | | 5 | - | - | 0 | ↓ | ↓ |
| MUL | Unsigned multiply | $X : A \leftarrow (X) \times (A)$ | INH | 42 | | 11 | 0 | - | - | - | 0 |
| NEG opr NEGA NEGX NEG opr NEG opr | Negate memory or register contents (two's complement) | $M \leftarrow \neg(M) = \$00 - (M)$ $A \leftarrow \neg(A) = \$00 - (A)$ $X \leftarrow \neg(X) = \$00 - (X)$ $M \leftarrow \neg(M) = \$00 - (M)$ $M \leftarrow \neg(M) = \$00 - (M)$ | DIR | 30 | dd | 5 | - | - | ↓ | ↓ | ↓ |
| | | | INH | 40 | | 3 | - | - | ↓ | ↓ | ↓ |
| | | | INH | 50 | | 3 | - | - | ↓ | ↓ | ↓ |
| | | | IX1 | 60 | ff | 6 | - | - | ↓ | ↓ | ↓ |
| | | | IX | 70 | | 5 | - | - | ↓ | ↓ | ↓ |
| NOB | No operation | | INH | 9D | | 2 | - | - | - | - | - |
| ORA opr | Inclusive OR accumulator contents with memory contents | $A \leftarrow (A) \vee (M)$ | 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 | - | - | ↓ | ↓ | - | | | |
| ROL opr ROLA ROLX ROL opr ROL opr | Rotate left through carry | | DIR | 39 | dd | 5 | - | - | ↓ | ↓ | ↓ |
| | | | INH | 49 | | 3 | - | - | ↓ | ↓ | ↓ |
| | | | INH | 59 | | 3 | - | - | ↓ | ↓ | ↓ |
| | | | IX1 | 69 | ff | 6 | - | - | ↓ | ↓ | ↓ |
| | | | IX | 79 | | 5 | - | - | ↓ | ↓ | ↓ |

Table 3-11. Freescale Semiconductor, Inc.

| Source Form(s) | Operation | Description | Addressing Mode for Operand | Machine Coding (hexadecimal) | | Cycles | Condition Code | | | | |
|---|---|--|---------------------------------------|----------------------------------|----------------------------------|----------------------------|----------------|---|---|---|---|
| | | | | Opcode | Operand | | H | I | N | Z | C |
| ROR opr RORA RORX ROR opr ROR opr | Rotate right through carry | | DIR INH INH IX1 IX | 36 46 56 66 76 | dd ff | 5 3 3 6 5 | - | - | ↕ | ↕ | ↕ |
| RSP | Reset stack pointer | $SP \leftarrow \$00FF$ | INH | 9C | | 2 | From Stack | | | | |
| RTI | Return from interrupt | $SP \leftarrow (SP) + 1$; pull (CCR) $SP \leftarrow (SP) + 1$; pull (A) $SP \leftarrow (SP) + 1$; pull (X) $SP \leftarrow (SP) + 1$; pull (PCH) $SP \leftarrow (SP) + 1$; pull (PCL) | INH | 80 | | 9 | ↕ | ↕ | ↕ | ↕ | ↕ |
| RTS | Return from subroutine | $SP \leftarrow (SP) + 1$; pull (PCH) $SP \leftarrow (SP) + 1$; pull (PCL) | INH | 81 | | 6 | - | - | - | - | - |
| SBC opr | Subtract memory contents and carry bit from accumulator contents | $A \leftarrow (A) - (M) - C$ | 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 | $C \leftarrow 1$ | INH | 99 | | 2 | - | - | - | - | 1 |
| SEI | Set interrupt mask | $I \leftarrow 1$ | INH | 9B | | 2 | - | 1 | - | - | - |
| STA opr | Store accumulator contents in memory | $M \leftarrow (A)$ | DIR EXT IX2 IX1 IX | B7 C7 D7 E7 F7 | dd hh ll ee ff ff | 4 5 6 5 4 | - | - | ↕ | ↕ | - |
| STOP | Enable \overline{IRQ} ; stop oscillator | | INH | 8E | | 2 | - | 0 | - | - | - |
| STX opr | Store index register contents in memory | $M \leftarrow (X)$ | DIR EXT IX2 IX1 IX | BF CF DF EF FF | dd hh ll ee ff ff | 4 5 6 5 4 | - | - | ↕ | ↕ | - |
| SUB opr | Subtract memory contents from accumulator contents | $A \leftarrow (A) - (M)$ | IMM DIR EXT IX2 IX1 IX | A0 B0 C0 D0 E0 F0 | ii dd hh ll ee ff ff | 2 3 4 5 4 3 | - | - | ↕ | ↕ | ↕ |
| SWI | Software interrupt | $PC \leftarrow (PC) + 1$; push (PCL) $SP \leftarrow (SP) - 1$; push (PCH) $SP \leftarrow (SP) - 1$; push (X) $SP \leftarrow (SP) - 1$; push (A) $SP \leftarrow (SP) - 1$; push (CCR) $SP \leftarrow (SP) - 1$; $I \leftarrow 1$ $PCH \leftarrow$ Interrupt vector hi byte $PCL \leftarrow$ Int. vector low byte | INH | 83 | | 10 | - | 1 | - | - | - |
| TAX | Transfer accumulator contents to index register | $X \leftarrow (A)$ | INH | 97 | | 2 | - | - | - | - | - |
| TST opr TSTA TSTX TST opr TST opr | Test memory, accumulator, or index register contents for negative or zero | $(M) - \$00$ | DIR INH INH IX1 IX | 3D 4D 5D 6D 7D | dd ff | 4 3 3 5 4 | - | - | ↕ | ↕ | 0 |
| TXA | Transfer index register contents to accumulator | $A \leftarrow (X)$ | INH | 9F | | 2 | - | - | - | - | - |
| WAIT | Enable interrupts; halt CPU | | INH | 8F | | 2 | - | 0 | - | - | - |

Freescale Semiconductor, Inc.

3.4.1 Opcode Map

Table 3-13 is an opcode map for the MC68HC05P4 instructions.

| HI | Bit Manipulation | | | Branch | | | Read-Modify-Write | | | Control | | | Register/Memory | | | LO | |
|------|------------------|-------|------|--------|------|------|-------------------|------|------|---------|-----|-----|-----------------|-----|-----|-----|-----|
| | DIR | DIR | DIR | REL | DIR | DIR | INH | INH | INH | IX | INH | IMM | DIR | EXT | IX2 | | IX1 |
| 0 | 0000 | 0001 | 0010 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 0 | 1 | A | B | C | D | E | F |
| 0000 | BRSET0 | BSET0 | BRA | REL | NEG | NEGA | | NEG | NEG | RTI | | SUB | SUB | SUB | SUB | SUB | SUB |
| 0001 | BRCLR0 | BCLR0 | BRN | REL | | | | | | RTS | | CMP | CMP | CMP | CMP | CMP | CMP |
| 0010 | BRSET1 | BSET1 | BHI | REL | | MUL | | | | | | SBC | SBC | SBC | SBC | SBC | SBC |
| 0011 | BRCLR1 | BCLR1 | BLS | REL | COM | COMA | | COM | COM | SWI | | CPX | CPX | CPX | CPX | CPX | CPX |
| 0100 | BRSET2 | BSET2 | BCC | REL | LSR | LSRA | | LSR | LSR | | | AND | AND | AND | AND | AND | AND |
| 0101 | BRCLR2 | BCLR2 | BCS | REL | | | | | | | | BIT | BIT | BIT | BIT | BIT | BIT |
| 0110 | BRSET3 | BSET3 | BNE | REL | ROR | RORA | | ROR | ROR | | | LDA | LDA | LDA | LDA | LDA | LDA |
| 0111 | BRCLR3 | BCLR3 | BEQ | REL | ASR | ASRA | | ASR | ASR | | TAX | | STA | STA | STA | STA | STA |
| 1000 | BRSET4 | BSET4 | BHCC | REL | LSL | LSLA | | LSL | LSL | | | EOR | EOR | EOR | EOR | EOR | EOR |
| 1001 | BRCLR4 | BCLR4 | BHCS | REL | ROL | ROLA | | ROL | ROL | | | ADC | ADC | ADC | ADC | ADC | ADC |
| A | BRSET5 | BSET5 | BPL | REL | DEC | DECA | | DEC | DEC | | CLI | | ORA | ORA | ORA | ORA | ORA |
| 1010 | BRCLR5 | BCLR5 | BMI | REL | | | | | | | | ADD | ADD | ADD | ADD | ADD | ADD |
| 1011 | BRSET6 | BSET6 | BMC | REL | INC | INCA | | INC | INC | | RSP | | JMP | JMP | JMP | JMP | JMP |
| C | BRCLR6 | BCLR6 | BMS | REL | TST | TSTA | | TST | TST | | NOP | | BSR | BSR | BSR | BSR | BSR |
| D | BRSET7 | BSET7 | BIL | REL | | | | | | | | LDX | LDX | LDX | LDX | LDX | LDX |
| 1100 | BRCLR7 | BCLR7 | BIH | REL | CLR | CLRA | | CLR | CLR | STOP | | STX | STX | STX | STX | STX | STX |
| 1101 | BRSET8 | BSET8 | BIM | REL | | | | | | WAIT | | TXA | TXA | TXA | TXA | TXA | TXA |
| 1110 | BRCLR8 | BCLR8 | BIM | REL | | | | | | | | | | | | | |
| 1111 | BRSET9 | BSET9 | BIM | REL | | | | | | | | | | | | | |

LEGEND

| | |
|------|------------------------------------|
| F | High Byte of Opcode in Hexadecimal |
| 1111 | High Byte of Opcode in Binary |
| SUB | Low Byte of Opcode in Hexadecimal |
| 0 | Low Byte of Opcode in Binary |

ABBREVIATIONS FOR ADDRESSING MODES

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

3.5 Low-Power Modes

The following paragraphs describe the STOP and WAIT modes. (See also 5.2 Data-Retention Mode.)

3.5.1 STOP Mode

The STOP instruction places the MCU in its lowest power-consumption mode. In STOP mode, the internal oscillator turns off, halting all internal processing including capture/compare timer operation and computer operating properly (COP) timeout operation. (See Figure 3-8.)

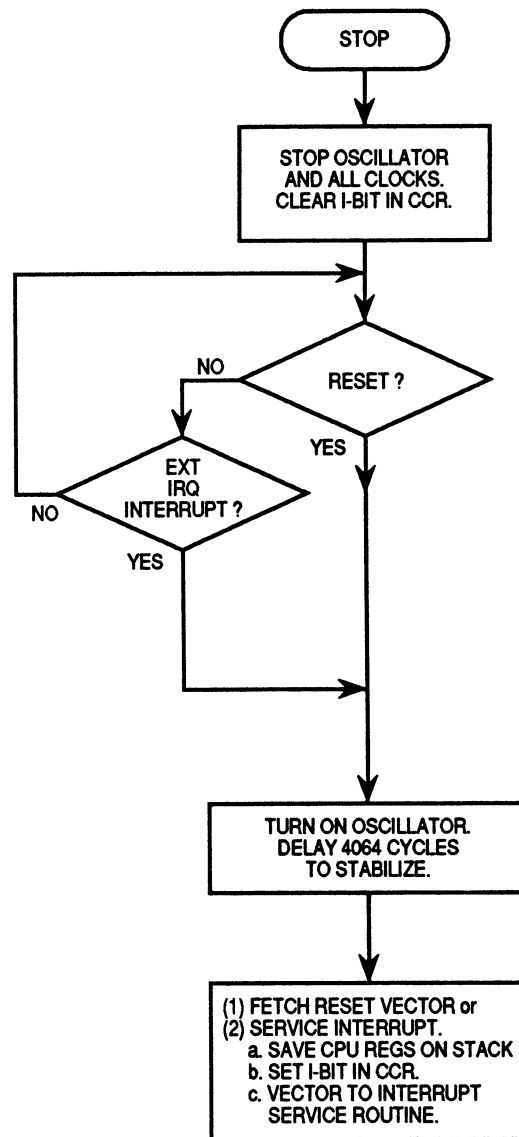


Figure 3-8. STOP Function Flowchart

During STOP mode, the ICIE, OCIE, and TOIE bits in the capture/compare timer control register are cleared to disable timer interrupts. The capture/compare timer prescaler is cleared. The interrupt mask (I-bit) in the condition code register is cleared to enable external interrupts. All other registers and memory locations remain unchanged. All I/O lines remain unchanged. The MCU can be brought out of STOP mode only by an external interrupt or a reset. An external interrupt automatically loads the program counter with the contents of \$1FFA and \$1FFB, the location of the vector address of the interrupt service routine. A reset automatically loads the program counter with the contents of \$1FFE and \$1FFF, the location of the vector address of the reset service routine.

3.5.2 WAIT Mode

The WAIT instruction places the MCU in an intermediate power-consumption mode. All CPU action stops, but the capture/compare timer remains active. An interrupt from the capture/compare timer can cause the MCU to exit WAIT mode. (See Figure 3-9.)

The COP watchdog is not disabled in WAIT mode. The user should exit from WAIT and reset the COP timer before timeout to prevent a watchdog reset.

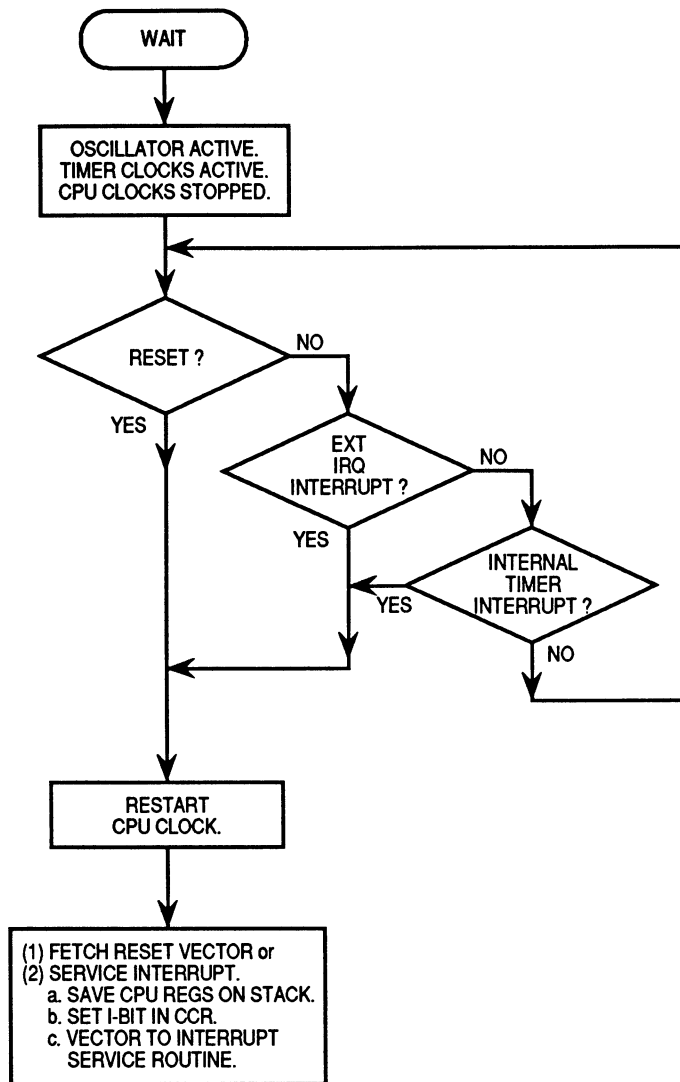


Figure 3-9. WAIT Function Flowchart

During WAIT mode, the interrupt mask (I-bit) in the condition code register is cleared to enable interrupts. All other registers, memory locations, and I/O lines remain in their previous states.

SECTION 4

RESETS AND INTERRUPTS

This section describes the three ways that the CPU can be reset and the three kinds of interrupts.

4.1 Resets

A reset immediately stops execution of the current instruction. A reset forces the program counter to a known starting address and forces certain control and status bits to known conditions. The CPU can be reset three ways:

- Initial power-up (power-on reset)
- An external, logical zero signal on the reset pin ($\overline{\text{RESET}}$)
- Timeout of the COP watchdog timer

NOTE

The current instruction is the one already fetched and being operated on.

The following internal actions occur as a result of any CPU reset:

- All implemented data direction register bits are cleared to zero, so that the corresponding I/O pins become high-impedance inputs.
- The stack pointer is loaded with \$FF.
- The interrupt mask is set, inhibiting interrupts.
- The capture/compare timer clock divider stages are reset. The capture/compare timer is loaded with \$FFFC. The output compare bit (TCMP) and the output level bit (OLVL) are cleared. All capture/compare timer interrupt enable bits (ICIE, OCIE, and TOIE) are cleared to disable timer interrupts.
- The STOP latch is cleared to enable MCU clocks.
- The WAIT latch is cleared to wake the CPU from the WAIT mode.
- The program counter is loaded with the user-defined reset vector address; the high byte of the program counter is loaded with the contents

of location \$1FFE, and the low byte of the program counter is loaded from location \$1FFF.

4.1.1 Power-On Reset (POR)

An internal reset is generated on power-up when a positive transition occurs on V_{DD} . The power-on reset is strictly for power turn-on conditions and cannot be used to detect a drop in the power supply voltage.

To allow the clock generator to stabilize, there is a $4064 t_{cyc}$ (internal clock cycle) delay after the oscillator becomes active. If the \overline{RESET} pin is at logical zero at the end of $4064 t_{cyc}$, the CPU remains in the reset condition until \overline{RESET} goes to logical one.

4.1.2 External \overline{RESET} Input

The CPU is reset when a logical zero is applied to the \overline{RESET} input for a period of one and one-half internal clock cycles (t_{cyc}). The \overline{RESET} input consists of a Schmitt trigger that senses the logic level at the \overline{RESET} pin.

\overline{RESET} is an input-only pin and does not become active (go to logical zero) when an internal reset (power-on reset or computer operating properly watchdog reset) is generated.

4.1.3 Computer Operating Properly (COP) Reset

The MCU contains a watchdog timer as a factory-set mask option that automatically times out if not cleared within a specific time by a program sequence. The COP watchdog timer system is used to detect software errors. If the COP watchdog timer is allowed to time out, an internal reset is generated. The COP is implemented with an 18-stage ripple counter that provides a timeout period of 65.5 ms at an internal clock rate of 2 MHz. When the COP times out, an internal reset occurs, and the MCU is reinitialized in the same fashion as a power-on reset or external reset. A COP reset is prevented by writing a zero to bit 0 (COPR) of the COP control register at location \$1FF0. Writing a zero to COPR resets the counter and begins the timeout period again.

The COP register is a write-only register that is used to prevent a COP watchdog timeout. Reading this location returns the contents of a ROM location. Figure 4-1 shows the COP register.

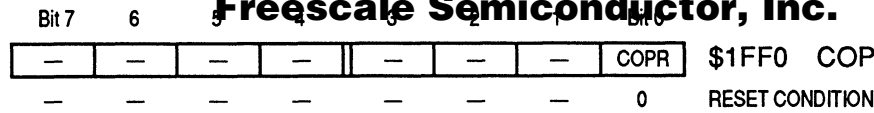


Figure 4-1. COP Control Register

COPR — COP Timer Reset

Periodically writing a zero to COPR prevents the COP watchdog timer from resetting the CPU.

4.2 Interrupts

An interrupt temporarily stops normal processing so that some unusual event can be processed. Unlike a reset, an interrupt does not stop the current instruction. An interrupt is considered pending until the current instruction is complete. There are three kinds of CPU interrupts:

- External interrupt — If the interrupt mask is a logical zero, and the external interrupt pin (\overline{IRQ}) goes to logical zero, then the CPU recognizes an external interrupt.
- Capture/compare timer interrupt — When the interrupt mask is a logical zero, the CPU can recognize interrupts from the capture/compare timer. If one of the three timer interrupt flags (ICF, OCF, TOF) goes to logical one in the timer status register, and its corresponding interrupt enable bit (ICIE, OCIE, TOIE) in the timer control register is a logical one, a timer interrupt is requested.
- Software interrupt — The software interrupt is an executable instruction. It is executed regardless of the state of the interrupt mask.

The following internal actions occur as a result of any CPU interrupt:

- CPU register contents are stored on the stack in the order PCL, PCH, X, A, CCR.
- The interrupt mask is automatically set to prevent additional interrupts.
- An interrupt vector is fetched that causes processing to continue at the starting address of the interrupt routine.
- The RTI (return from interrupt) instruction causes the register contents to be recovered from the stack in the order CCR, A, X, PCH, PCL. Normal processing resumes.

Figure 4-2 shows the stacking and recovery sequence.

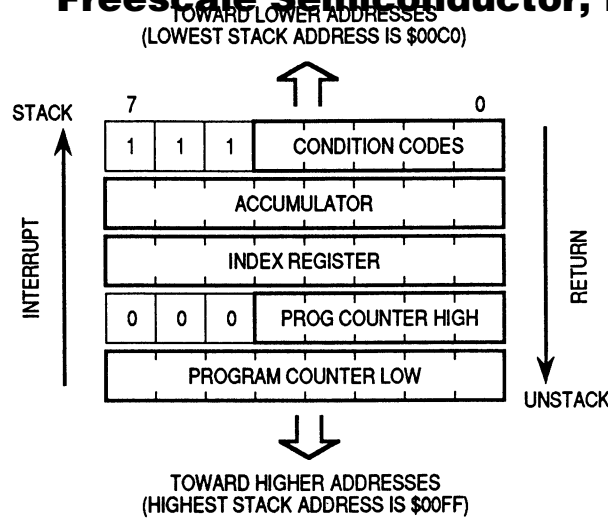


Figure 4-2. Interrupt Stacking Order

As each instruction is completed, the CPU checks for the presence of enabled external interrupt requests and enabled timer interrupt requests. For an external interrupt request to be recognized, the interrupt mask (I-bit) in the condition code register must be a logical zero. If the interrupt mask is set or if no qualified interrupt request is pending, the processor fetches and executes the next program instruction.

For a timer interrupt request to be recognized, the interrupt mask must be a logical zero, and the corresponding interrupt enable bit (OCIE, ICIE, or TOIE) in the timer control register must be a logical one. If the interrupt mask is set or if no qualified interrupt request is pending, the processor fetches and executes the next program instruction.

If both an external interrupt and a capture/compare timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first.

A software interrupt (SWI) is executed as an instruction, regardless of the state of the interrupt mask. Figure 4-3 shows how interrupts relate to normal instruction execution. CPU control logic determines sequence of operations.

4.2.1 External Interrupt

The CPU recognizes an external interrupt when the external interrupt pin (\overline{IRQ}) goes to a logical zero while the interrupt mask (I-bit) is at logical zero. The current state of the CPU is pushed onto the stack, and the interrupt mask is set to inhibit further interrupts until the present one is serviced. The address of the interrupt service routine is contained in memory locations \$1FFA and \$1FFB.

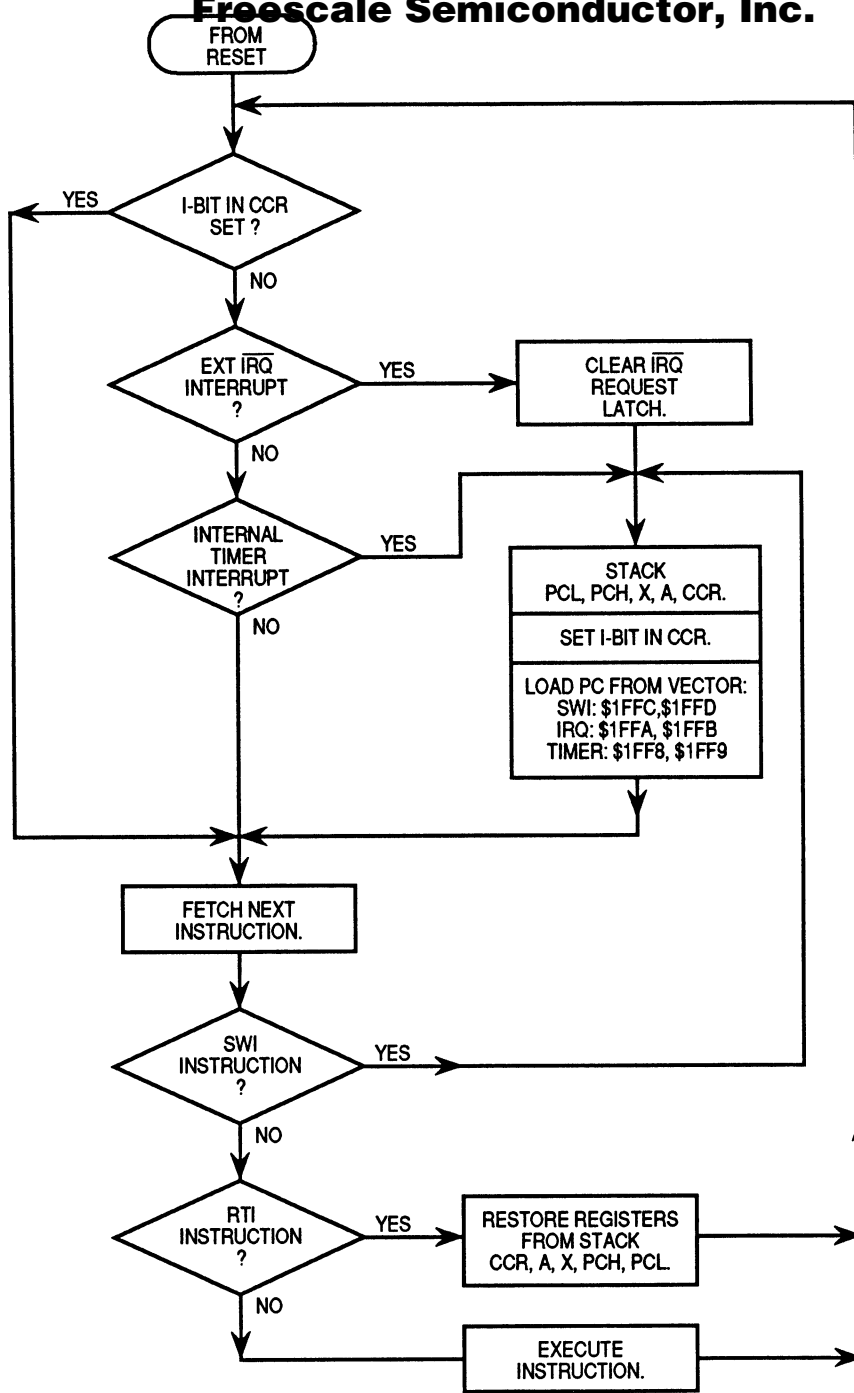


Figure 4-3. Reset and Interrupt Flowchart

Either an edge-sensitive and level-sensitive trigger or an edge-sensitive-only trigger is available as a factory-set mask option. Figure 4-4 shows the internal logic of this mask option. The internal interrupt latch is cleared while the interrupt vector is being fetched. During the interrupt service routine, a new interrupt request can be initiated and latched. As soon as the interrupt mask is

cleared (usually during the return from interrupt), the latched request is recognized and serviced.

The level-sensitive trigger option allows multiple interrupt sources to be wire-ORed to the $\overline{\text{IRQ}}$ pin. As long as any source is holding the $\overline{\text{IRQ}}$ pin at logical zero, an external interrupt request is considered to be pending.

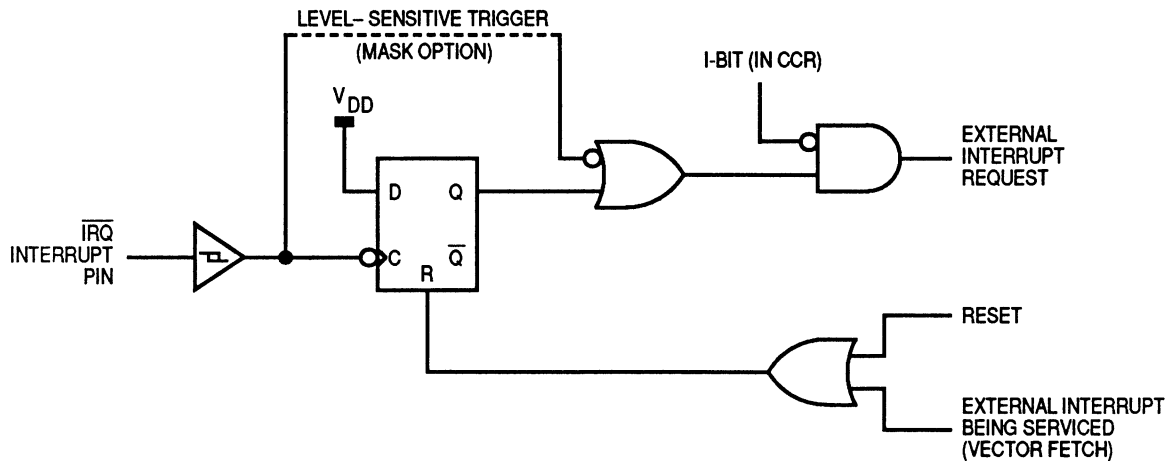


Figure 4-4. $\overline{\text{IRQ}}$ Mask Option Logic

4.2.2 Software Interrupt (SWI)

The SWI is an executable instruction. The SWI instruction is executed regardless of the state of the interrupt mask (I-bit) in the condition code register. The address of the SWI interrupt service routine is in memory locations \$1FFC and \$1FFD.

4.2.3 Capture/Compare Timer Interrupt

Three interrupts can be generated by the capture/compare timer when the interrupt mask (I-bit) is a logical zero. When one of the three timer interrupt flags in the timer status register is at logical one, and the corresponding interrupt enable flag in the timer control register is at logical one, the CPU recognizes a timer interrupt. (See **SECTION 6 CAPTURE/COMPARE TIMER** for more information.) All three timer interrupts use the same interrupt vector at \$1FF8 and \$1FF9.

SECTION 5 MEMORY

This section describes the organization of the on-chip memory. The CPU of the MC68HC05P4 MCU can address 8K bytes of memory space.

5.1 Memory Map

The program counter normally advances one address at a time through the on-chip memory, reading the instructions and data necessary to execute the program. The ROM portion of memory holds the program instructions, user-defined vectors, and service routines. The RAM portion of memory holds variable data. I/O, control, and status registers are memory-mapped so that the CPU can access their locations the same way it accesses any other memory location.

On-chip ROM includes 4160 bytes of factory-programmed memory for storage of application program instructions and fixed data. The last eight memory addresses (\$1FF8–\$1FFF) are ROM addresses that contain user-defined vectors for servicing interrupts and resets. When ordering the MCU, the user specifies the instructions and data to be programmed into the user ROM.

The 240 bytes between \$1F00 and \$1FEF are reserved ROM addresses that contain the instructions for a series of self-check tests.

The MCU has 176 bytes of fully static read-write memory for storage of variable and temporary data during program execution. The CPU uses the top 64 RAM addresses (\$00C0–\$00FF) for the stack. The CPU uses the stack to save CPU register contents before processing an interrupt or subroutine call. The stack pointer decrements during pushes and increments during pulls.

The first 32 bytes of the memory space contain port data registers, port data direction registers, SIOP control, status, and data registers, and timer control, status, and counter registers.

Figure 5-1 is a memory map of the MCU, and Figure 5-2 is a more detailed memory map of the 32-byte I/O register area.

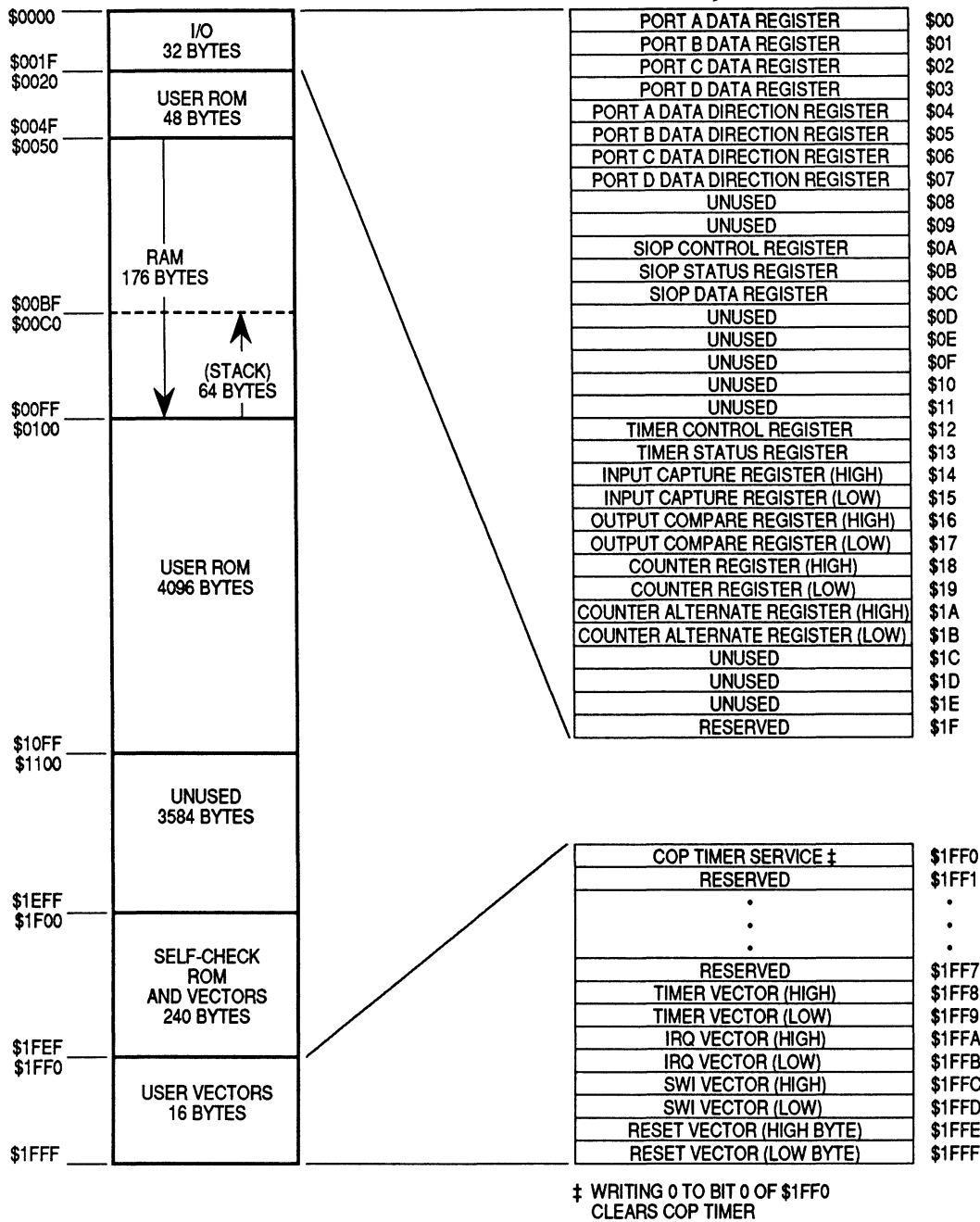


Figure 5-1. MC68HC05P4 Memory Map

NOTE

Using the stack area for data storage or as a temporary work area requires care to prevent data from being overwritten due to stacking from an interrupt or subroutine call.

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
|--------|----------|---------|---------|-------|-------|-------|-------|-------|---------------------------|
| \$0000 | I/O | I/O | I/O | I/O | I/O | I/O | I/O | I/O | PORTA |
| | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | PORT A PIN NUMBERS (REF.) |
| | PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 | PORT A PIN NAMES (REF.) |
| \$0001 | I/O | I/O | I/O | 0 | 0 | 0 | 0 | 0 | PORTB |
| | 13 | 12 | 11 | - | - | - | - | - | PORT B PIN NUMBERS (REF.) |
| | PB7/SCK | PB6/SDI | PB5/SDO | - | - | - | - | - | PORT B PIN NAMES (REF.) |
| \$0002 | I/O | I/O | I/O | I/O | I/O | I/O | I/O | I/O | PORTC |
| | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | PORT C PIN NUMBERS (REF.) |
| | PC7 | PC6 | PC5 | PC4 | PC3 | PC2 | PC1 | PC0 | PORT C PIN NAMES (REF.) |
| \$0003 | I only | 0 | I/O | 1 | 0 | 0 | 0 | 0 | PORTD |
| | 25 | - | 23 | - | - | - | - | - | PORT D PIN NUMBERS (REF.) |
| | PD7/TCAP | - | PD5 | - | - | - | - | - | PORT D PIN NAMES (REF.) |
| \$0004 | DDRA7 | DDRA6 | DDRA5 | DDRA4 | DDRA3 | DDRA2 | DDRA1 | DDRA0 | DDRA |
| \$0005 | DDRB7 | DDRB6 | DDRB5 | 1 | 1 | 1 | 1 | 1 | DDRB |
| \$0006 | DDRC7 | DDRC6 | DDRC5 | DDRC4 | DDRC3 | DDRC2 | DDRC1 | DDRC0 | DDRC |
| \$0007 | 0 | 0 | DDRD5 | 0 | 0 | 0 | 0 | 0 | DDRD |
| \$0008 | | | | | | | | | Unused |
| \$0009 | | | | | | | | | Unused |
| \$000A | 0 | SPE | 0 | MSTR | 0 | 0 | 0 | 0 | SCR |
| \$000B | SPIF | DCOL | - | - | - | - | - | - | SSR |
| \$000C | Bit 7 | | | | | | | Bit 0 | SDR |
| \$000D | | | | | | | | | Unused |
| \$000E | | | | | | | | | Unused |
| \$000F | | | | | | | | | Unused |
| \$0010 | | | | | | | | | Unused |
| \$0011 | | | | | | | | | Unused |
| \$0012 | ICIE | OCIE | TOIE | 0 | 0 | 0 | IEDG | OLVL | TCR |
| \$0013 | ICF | OCF | TOF | 0 | 0 | 0 | 0 | 0 | TSR |
| \$0014 | Bit 15 | | | | | | | Bit 8 | TCAP (HIGH) |
| \$0015 | Bit 7 | | | | | | | Bit 0 | TCAP (LOW) |
| \$0016 | Bit 15 | | | | | | | Bit 8 | TCMP (HIGH) |
| \$0017 | Bit 7 | | | | | | | Bit 0 | TCMP (LOW) |
| \$0018 | Bit 15 | | | | | | | Bit 8 | TCNT (HIGH) |
| \$0019 | Bit 7 | | | | | | | Bit 0 | TCNT (LOW) |
| \$001A | Bit 15 | | | | | | | Bit 8 | ALTCNT (HIGH) |
| \$001B | Bit 7 | | | | | | | Bit 0 | ALTCNT (LOW) |
| \$001C | | | | | | | | | Unused |
| \$001D | | | | | | | | | Unused |
| \$001E | | | | | | | | | Unused |
| \$001F | | | | | | | | | RESERVED |
| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
| \$1FF0 | - | - | - | - | - | - | - | COPR | COP |

READS ACCESS A ROM LOCATION;
WRITES ACCESS THE COP WATCHDOG RESET LOGIC.

Figure 5-2. I/O and Control Register Summary



In data-retention mode, the MCU retains RAM contents and CPU register contents at V_{DD} voltages as low as 2.0 Vdc. The $\overline{\text{RESET}}$ line must be driven to logical zero before the V_{DD} voltage is lowered, and $\overline{\text{RESET}}$ must remain low continuously during data-retention mode. The data-retention feature allows the MCU to be left in a low power-consumption mode during which data is held, but the CPU cannot execute instructions. To exit the data-retention mode, V_{DD} must be returned to its normal operating voltage before allowing $\overline{\text{RESET}}$ to go to logical one.

SECTION 6

CAPTURE/COMPARE TIMER

This section describes the operation of the capture/compare timer. The capture/compare timer provides a means to latch the times at which external events occur, to measure input waveforms, and to generate output waveforms and timing delays. A 16-bit free-running counter, preceded by a prescaler that divides the internal clock by four, provides the timing reference for the input capture and output compare functions.

PD7/TCAP is the input pin for the input capture function, and TCMP is the output pin for the output compare function. The timer uses 10 addressable 8-bit registers:

- Counter high register (\$18)
- Counter low register (\$19)

- Alternate counter high register (\$1A)
- Alternate counter low register (\$1B)

- Input capture high register (\$14)
- Input capture low register (\$15)

- Output compare high register (\$16)
- Output compare low register (\$17)

- Timer control register (\$12)

- Timer status register (\$13)

Since the capture/compare timer has a 16-bit architecture, the counter values and the capture and compare values are stored in pairs of 8-bit registers. One of the 8-bit registers contains the high byte, and the other contains the low byte.

Figure 6-1 shows the structure of the capture/compare timer.

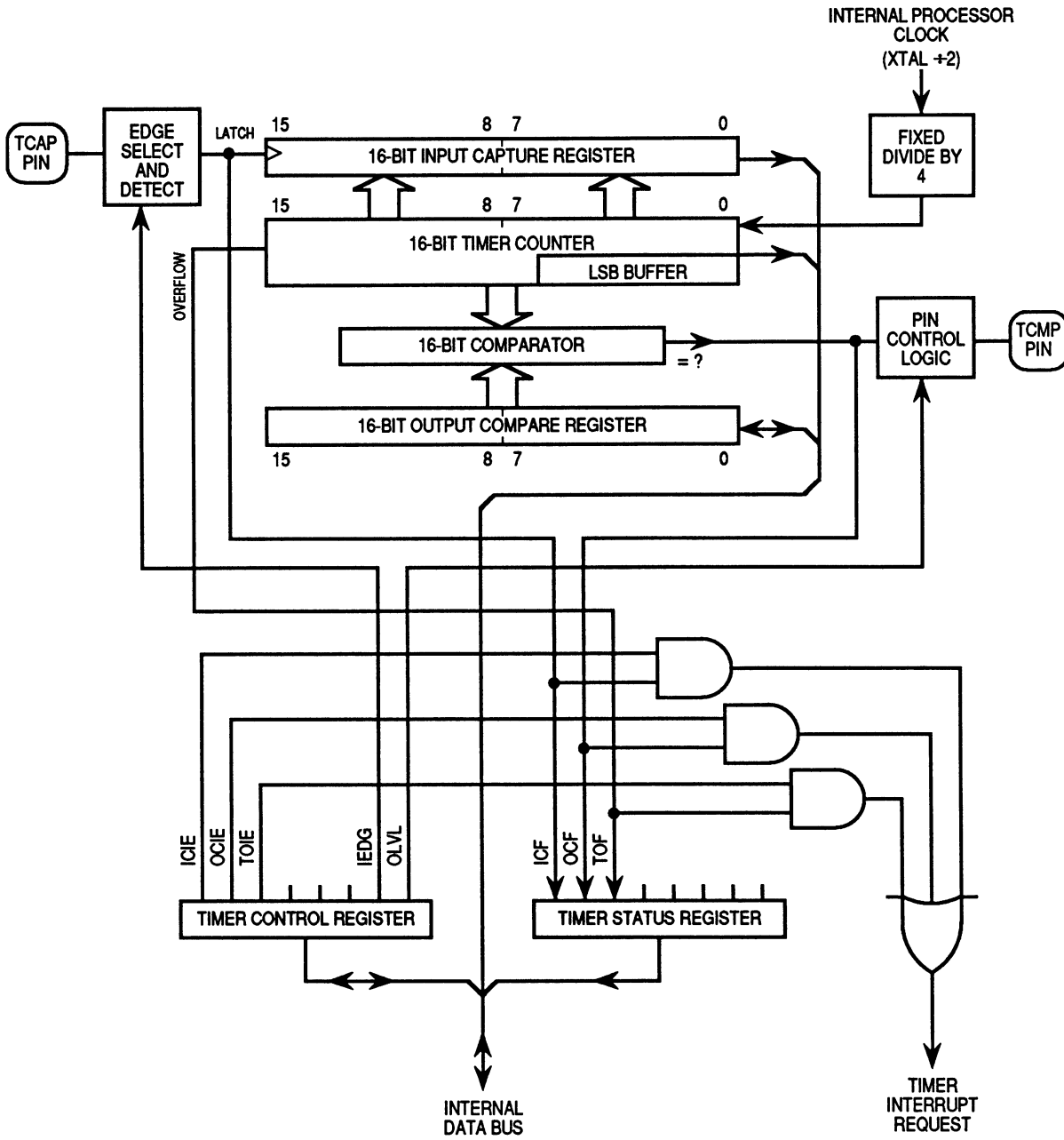


Figure 6-1. Capture/Compare Timer Block Diagram

6.1 Input Capture Operation

The input capture feature provides a means to record the time at which an external event occurs. When the timer detects a selected (negative-going or positive-going) edge on the TCAP pin, it latches the contents of the free-running counter into the input capture register. The IEDG bit in the timer control register allows software to select the edge polarity that triggers the input capture function. The ICIE bit in the timer control register allows software to determine whether or not the input capture function generates a hardware interrupt request. (See Figure 6-2.)

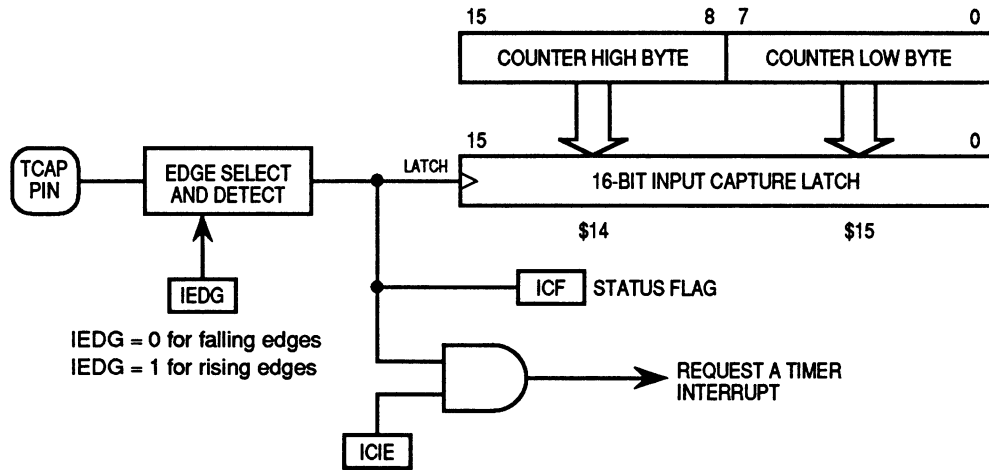


Figure 6-2. Input Capture Operation

Latching the counter values at successive edges of the same polarity measures the period of the input signal on the TCAP pin. Latching the counter values at successive edges of opposite polarity measures the pulse width of the signal.

6.2 Output Compare Operation

The output compare feature provides a means to generate an output signal when the free-running counter reaches a selected value. The selected value is written into the output compare register. On every fourth internal clock cycle the capture/compare timer compares the value of the free-running counter to the contents of the output compare register. When a match occurs, the timer transfers the output level bit (OLVL) from the timer control register to the TCMP output pin and sets the output compare flag (OCF), which optionally generates an interrupt. (See Figure 6-3.)

Freescale Semiconductor, Inc.

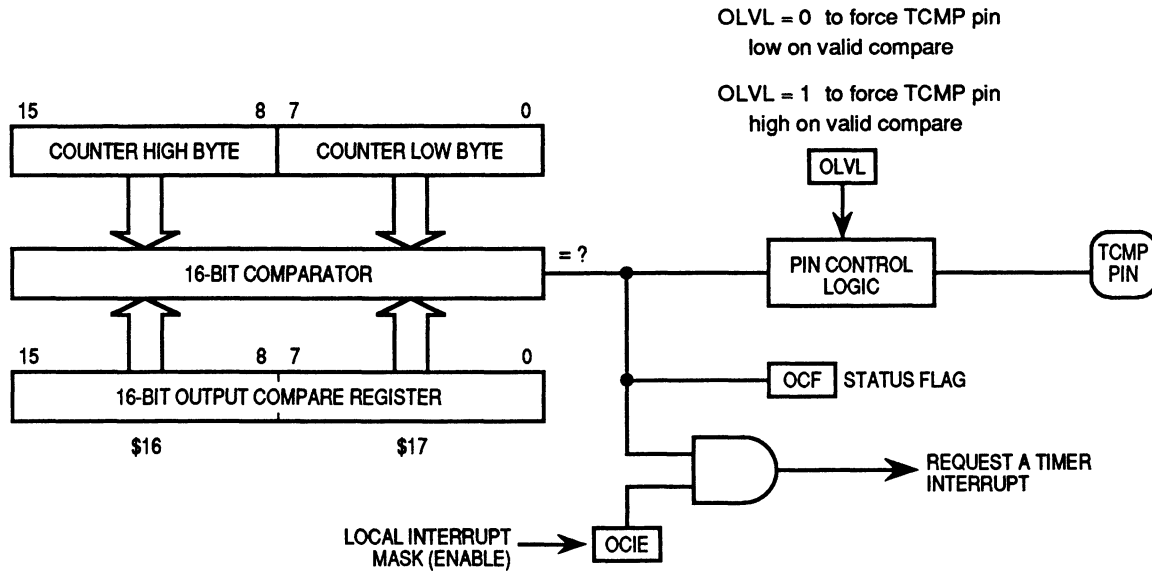


Figure 6-3. Output Compare Operation

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

6.3 Timer Counter

The key element in the programmable capture/compare timer is a 16-bit, free-running counter, preceded by a prescaler that divides the internal clock by four. Software can read the counter at any time without affecting its counting sequence.

The high and low bytes of the free-running counter can be read from the counter register at locations \$18 and \$19 or from the counter alternate register at locations \$1A and \$1B. (See Figure 6-4.) Reading the counter register low byte is one step in the procedure for clearing the timer overflow flag (TOF), but reading the counter alternate register does not affect TOF. Therefore, the counter alternate register can be read at any time without risk of missing timer overflow interrupts due to a cleared TOF. Normally, the timer counter is read from the counter alternate register unless the read sequence is intended to clear TOF.

To prevent interrupts from occurring between readings of the high and low bytes of the counter register or counter alternate register, the interrupt mask (I-bit) can be set before reading the high byte and cleared after reading the low byte.

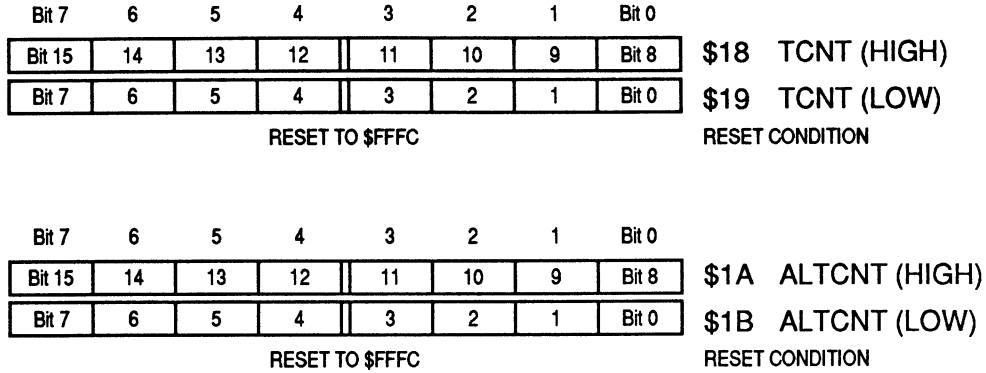
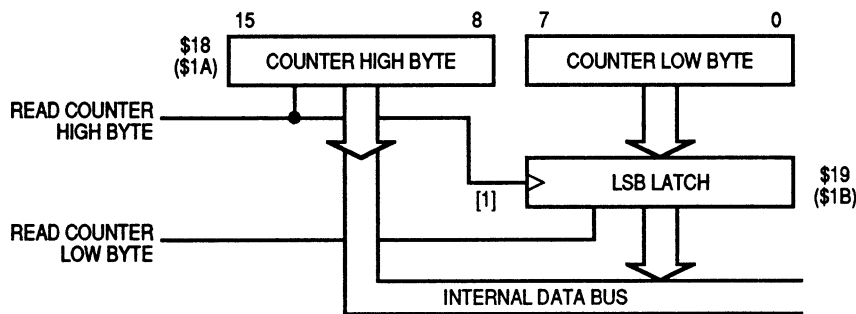


Figure 6-4. Counter Register/Counter Alternate Register

Reading the high byte of the counter register or counter alternate register accesses the high byte value at the time of the read and causes the low byte to be latched into a buffer. (See Figure 6-5.) This buffer value remains fixed after the first high-byte read, even if the high byte is read more than once. The buffer is accessed when the read sequence is completed by reading the low byte of the counter register or the counter alternate register. If the high byte is read, the low byte must also be read to complete the read sequence.



[1]LSB latch is normally transparent, becomes latched when high byte of counter is read, and becomes transparent again when low byte of counter is read.

Figure 6-5. 16-Bit Counter Reads

The free-running counter is preset to \$FFFC during reset and is always a read-only register. During a power-on reset, the counter is preset to \$FFFC and begins running after the oscillator startup delay. Because the free-running counter is 16 bits preceded by a fixed divide-by-four prescaler, the value in the counter repeats every 262,144 internal clock cycles. When the counter rolls over from \$FFFF to \$0000, the TOF bit is set. An interrupt is requested when counter rollover occurs if the timer overflow interrupt enable bit (TOIE) is set.

6.4 Output Compare Register

The high and low bytes of the output compare register are at memory locations \$16 and \$17. (See Figure 6-6.) All bits are readable and writable and are not altered by the capture/compare timer hardware or by a reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

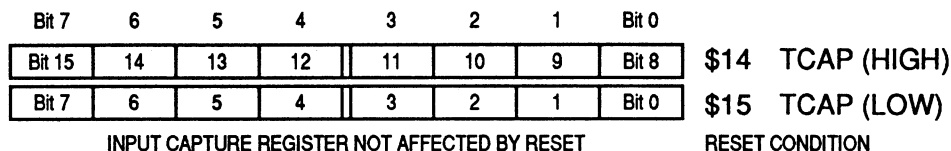


Figure 6-6. Input Capture Register

The output compare register contents are continually compared with the contents of the free-running counter. When a match occurs, the output compare flag (OCF) is set, and the OLVL bit is clocked to the TCMP output pin. OLVL appears on TCMP whether or not OCF was previously set. An output compare interrupt is enabled if the output compare interrupt enable bit (OCIE) is set. The output compare register values and the output level bit are typically changed after each successful comparison to establish a new timeout period. Writing to either byte of the output compare register does not affect the other byte.

Writing the high byte of the output compare register inhibits the output compare function until the low byte is also written. Both bytes must be written if the high byte is written first. Writing only the low byte does not inhibit the output compare function.

6.5 Input Capture Register

The high and low bytes of the input capture register are at memory locations \$14 and \$15. (See Figure 6-7.) The input capture register is a read-only register and is not affected by a reset.

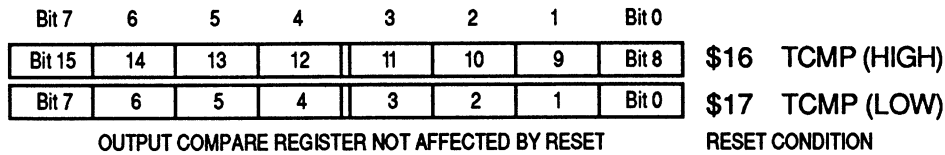


Figure 6-7. Output Compare Register

When the input capture edge detector senses a defined transition on the TCAP pin, the input capture flag (ICF) is set, and the input capture register latches the value of the free-running counter. The counter contents are transferred to the input capture register on every defined signal transition whether or not ICF was previously set. The input capture register always contains the free-running counter value at the time of the most recent input capture. The polarity of the level transition that triggers the counter capture is defined by the input edge bit (IEDG).

The counter increments every fourth cycle of the internal clock. The counter value latched into the input capture register is one count more than the count at the time of the last rising edge of the clock before the defined transition on the TCAP pin occurred. This delay is required for internal synchronization.

Reading the high byte of the input capture register inhibits the input capture function until the low byte is also read. Both bytes must be read if the high byte is read first. Reading only the low byte does not inhibit the input capture function.

NOTE

To prevent interrupts from occurring between readings of the high and low bytes of the input capture register, the interrupt flag can be set before reading the high byte and cleared after reading the low byte.

6.6 Timer Status Register (TSR)

TSR is a read-only register with three status flags that indicate the following conditions:

- A selected transition occurred at the TCAP pin, and the contents of the free-running counter were transferred to the input capture register.
- A match occurred between the free-running counter and the output compare register, and the OLVL bit was transferred to the TCMP pin.
- A free-running counter transition from \$FFFF to \$0000 occurred.

| | | | | | | | | |
|-------|-----|-----|---|---|---|---|-------|-----------------|
| Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
| ICF | OCF | TOF | 0 | 0 | 0 | 0 | 0 | \$13 TSR |
| U | U | U | 0 | 0 | 0 | 0 | 0 | RESET CONDITION |

(U = UNAFFECTED)

Figure 6-8. Timer Status Register

ICF — Input Capture Flag

ICF is automatically set when an edge of the selected polarity occurs on TCAP. The flag is cleared by reading the low byte (\$15) of the input capture register after reading TSR with ICF set.

OCF — Output Compare Flag

OCF is automatically set when the value of the free-running counter matches the contents of the output compare register. The flag is cleared by accessing the low byte (\$17) of the output compare register after reading TSR with OCF set.

TOF — Timer Overflow Flag

TOF is automatically set when the free-running counter changes from \$FFFF to \$0000. The flag is cleared by accessing the low byte (\$19) of the free-running counter after reading TSR with TOF set.

Bits 4–0 — Not used; always read zero

Reading the capture/compare timer status register is the first step in clearing a status bit. The remaining step is to access the low byte of the register associated with the status bit.

A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. TOF could unintentionally be cleared by reading TSR and the low byte of the free-running counter, but not for the purpose of servicing the flag.

The counter alternate register at locations \$1A and \$1B contains the same value as the counter register at locations \$18 and \$19. Reading the counter alternate register has no effect on TSR. Therefore, the counter alternate register can be read at any time without clearing TOF.

6.7 Timer Control Register (TCR)

TCR is a read/write register with five control bits. Three bits control interrupts associated with the TSR flags ICF, OCF, and TOF. Another bit determines the edge polarity (positive-going or negative-going) that activates the input capture edge detector. Another bit determines the output level to be clocked onto TCMP when a successful output compare occurs.

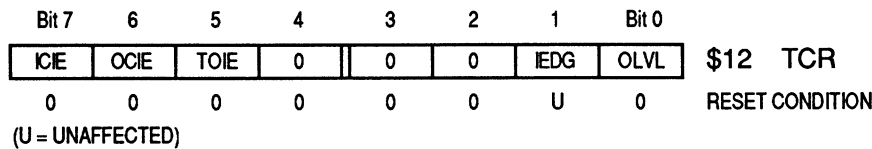


Figure 6-9. Timer Control Register

- ICIE — Input Capture Interrupt Enable
 - 1 = ICF interrupt enabled
 - 0 = ICF interrupt disabled

- OCIE — Output Compare Interrupt Enable
 - 1 = OCF interrupt enabled
 - 0 = OCF interrupt disabled

- TOIE — Timer Overflow Interrupt Enable
 - 1 = TOF interrupt enabled
 - 0 = TOF interrupt disabled

IEDG — Input Edge

This bit determines which level transition on the TCAP pin triggers a free-running counter transfer to the input capture register.

1 = Positive edge (low level to high level)

0 = Negative edge (high level to low level)

OLVL — Output Level

This bit determines the output level on the TCMP pin when a successful output compare occurs.

1 = High output

0 = Low output

Bits 4, 3, and 2 — Not used; always read zero

6.8 Timer during WAIT Mode

The internal clock halts during WAIT mode, but the capture/compare timer and COP counter remain active. An interrupt from the capture/compare timer causes the processor to exit WAIT mode.

6.9 Timer during STOP Mode

In STOP mode, the capture/compare timer stops counting and holds the last count value. If \overline{IRQ} is used to exit STOP mode, the timer resumes counting from the count value that was present when STOP mode was entered. If \overline{RESET} is used, the counter is forced to \$FFFC.

If a defined transition occurs on the TCAP pin during STOP mode, ICF goes high as soon as an external interrupt brings the MCU out of STOP mode. If a power-on reset or a logical zero on the \overline{RESET} pin brings the MCU out of STOP mode, all timer interrupt enable bits are cleared.

SECTION 7 SERIAL I/O PORT SYSTEM (SIOP)

This section describes the serial I/O port system. The SIOP provides for simple high-speed synchronous serial data transfer to allow the MCU to communicate with peripheral devices. The SIOP can be used with simple shift registers to

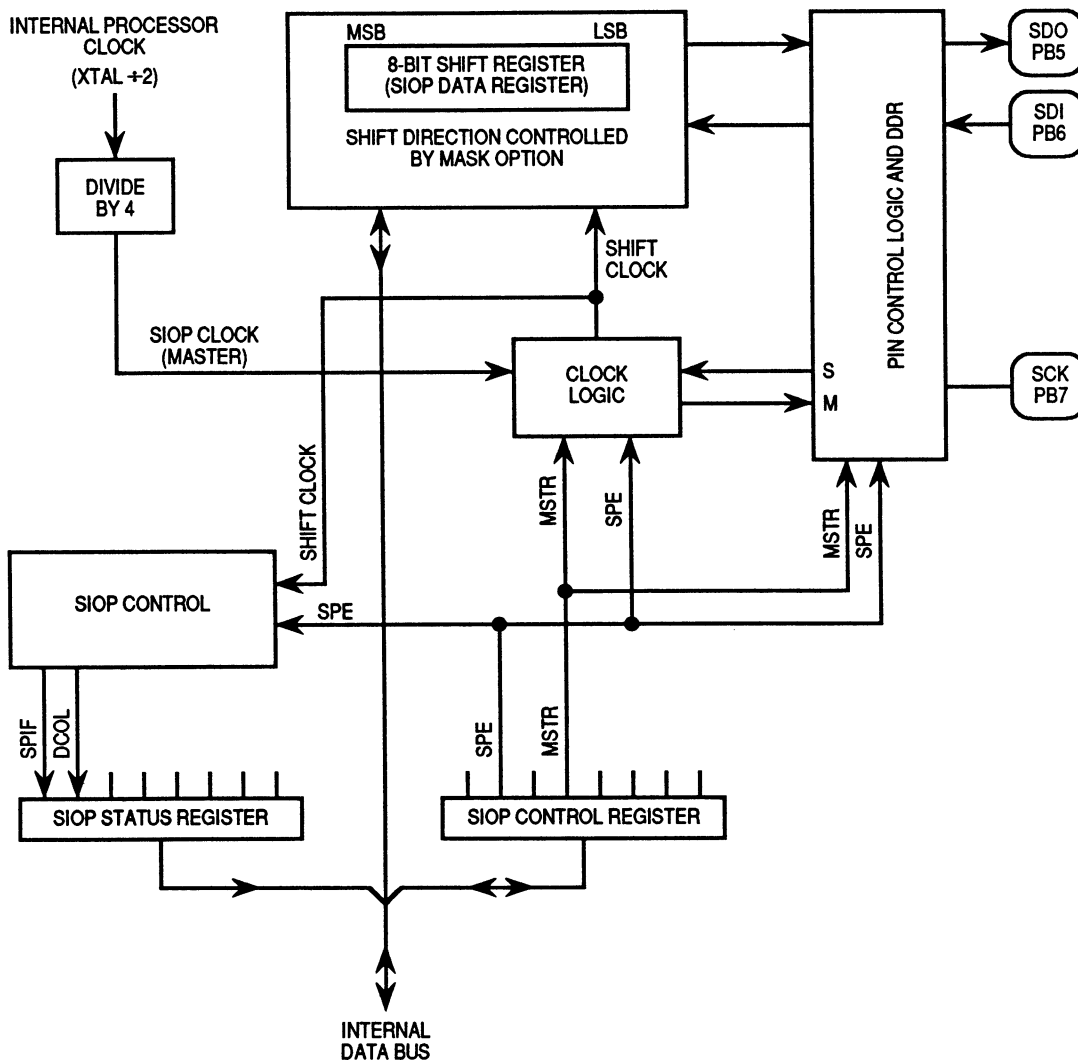
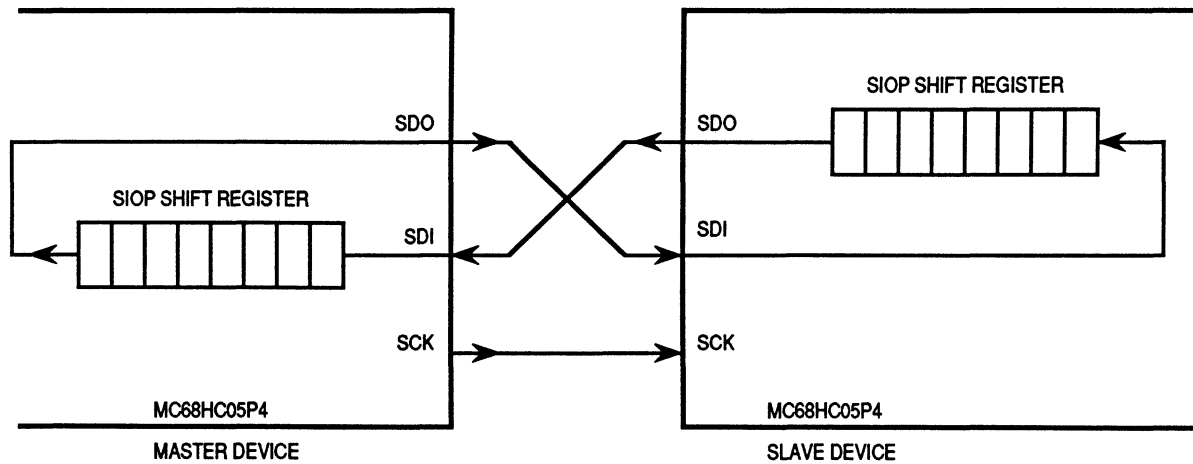


Figure 7-1. Serial I/O Port (SIOP) Block Diagram

increase the number of parallel I/O pins controlled by the MCU. More powerful peripherals such as analog-to-digital (A/D) converters and real-time clocks are also compatible with this interface. A factory-set mask option in the MC68HC05P4 allows the SIOP to transfer data MSB first or LSB first. Figure 7-1 shows the structure of the SIOP system.

The SIOP data register in a master MCU and the SIOP data register in a slave MCU are connected to form a 16-bit circular shift register. During an SIOP transfer, the master shifts out the contents of its SIOP data register on its SDO pin. At the same time, the slave MCU shifts out the contents of its SIOP data register on its SDO pin, so that the master and slave exchange the contents of their data registers. (See Figure 7-2.)



NOTE: Both MC68HC05P4 devices shown have the MSB-first mask option.

Figure 7-2. SIOP Shift Register Operation

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 output 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 system is simpler than the SPI system on some other Motorola MCUs. The polarity of the serial clock (SCK) is fixed. There is no slave select pin on the SIOP system, and 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.

7.1 SIOP Pin Descriptions

PB7/SCK, PB6/SDI, and PB5/SDO form the 3-bit shared-function I/O port B. Port B can be either the SIOP or a general-purpose I/O port.

When bit 6 (SPE) of the SIOP control register (SCR) is a logical one, port B is dedicated to SIOP functions. When SPE is cleared, port B reverts to standard parallel I/O without affecting the port B data register or data direction register.

After SPE is set, the SDO output driver can be disabled by writing a zero to DDRB5, configuring SDO as a high-impedance input.

NOTE

Port B should not be used for general-purpose I/O while the SIOP system is enabled.

When the master mode select (MSTR) bit of the SIOP control register is set, the SIOP is configured for master mode. The SCK pin is an output whose signal is derived from the internal clock. SDI is the serial input, and SDO is the serial output. 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 SDR is parallel-loaded and shifted out serially to the slave device(s). MSTR may be set regardless of the state of SPE.

When MSTR is a logical zero, the SIOP is configured for slave mode. SDI and SDO have the same functions that they do in master mode, but SCK is configured as an input.

7.1.1 SIOP Clock

SCK synchronizes the movement of data into and out of the MCU through the SDI and SDO pins. In master mode, the SCK pin is an output. The transmission rate for master mode is one-fourth the internal clock rate. For example, if the OSC1 input frequency is 4 MHz, the internal clock frequency is 2 MHz, and the SCK frequency is 0.5 MHz.

.n slave mode, the SCK pin is an input. The maximum SCK input rate for slave mode is the internal clock divided by four. There is no minimum SCK frequency for slave mode.

Figure 7-3 shows the timing relationships between SCK, SDI, and SDO. The state of SCK between transmissions is logical one. The first falling edge of SCK signals the beginning of a transmission, and data is presented to the SDO pin on the falling edge of SCK. Data is captured at the SDI pin on the rising edge of SCK, and the transmission is ended on the eighth rising edge of SCK.

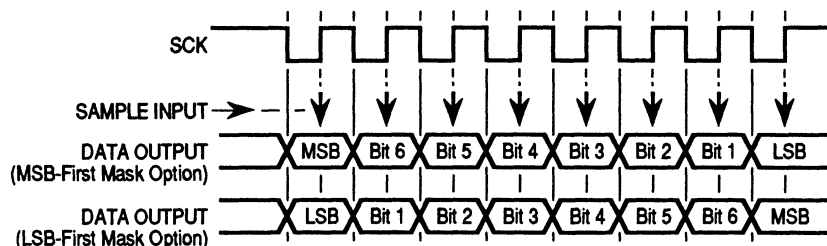


Figure 7-3. SIOP Data/Clock Timing Diagram

7.1.2 SIOP Data Input

The SDI pin becomes a serial input as soon as the SIOP is enabled. As shown in Figure 7-3, valid SDI data must be present for an SDI setup time before the rising edge of SCK and remain valid for an SDI hold time after the rising edge of SCK. (See 9.8 SIOP Timing ($V_{DD} = 5.0 \text{ Vdc}$) and 9.9 SIOP Timing ($V_{DD} = 3.3 \text{ Vdc}$.)

7.1.3 SIOP Data Output

The SDO pin becomes a serial output as soon as the SIOP is enabled. Before a transfer, the state of the SDO pin reflects the value of the last bit received on the previous transmission, if one occurred. To preset the beginning state, PB5 can be written before the SIOP is enabled. SDO cannot be used as a standard output while the SIOP is enabled, because it is coupled to the last stage of the serial shift register. On the first falling edge of SCK, the first data bit to be shifted out is presented to the SDO pin.

SCR is a read/write register containing only two bits. (See Figure 7-4.) One bit enables the SIOP, and the other configures the SIOP for master or slave mode.

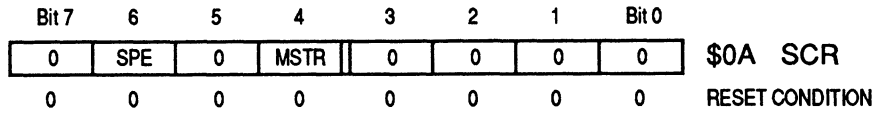


Figure 7-4. SIOP Control Register

SPE — Serial Port Enable

SPE is readable and writable any time, but clearing SPE during a transmission aborts the transmission, resets the bit counter, and returns port B to its normal I/O function.

- 1 = Enables the SIOP and initializes the port B data direction register such that SCK is an input (in slave mode) or an output (in master mode), SDI is an input, and SDO is an output
- 0 = Disables the SIOP and returns port B to its normal I/O function

MSTR — Master Mode Select

Clearing MSTR aborts any transmission in progress.

- 1 = Configures SIOP as a master
- 0 = Configures SIOP as a slave

Bits 7, 5, and 3–0 — Not used; always read zero

7.3 SIOP Status Register (SSR)

SSR is a read-only register containing only two bits. (See Figure 7-5.) One bit indicates that an SIOP transfer is complete, and the other indicates that an invalid access of the SDR occurred while a transfer was in progress.

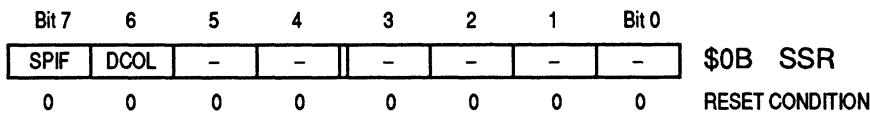


Figure 7-5. SIOP Status Register

SPIF — SIOPI Peripheral Interface Transfer Complete Flag

SPIF is automatically set on the eighth rising edge of SCK and indicates that a data transfer took place. SPIF does not inhibit further transmissions and can be ignored in master mode. The flag is cleared by reading SSR while SPIF is set, and then reading or writing SDR. SPIF is also cleared by a reset.

DCOL — Data Collision Flag

DCOL is automatically set if SDR is accessed while a data transfer is in progress. Reading or writing SDR while a data transfer is in progress results in invalid data being transmitted or read. The flag is cleared by reading SSR with SPIF set, and then reading or writing SDR. To clear DCOL when SPIF is not set, the SIOPI must be turned off by writing a zero to SPE and then turned back on by writing a one to SPE. If the clearing sequence is not completed before another transmission starts, DCOL is set again. DCOL is also cleared by a reset.

Bits 5–0 — Not used; always read zero

7.4 SIOPI Data Register (SDR)

SDR is both the transmit and receive data register. (See Figure 7-6.)

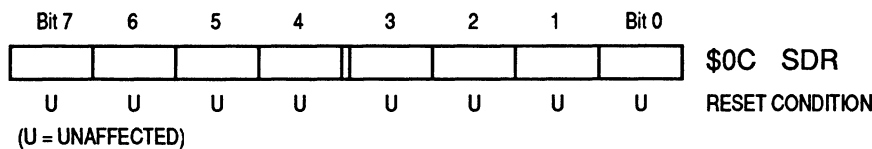


Figure 7-6. SIOPI Data Register

This register is not double buffered, i.e., writing to SDR overwrites the previous contents. Reading or writing to SDR while a transmission is in progress can cause invalid data to be transmitted or received. SDR can be read and written only when the SIOPI is enabled (SPE = 1).

SECTION 8 SELF-CHECK MODE

This section describes how to use the self-check mode to test the operation of the MCU.

8.1 Self-Check Circuit

The self-check function determines if the MCU is functioning properly. The self-check circuit is shown in Figure 8-1. If 9 Vdc is applied to the $\overline{\text{IRQ}}$ pin, and a logical one is applied to the TCAP/PD7 pin, the MCU enters the self-check mode when reset. Port C pins PC3–PC0 are monitored for the self-check results. The following six tests are performed automatically in self-check mode:

- I/O — Functional test of ports A, B, and C
- RAM — Counter test for each RAM byte
- ROM — Checksum of entire ROM pattern
- Capture/compare timer — Test of counter register and OCF bit
- Interrupts — Test of external and capture/compare timer interrupts
- SIOPI — Test of data transmission from SDO to SDI in master mode

8.2 Self-Check Results

Table 8-1 gives the codes displayed by the light-emitting diodes to indicate the self-check results.

Table 8-1. Self-Check Results

| PC3 | PC2 | PC1 | PC0 | Remarks |
|------------|-----|-----|-----|---|
| 1 | 0 | 0 | 1 | Bad I/O |
| 1 | 0 | 1 | 0 | Bad RAM |
| 1 | 0 | 1 | 1 | Bad Capture/Compare Timer |
| 1 | 1 | 0 | 0 | Bad ROM |
| 1 | 1 | 0 | 1 | Bad SIOPI |
| 1 | 1 | 1 | 0 | Bad Interrupts or $\overline{\text{IRQ}}$ Request |
| Flashing | | | | Good Device |
| All Others | | | | Bad Device, Bad Port C, etc. |

NOTE: Zero indicates LED is on; 1 indicates LED is off.

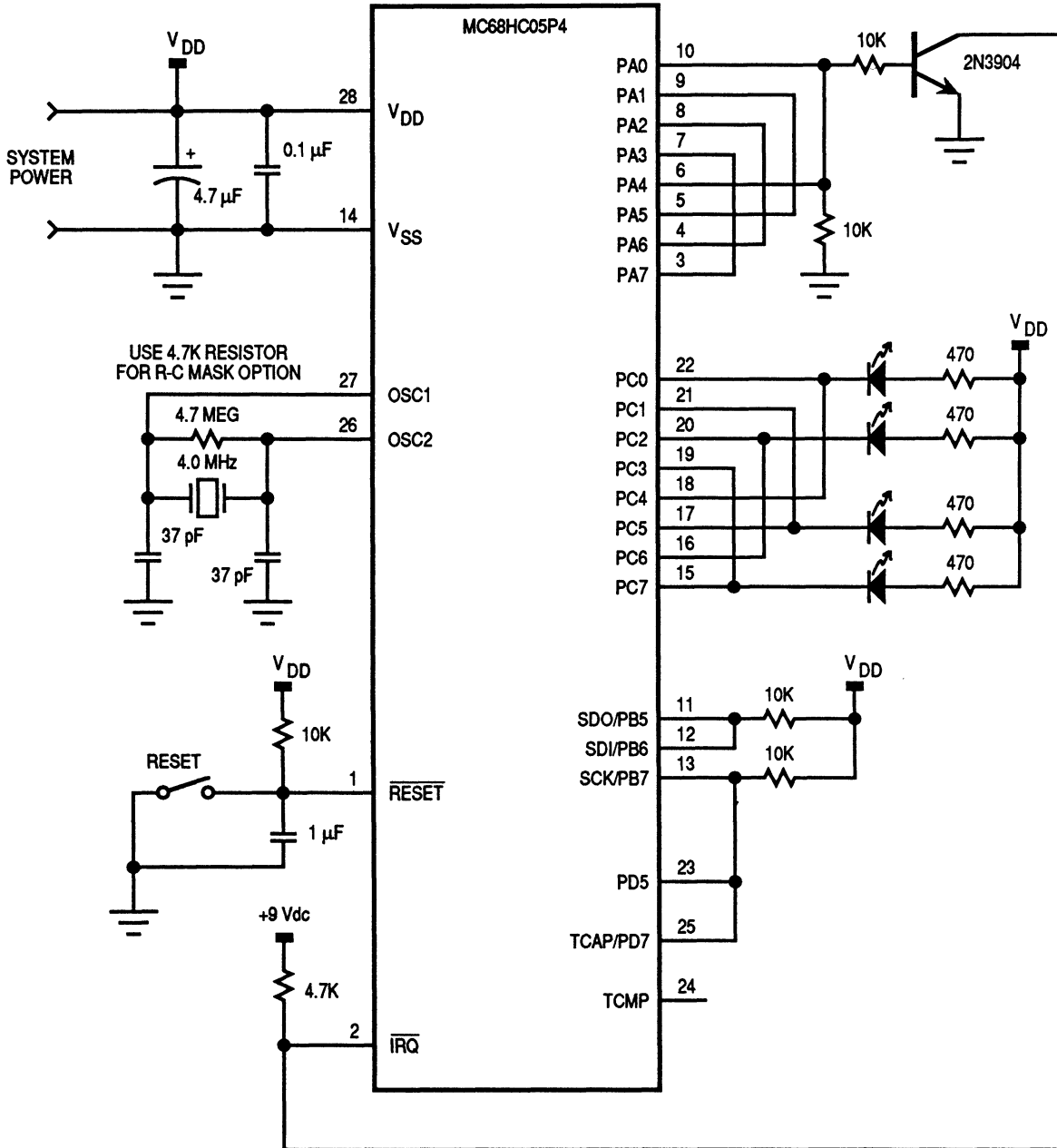


Figure 8-1. Self-Check Circuit Schematic

SECTION 9 ELECTRICAL SPECIFICATIONS

This section contains MCU electrical specifications and timing information.

9.1 Maximum Ratings

The MCU contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, precautions should be taken to avoid application of voltages higher than those shown in Table 9-1. For proper operation, it is recommended that V_{in} and V_{out} be kept within the range $V_{SS} \leq (V_{in} \text{ or } V_{out}) \leq V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logical voltage level (e.g., either V_{SS} or V_{DD}).

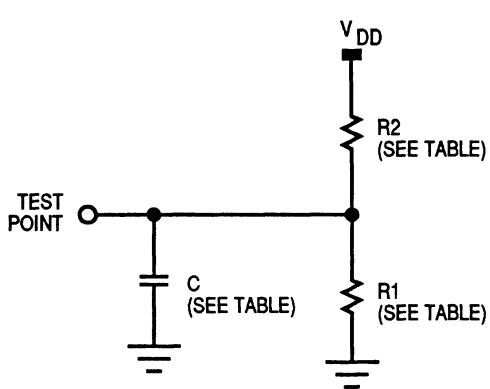
Table 9-1. Maximum Ratings

| Rating | Symbol | Value | Unit |
|--|-----------|---|-------------|
| Supply voltage | V_{DD} | -0.3 to +7.0 | V |
| Input voltage | V_{in} | $V_{SS} - 0.3$ to $V_{DD} + 0.3$ | V |
| Self-check mode (\overline{IRQ} pin only) | V_{in} | $V_{SS} - 0.3$ to $2 \times V_{DD} + 0.3$ | V |
| Current drain per pin (excluding V_{DD} and V_{SS}) | I | 25 | mA |
| Operating temperature range MC68HC05P4P (Standard) MC68HC05P4CP (Extended) | T_A | T_L to T_H 0 to 70 -40 to +85 | $^{\circ}C$ |
| Storage temperature range | T_{stg} | -65 to +150 | $^{\circ}C$ |

9.2 Thermal Characteristics

Table 9-2. Thermal Characteristics

| Characteristic | Symbol | Value | Unit |
|-----------------------------------|-----------------|-------|---------------|
| Thermal resistance Plastic DIP | $R_{\theta JA}$ | 60 | $^{\circ}C/W$ |



| V _{DD} = 4.5 V | | | |
|-------------------------|----------|---------|-------|
| Pins | R1 | R2 | C |
| PA7-PA0 | 3.26 kΩ | 2.38 kΩ | 50 pF |
| PB5-PB0 | | | |
| PC7-PC0 | | | |
| PD5, TCMP | | | |
| V _{DD} = 3.0 V | | | |
| Pins | R1 | R2 | C |
| PA7-PA0 | 10.91 kΩ | 6.32 kΩ | 50 pF |
| PB7-PB5 | | | |
| PC7-PC0 | | | |
| PD5, TCMP | | | |

Figure 9-1. Test Load

Freescale Semiconductor, Inc.

9.3 Power Considerations

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

$$T_J = T_A + (P_D \times R_{\theta JA}) \tag{1}$$

where:

T_A = Ambient temperature in °C

R_{θJA} = Package thermal resistance, junction-to-ambient in °C/W

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

P_{INT} = I_{DD} × V_{DD}, watts — chip internal power

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

For most applications P_{I/O} ≪ P_{INT} and can be neglected.

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

$$P_D = K \div (T_J + 273^\circ\text{C}) \tag{2}$$

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

$$K = P_D \times (T_A + 273^\circ\text{C}) + R_{\theta JA} \times P_D \tag{3}$$

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.

Table 9-3. DC Electrical Characteristics (V_{DD} = 5.0 Vdc)

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H, unless otherwise noted)

| Characteristic | Symbol | Min | Typ | Max | Unit |
|---|------------------------------------|----------------------------|--------|-----------------------|------|
| Output voltage (I _{Load} ≤ 10.0 μA) | V _{OL} V _{OH} | - V _{DD} - 0.1 | - - | 0.1 - | V |
| Output high voltage (I _{Load} = 0.8 mA) PA7-PA0, PB7-PB5, PC7-PC0, PD5, TCMF | V _{OH} | V _{DD} - 0.8 | - | - | V |
| Output low voltage (I _{Load} = 1.6 mA) PA7-PA0, PB7-PB5, PC7-PC0, PD5, TCMF | V _{OL} | - | - | 0.4 | V |
| Input high voltage PA7-PA0, PB7-PB5, PC7-PC0, PD5, PD7/TCAP, <u>IRQ</u> , <u>RESET</u> , OSC1 | V _{IH} | 0.7 × V _{DD} | - | V _{DD} | V |
| Input low voltage PA7-PA0, PB7-PB5, PC7-PC0, PD5, PD7/TCAP, <u>IRQ</u> , <u>RESET</u> , OSC1 | V _{IL} | V _{SS} | - | 0.2 × V _{DD} | V |
| Data-retention mode supply voltage (0 to 70°C) | V _{RM} | 2 | - | - | V |
| Supply current (See NOTES.) | I _{DD} | | | | |
| RUN | | - | TBD | TBD | mA |
| WAIT | | - | TBD | TBD | mA |
| STOP | | - | - | - | μA |
| 25°C | | - | TBD | TBD | μA |
| 0 to 70°C (standard) | | - | - | TBD | μA |
| I/O ports hi-Z leakage current PA7-PA0, PB7-PB5, PC7-PC0, PD5 | I _{IL} | - | - | ±10 | μA |
| Input current <u>RESET</u> , <u>IRQ</u> , OSC1, PD5, PD7/TCAP | I _{in} | - | - | ±1 | μA |
| Capacitance | | | | | |
| Ports (as input or output) | C _{out} | - | - | 12 | pF |
| <u>RESET</u> , <u>IRQ</u> , PD5, PD7/TCAP | C _{in} | - | - | 8 | pF |

NOTES:

1. Typical values at midpoint of voltage range, 25°C only.
2. RUN (operating) I_{DD}, WAIT I_{DD} measured using external square wave clock source (f_{osc} = 4.2 MHz), all inputs 0.2 V from rail; no dc loads, less than 50 pF on all outputs, C_L = 20 pF on OSC2.
3. WAIT I_{DD}, STOP I_{DD}: all ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD} - 0.2 V.
4. STOP I_{DD} measured with OSC1 = V_{SS}.
5. Standard temperature range is 0 to 70°C.
6. WAIT I_{DD} is affected linearly by the OSC2 capacitance.

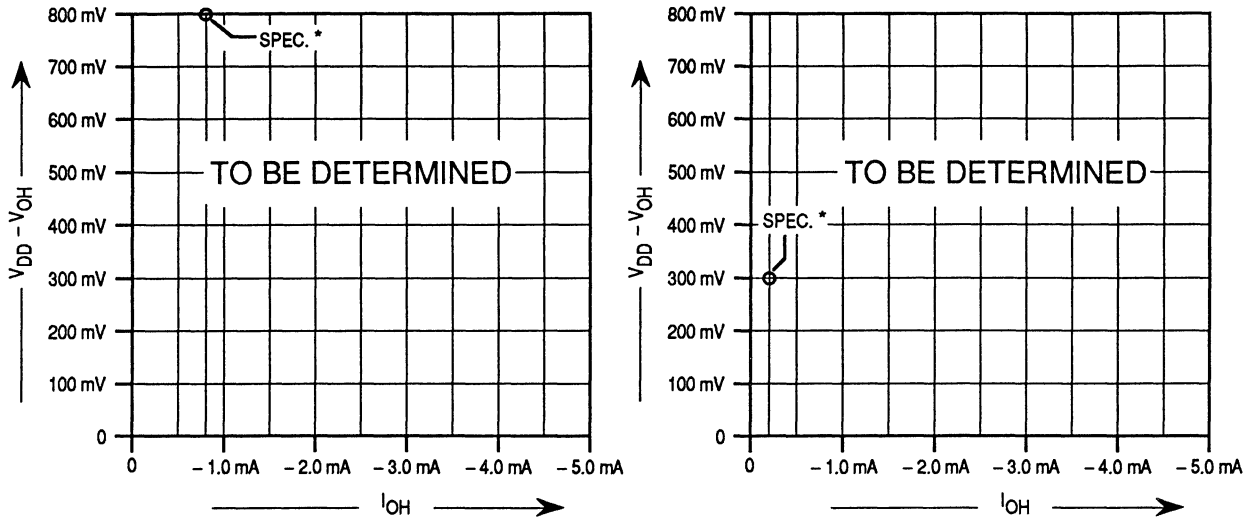
Table 9-4. DC Electrical Characteristics ($V_{DD} = 3.3 \text{ Vdc}$)

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

| Characteristic | Symbol | Min | Typ | 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-PB5, PC7-PC0, PD5, TCMP | V_{OH} | $V_{DD} - 0.3$ | - | - | V |
| Output low voltage ($I_{Load} = 0.4 \text{ mA}$) PA7-PA0, PB7-PB5, PC7-PC0, PD5, TCMP | V_{OL} | - | - | 0.3 | V |
| Input high voltage PA7-PA0, PB7-PB5, PC7-PC0, PD5, PD7/TCAP, \overline{IRQ} , \overline{RESET} , OSC1 | V_{IH} | $0.7 \times V_{DD}$ | - | V_{DD} | V |
| Input low voltage PA7-PA0, PB7-PB5, PC7-PC0, PD5, PD7/TCAP, \overline{IRQ} , \overline{RESET} , OSC1 | V_{IL} | V_{SS} | - | $0.2 \times V_{DD}$ | V |
| Data-retention mode supply voltage (0 to 70°C) | V_{RM} | 2.0 | - | - | V |
| Supply current (See NOTES.) | I_{DD} | | | | |
| RUN | | - | TBD | TBD | mA |
| WAIT | | - | TBD | TBD | mA |
| STOP | | - | TBD | TBD | μA |
| 25°C | | - | TBD | TBD | μA |
| 0 to 70°C (standard) | | - | - | TBD | μA |
| I/O ports hi-Z leakage current PA7-PA0, PB7-PB5, PC7-PC0, PD5 | I_{IL} | - | - | ± 10 | μA |
| Input current \overline{RESET} , \overline{IRQ} , OSC1, PD5, PD7/TCAP | I_{in} | - | - | ± 1 | μA |
| Capacitance | | | | | |
| Ports (as input or output) | C_{out} | - | - | 12 | pF |
| \overline{RESET} , \overline{IRQ} , PD5, PD7/TCAP | C_{in} | - | - | 8 | pF |

NOTES:

1. Typical values at midpoint of voltage range, 25°C only.
2. RUN (operating) I_{DD} , WAIT 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.
3. WAIT I_{DD} , STOP I_{DD} : all ports configured as inputs, $V_{IL} = 0.2 \text{ V}$, $V_{IH} = V_{DD} - 0.2 \text{ V}$.
4. STOP I_{DD} measured with OSC1 = V_{SS} .
5. Standard temperature range is 0 to 70°C.
6. WAIT I_{DD} is affected linearly by the OSC2 capacitance.

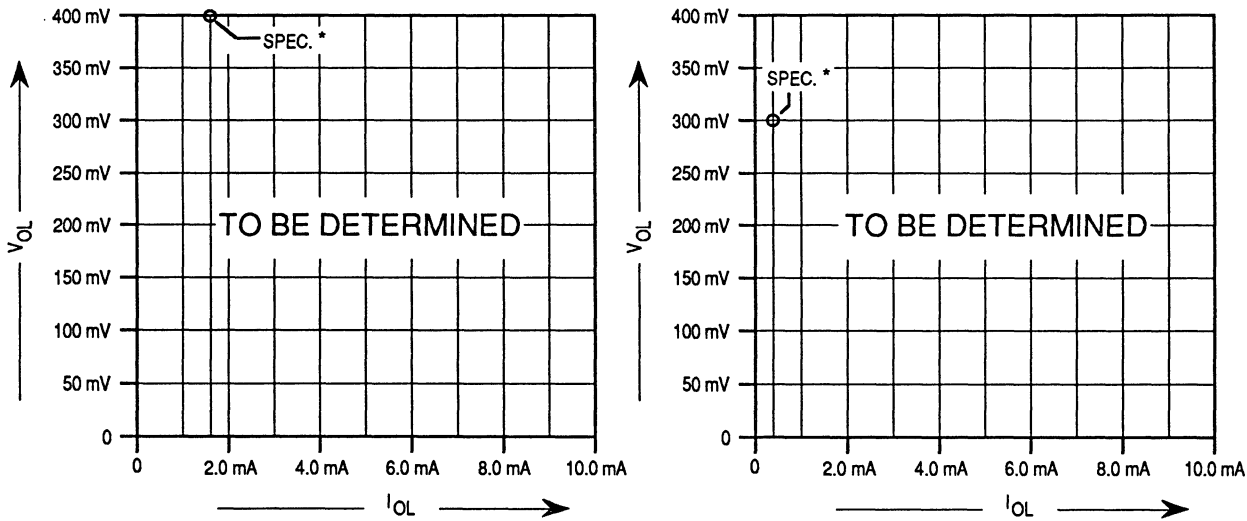


* At $V_{DD} = 5.0\text{ V}$, devices are specified and tested for $(V_{DD} - V_{OH}) \leq 800\text{ mV}$ @ $I_{OH} = -0.8\text{ mA}$.

* At $V_{DD} = 3.3\text{ V}$, devices are specified and tested for $(V_{DD} - V_{OH}) \leq 300\text{ mV}$ @ $I_{OH} = -0.2\text{ mA}$.

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.

Figure 9-2. Typical High-Side Driver Characteristics



* At $V_{DD} = 5.0\text{ V}$, devices are specified and tested for $V_{OL} \leq 400\text{ mV}$ @ $I_{OL} = 1.6\text{ mA}$.

* At $V_{DD} = 3.3\text{ V}$, devices are specified and tested for $V_{OL} \leq 300\text{ mV}$ @ $I_{OL} = 0.4\text{ mA}$.

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.

Figure. 9-3. Typical Low-Side Driver Characteristics

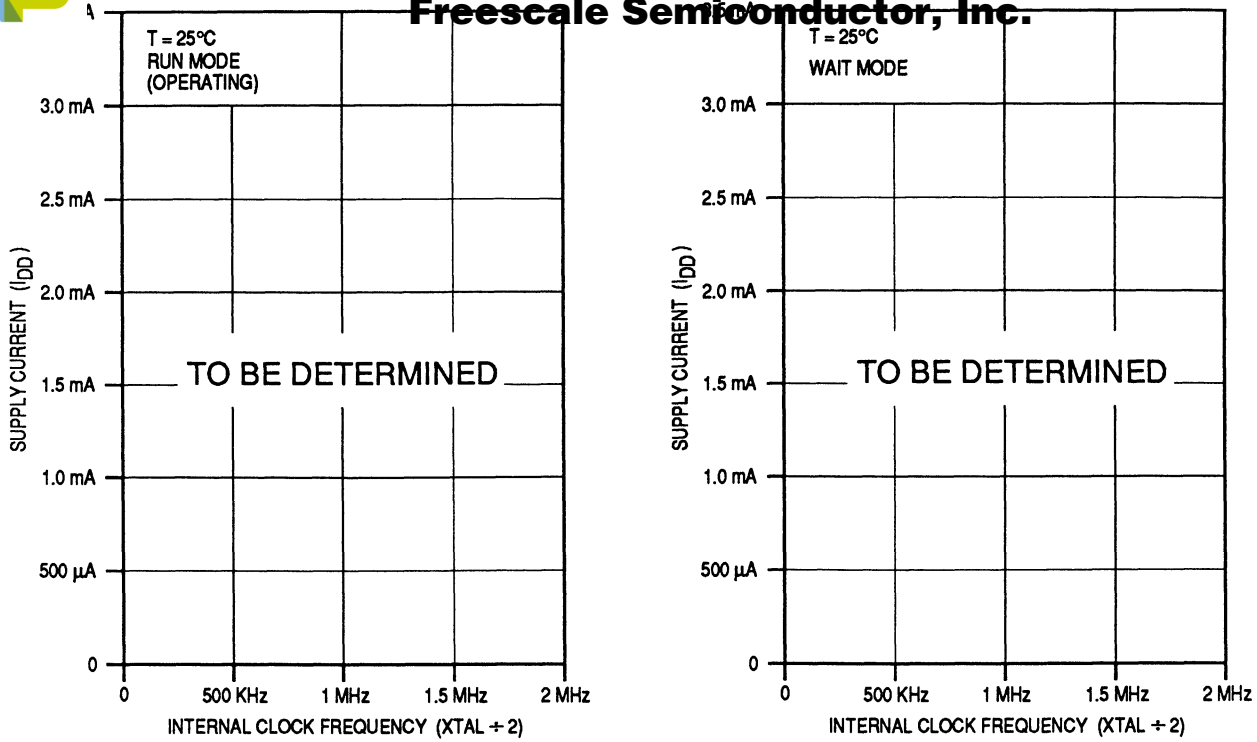


Figure 9-4. Typical Supply Current vs Internal Clock Frequency

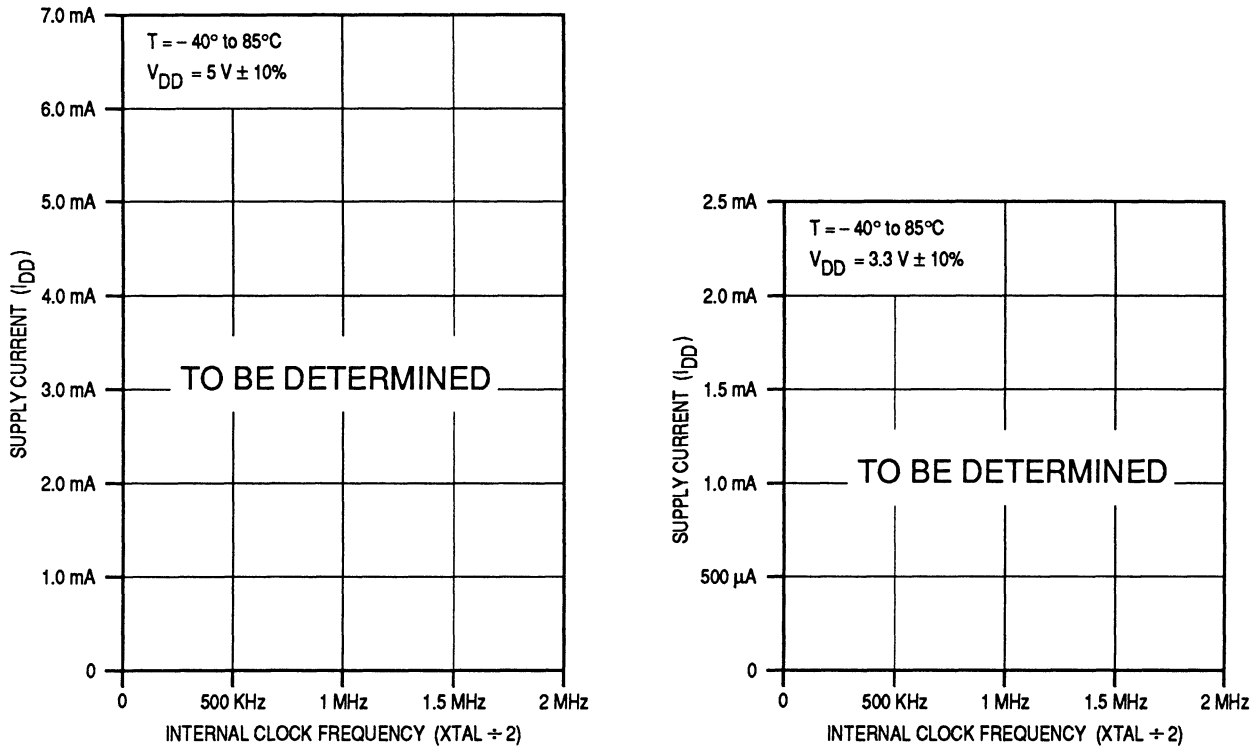


Figure 9-5. Maximum Supply Current vs Internal Clock Frequency

Table 9-5. Control Timing ($V_{DD} = 5.0 \text{ Vdc}$)

($V_{DD} = 5.0 \text{ Vdc} \pm 10\%$, $V_{SS} = 0 \text{ Vdc}$, $T_A = T_L$ to T_H)

| Characteristic | Symbol | Min | Max | Unit |
|--|------------------|---------------|-----|-----------|
| Oscillator frequency | f_{osc} | | | |
| Crystal option | | – | 4.2 | MHz |
| External clock option | | dc | 4.2 | MHz |
| Internal operating frequency | f_{op} | | | |
| Crystal ($f_{osc} + 2$) | | – | 2.1 | MHz |
| External clock ($f_{osc} + 2$) | | dc | 2.1 | MHz |
| Internal clock cycle time | t_{cyc} | 480 | – | ns |
| RESET pulse width | t_{RL} | 1.5 | – | t_{cyc} |
| Capture/compare timer | | | | |
| Resolution (See NOTE 1.) | t_{RESL} | 4.0 | – | t_{cyc} |
| Input capture pulse width | t_{TH}, t_{TL} | 125 | – | ns |
| Input capture pulse period | t_{TLTL} | (See NOTE 2.) | – | t_{cyc} |
| Interrupt pulse width low (edge-triggered) | t_{LILH} | 125 | – | ns |
| Interrupt pulse period | t_{LILL} | (See NOTE 3.) | – | t_{cyc} |
| OSC1 pulse width | t_{OH}, t_{OL} | 90 | – | ns |

NOTES:

1. Since a 2-bit prescaler in the capture/compare timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.
2. The minimum period t_{TLTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc} .
3. The minimum period t_{LILL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc} .

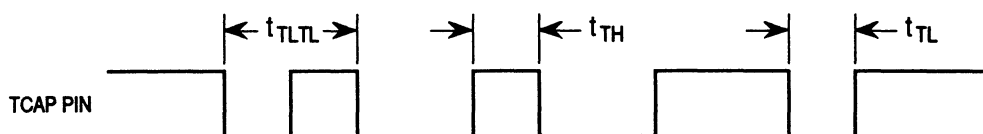
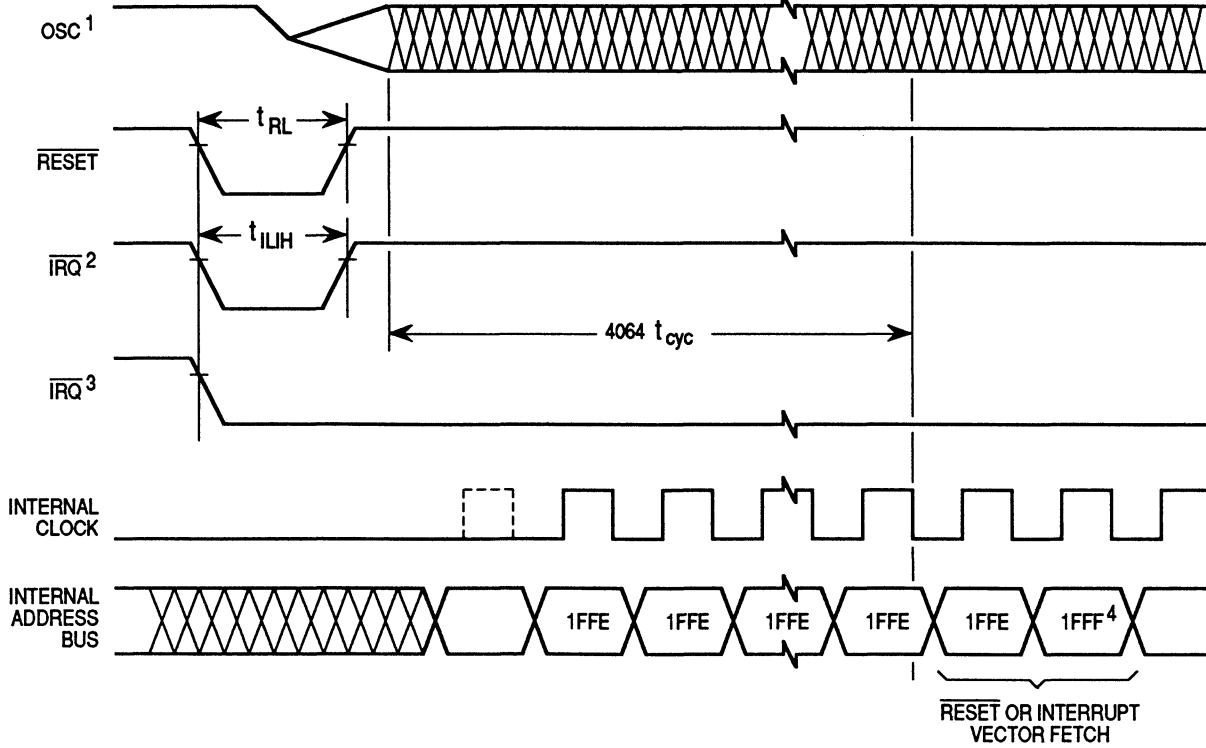


Figure 9-6. TCAP Timing



NOTES:

1. Represents the internal clocking of the OSC1 pin.
2. $\overline{\text{IRQ}}$ pin edge-sensitive mask option.
3. $\overline{\text{IRQ}}$ pin level- and edge-sensitive mask option.
4. $\overline{\text{RESET}}$ vector address shown for timing example.

Figure 9-7. STOP Recovery Timing

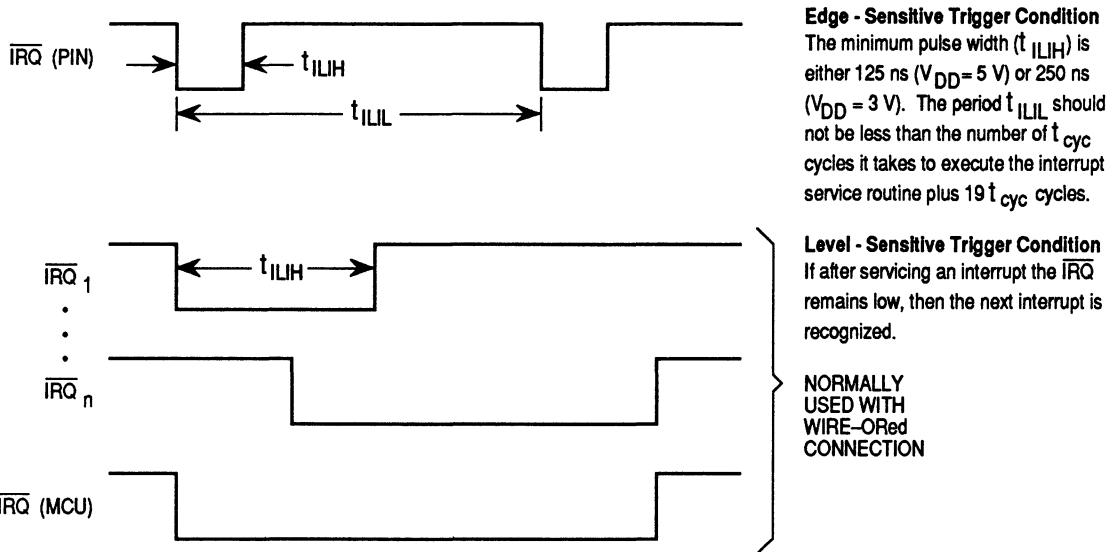


Figure 9-8. External Interrupt Timing

Table 9-6. Control Timing (V_{DD} = 3.3 Vdc)

(V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H.)

| Characteristic | Symbol | Min | Max | Unit |
|---|-----------------------------------|---------------|-----|------------------|
| Oscillator frequency | f _{osc} | | | |
| Crystal option | | – | 2.0 | MHz |
| External clock option | | dc | 2.0 | MHz |
| Internal operating frequency | f _{op} | | | |
| Crystal (f _{osc} + 2) | | – | 1.0 | MHz |
| External clock (f _{osc} + 2) | | dc | 1.0 | MHz |
| Cycle time | t _{cyc} | 1000 | – | ns |
| STOP recovery startup time (crystal oscillator) | t _{ILCH} | – | 100 | ms |
| RESET pulse width , excluding power-up | t _{RL} | 1.5 | – | t _{cyc} |
| Capture/compare timer | | | | |
| Resolution (See NOTE 1.) | t _{RESL} | 4.0 | – | t _{cyc} |
| Input capture pulse width | t _{TH} , t _{TL} | 250 | – | ns |
| Input capture pulse period | t _{TTL} | (See NOTE 2.) | – | t _{cyc} |
| Interrupt pulse width low (edge-triggered) | t _{ILIH} | 250 | – | ns |
| Interrupt pulse period | t _{ILIL} | (See NOTE 3.) | – | t _{cyc} |
| OSC1 pulse width | t _{OH} , t _{OL} | 200 | – | ns |

NOTES:

1. Since a 2-bit prescaler in the capture/compare timer must count four internal cycles (t_{cyc}), this is the limiting minimum factor in determining the timer resolution.
2. The minimum period t_{TTL} should not be less than the number of cycle times it takes to execute the capture interrupt service routine plus 24 t_{cyc}.
3. The minimum period t_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

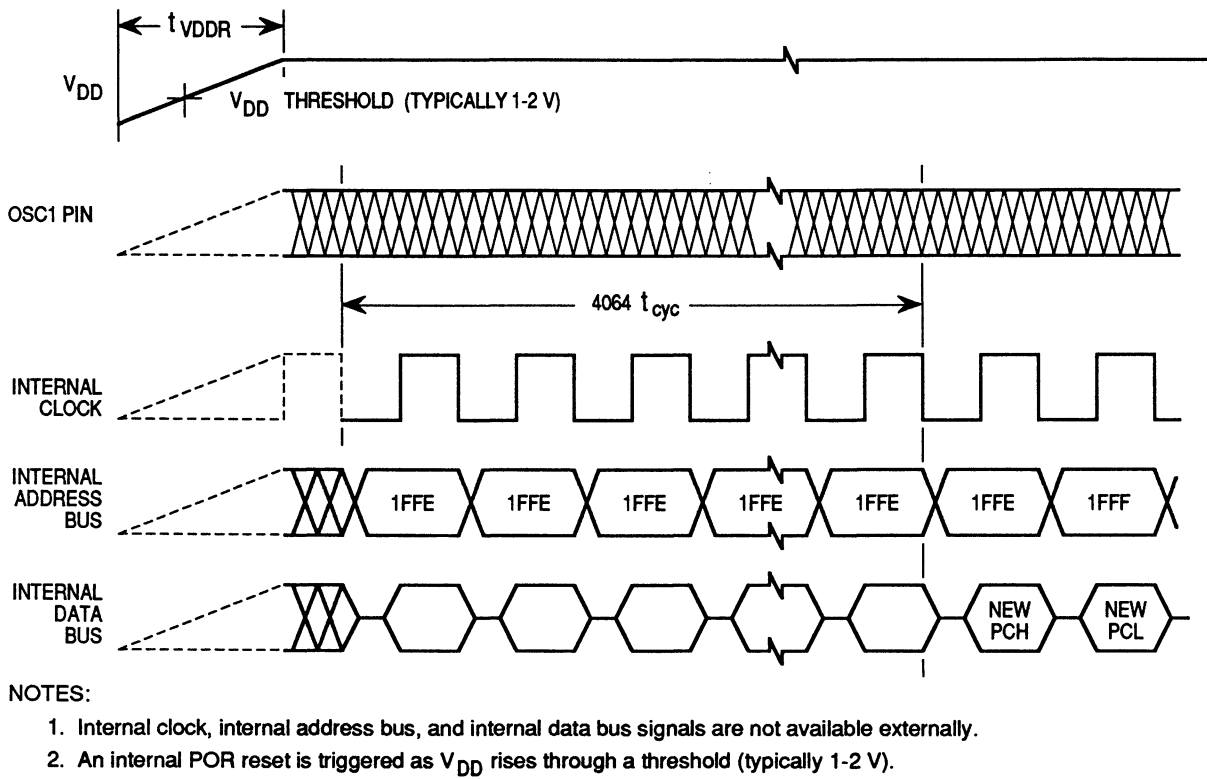


Figure 9-9. Power-On Reset Timing

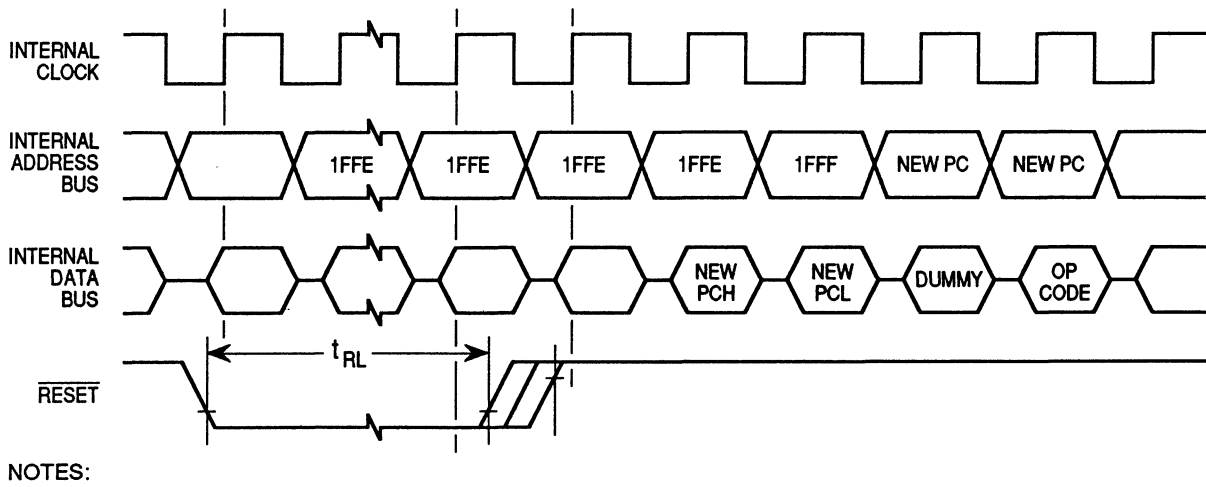


Figure 9-10. External Reset Timing

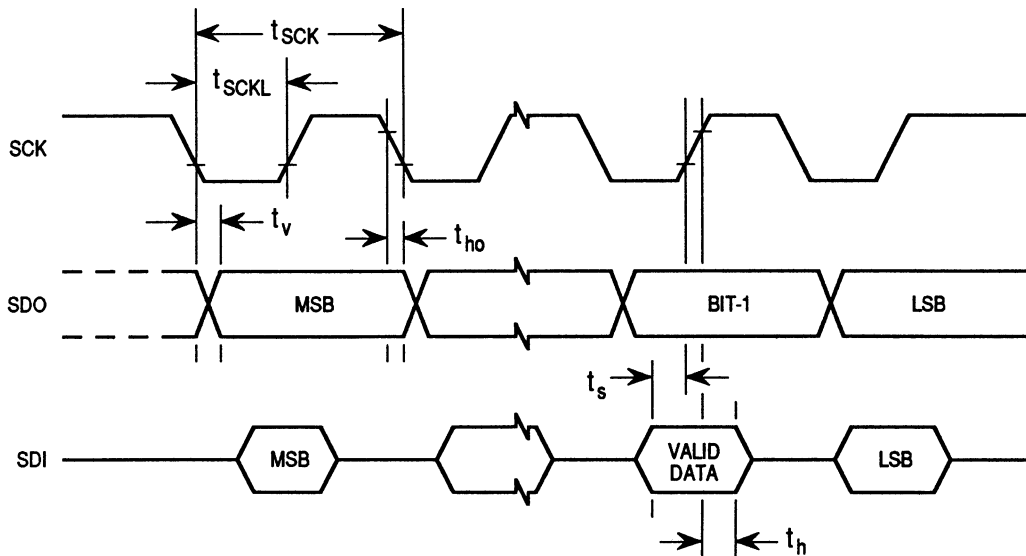
Table 9-7. SIOP Timing ($V_{DD} = 5.0$ Vdc)

($V_{DD} = 5.0$ Vdc \pm 10%, $V_{SS} = 0$ Vdc, $T_A = T_L$ to T_H , unless otherwise noted)

| Characteristic | Symbol | Min | Max | Unit |
|--|---------------|------|------|-----------|
| Frequency of operation | | | | |
| Master | $f_{SIOP(M)}$ | 0.25 | 0.25 | f_{op} |
| Slave | $f_{SIOP(S)}$ | dc | 0.25 | f_{op} |
| Cycle time | | | | |
| Master | $t_{SCK(M)}$ | 4.0 | 4.0 | t_{cyc} |
| Slave | $t_{SCK(S)}$ | – | 4.0 | t_{cyc} |
| Clock (SCK) low time ($f_{op} = 2.1$ MHz) | 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 |

NOTES:

- $f_{op} = f_{osc} + 2 = 2.1$ MHz maximum; $t_{cyc} = 1 + f_{op}$.
- In master mode, SCK is generated by dividing the internal clock (f_{op}) by 4.



NOTES:

- This diagram applies to both master and slave modes of the SIOP.
- Bit order is shown for MSB-first mask option.

Figure 9-11. SIOP Timing

Table 9-8. SIOP Timing (V_{DD} = 3.3 Vdc)

 (V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H, unless otherwise noted)

| Characteristic | Symbol | Min | Max | Unit |
|--|----------------------|------|------|------------------|
| Frequency of operation | | | | |
| Master | f _{SIOP(M)} | 0.25 | 0.25 | f _{op} |
| Slave | f _{SIOP(S)} | dc | 0.25 | f _{op} |
| Cycle time | | | | |
| Master | t _{SCK(M)} | 4.0 | 4.0 | t _{cyc} |
| Slave | t _{SCK(S)} | – | 4.0 | t _{cyc} |
| Clock (SCK) low time (f _{op} = 1.0 MHz) | 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 |

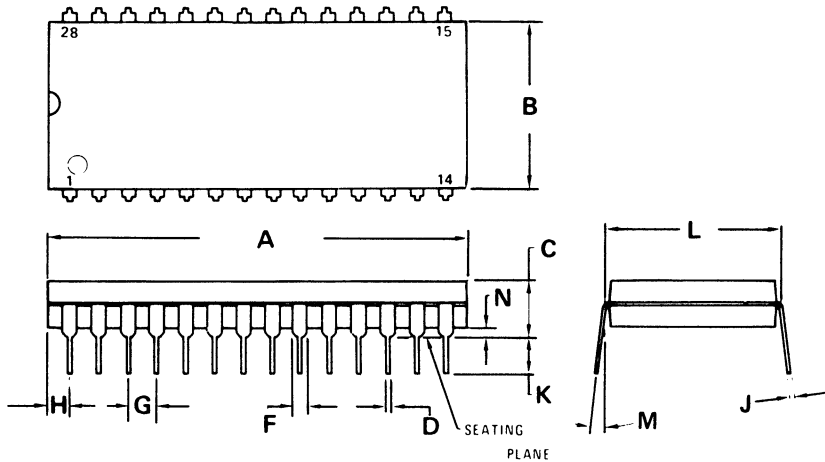
 NOTE: f_{op} = 1.0 MHz maximum

SECTION 10 MECHANICAL SPECIFICATIONS

This section describes the dimensions of the dual in-line package (DIP) and small outline integrated circuit (SOIC) MCU packages.

10.1 Dual In-Line Package (DIP)

P SUFFIX
PLASTIC PACKAGE
CASE 710-02



| 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 |

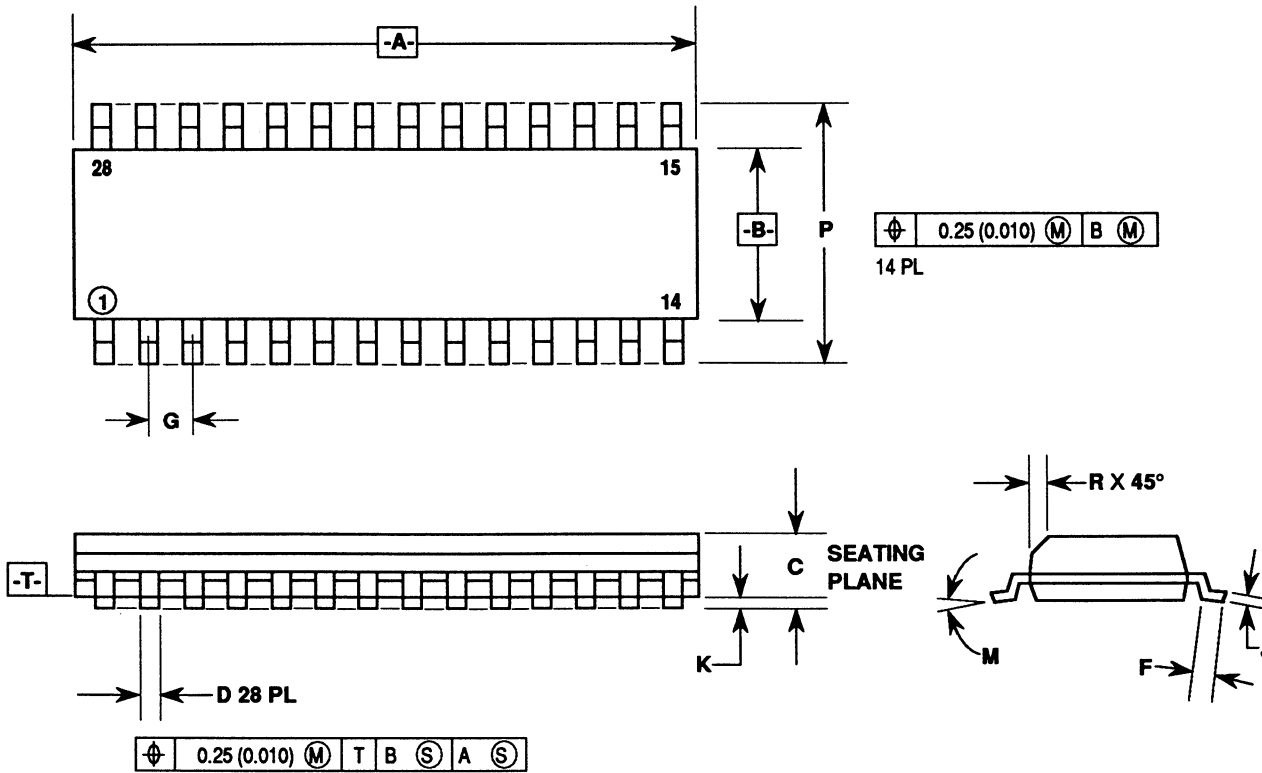
NOTES:

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.

Figure 10-1. Case 710-02 Dimensions

10.2 Small Outline Integrated Circuit (SOIC)

DW SUFFIX
CASE 751F-02



NOTES:

1. DIMENSIONS A AND B ARE DATUMS AND T IS A DATUM SURFACE.
2. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
3. CONTROLLING DIMENSION: MILLIMETER.
4. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION.
5. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.

| DIM | MILLIMETERS | | INCHES | |
|-----|-------------|-------|-----------|-------|
| | MIN | MAX | MIN | MAX |
| A | 17.80 | 18.05 | 0.701 | 0.710 |
| B | 7.40 | 7.60 | 0.292 | 0.299 |
| C | 2.35 | 2.65 | 0.093 | 0.104 |
| D | 0.35 | 0.49 | 0.014 | 0.019 |
| F | 0.50 | 0.90 | 0.020 | 0.035 |
| G | 1.27 BSC | | 0.050 BSC | |
| J | 0.25 | 0.32 | 0.010 | 0.012 |
| K | 0.10 | 0.25 | 0.004 | 0.009 |
| M | 0° | 7° | 0° | 7° |
| P | 10.05 | 10.55 | 0.395 | 0.415 |
| R | 0.25 | 0.75 | 0.010 | 0.029 |

Figure 10-2. Case 751F-02 Dimensions

SECTION 11

ORDERING INFORMATION

This section describes the information needed to order the MCU ROM pattern media.

11.1 ROM Pattern Media

Ordering information can be delivered to Motorola in the following media:

1. MSTM-DOS¹ or PC-DOS flexible disk (360K)
2. EPROM(s) 2764, MCM68764, MCM68766

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field service office, a sales person, or a Motorola representative.

11.1.1 Flexible Disks

A flexible disk containing the customer's program (using positive logic for address and data), may be submitted for pattern generation. The disk should be clearly labeled with the customer's name, data, project or product name, and the name of the file containing the pattern.

In addition to the program pattern, a file containing the program source code listing can be included. This data is kept confidential and used to expedite the process in case of any difficulty with the pattern file.

MS-DOS is the Microsoft Disk Operating System. PC-DOS is the IBM^{®2} Personal Computer (PC) Disk Operating System. Submitted disks must be standard density (360K) double-sided 5-1/4 in. disks. The disks must contain object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by M6805 cross assemblers and linkers on IBM PC-style machines.

¹MS-DOS is a trademark of Microsoft, Inc.

²IBM is a registered trademark of International Business Machines Corporation.

11.1.2 EPROMs

A type 2764, 68764, or 68766 EPROM containing the customer's program (using positive logic for address and data), may be submitted for pattern generation. User ROM is programmed at EPROM addresses \$0020 through \$004F (page zero) and \$0100 through \$10FF with vectors at addresses \$1FF0 to \$1FFF. All unused bytes, including those in the user's space, must be set to zero. For shipment to Motorola, EPROMs should be packed securely in a conductive IC carrier. Styrofoam is not acceptable for shipment.

11.2 ROM Pattern Verification

11.2.1 Verification Media

All original pattern media are filed for contractual purposes and are not returned. A computer listing of the ROM code is generated and returned along with a listing verification form. The listing should be thoroughly checked, and the verification form should be completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for the creation of the customer mask. To aid in the verification process, Motorola programs the *customer-supplied* blank EPROMs or DOS disks from the data file used to create the custom mask.

11.2.2 ROM Verification Units (RVUs)

Ten RVUs containing the customer's ROM pattern are sent for program verification. These units are made using the custom mask, but are for the purpose of ROM verification only. For expediency, the RVUs are unmarked, packaged in ceramic, and tested with 5 V at room temperature. These RVUs are free of charge with the minimum order quantity, but are not production parts. RVUs are not backed or guaranteed by Motorola Quality Assurance.

11.3 MC Order Numbers

Table 11-1 provides ordering information for available package types.

Table 11-1. MC Order Numbers

| Package Type | MC Order Number |
|--------------|-----------------|
| Plastic DIP | MC68HC05P4P |
| SOIC | MC68HC05P4DW |



Freescale Semiconductor, Inc.

MC68HC05P4 MCU ORDERING FORM

Date _____ Customer PO Number _____
 Customer Company _____
 Address _____
 City _____ State _____ Zip _____
 Country _____
 Phone _____ Extension _____
 Customer Contact Person _____
 Customer Part Number (if applicable - 12 characters maximum) _____
 Application _____

Freescale Semiconductor, Inc.

SIOP Data Format:

- MSB First
 LSB First

Internal Oscillator Input:

- Crystal/Resonator
 Resistor

Temperature Range:

- 0 to 70°C (Standard)
 -40 to +85°C

Special Electrical Provisions:

 (Customer specifications required)

Pattern Media:

- MS-DOS Disk File 2764 EPROM MCM68764 EPROM
 PC-DOS Disk File MCM68766 EPROM
 Other _____

(Requires prior factory approval)

Device Marking:

- Motorola Standard
 Motorola Logo
 Motorola Part Number
 Mask and Datecode
- Standard with Customer Part Number
 Motorola Logo
 Motorola Part Number
 Customer Part Number
 Mask and Datecode

- Other _____

Device marking other than the two standard forms requires prior factory approval.

 (SIGNATURE)

Device to be tested to Motorola data sheet specifications. Customer part number, if used as part of marking, is for reference purposes only.

 (SIGNATURE)

Device to be tested to customer specifications. (Customer specifications required)

ONLY ONE SIGNATURE IS REQUIRED TO PROCESS THIS ORDERING FORM.





Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**

Home Page:

www.freescale.com

email:

support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor
 Technical Information Center, CH370
 1300 N. Alma School Road
 Chandler, Arizona 85224
 (800) 521-6274
 480-768-2130

support@freescale.com

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
 Technical Information Center
 Schatzbogen 7
 81829 Muenchen, Germany
 +44 1296 380 456 (English)
 +46 8 52200080 (English)
 +49 89 92103 559 (German)
 +33 1 69 35 48 48 (French)
support@freescale.com

Japan:

Freescale Semiconductor Japan Ltd.
 Headquarters
 ARCO Tower 15F
 1-8-1, Shimo-Meguro, Meguro-ku
 Tokyo 153-0064, Japan
 0120 191014
 +81 2666 8080

support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.
 Technical Information Center
 2 Dai King Street
 Tai Po Industrial Estate,
 Tai Po, N.T., Hong Kong
 +800 2666 8080

support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor
 Literature Distribution Center
 P.O. Box 5405
 Denver, Colorado 80217
 (800) 441-2447
 303-675-2140
 Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

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

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

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

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



Freescale Semiconductor, Inc.

