

MC68HC11ED0

Technical Summary

HCMOS

Microcontroller Unit

MC68HC11ED0

Technical Summary

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

List of Sections

| | |
|--|-----------|
| Section 1. General Description | 15 |
| Section 2. Pin Assignments | 19 |
| Section 3. Central Processor Unit (CPU) | 23 |
| Section 4. Operating Modes and On-Chip Memory | 43 |
| Section 5. Resets and Interrupts | 57 |
| Section 6. Parallel Input/Output (I/O) Ports | 65 |
| Section 7. Serial Communications Interface (SCI). | 71 |
| Section 8. Serial Peripheral Interface (SPI). | 81 |
| Section 9. Timing System. | 87 |

List of Sections

Table of Contents

Section 1. General Description

| | | |
|-----|--------------------------------|----|
| 1.1 | Contents | 15 |
| 1.2 | Introduction | 15 |
| 1.3 | Features | 16 |
| 1.4 | Structure | 17 |
| 1.5 | Ordering Information | 18 |

Section 2. Pin Assignments

| | | |
|-----|---|----|
| 2.1 | Contents | 19 |
| 2.2 | Introduction | 19 |
| 2.3 | 44-Pin Plastic-Leaded Chip Carrier (PLCC) | 20 |
| 2.4 | 44-Pin Quad Flat Pack (QFP) | 21 |
| 2.5 | 40-Pin Plastic Dual In-Line Package (DIP) | 22 |

Section 3. Central Processor Unit (CPU)

| | | |
|-------|------------------------------------|----|
| 3.1 | Contents | 23 |
| 3.2 | Introduction | 24 |
| 3.3 | CPU Registers | 24 |
| 3.3.1 | Accumulators A, B, and D | 25 |
| 3.3.2 | Index Register X (IX) | 26 |
| 3.3.3 | Index Register Y (IY) | 26 |
| 3.3.4 | Stack Pointer (SP) | 26 |
| 3.3.5 | Program Counter (PC) | 28 |

Table of Contents

| | | |
|---------|---|----|
| 3.3.6 | Condition Code Register (CCR) | 29 |
| 3.3.6.1 | Carry/Borrow (C) | 29 |
| 3.3.6.2 | Overflow (V) | 29 |
| 3.3.6.3 | Zero (Z) | 29 |
| 3.3.6.4 | Negative (N) | 30 |
| 3.3.6.5 | I-Interrupt Mask (I) | 30 |
| 3.3.6.6 | Half Carry (H) | 30 |
| 3.3.6.7 | X-Interrupt Mask (X) | 30 |
| 3.3.6.8 | STOP Disable (S) | 31 |
| 3.4 | Data Types | 31 |
| 3.5 | Opcodes and Operands | 31 |
| 3.6 | Addressing Modes | 32 |
| 3.6.1 | Immediate | 32 |
| 3.6.2 | Direct | 33 |
| 3.6.3 | Extended | 33 |
| 3.6.4 | Indexed | 33 |
| 3.6.5 | Inherent | 33 |
| 3.6.6 | Relative | 34 |
| 3.7 | Instruction Set | 34 |

Section 4. Operating Modes and On-Chip Memory

| | | |
|-------|---|----|
| 4.1 | Contents | 43 |
| 4.2 | Introduction | 43 |
| 4.3 | Operating Modes | 44 |
| 4.3.1 | Bootstrap Mode | 44 |
| 4.3.2 | Special Test Mode | 44 |
| 4.3.3 | Expanded Operating Mode | 45 |
| 4.4 | Mode Selection | 46 |
| 4.5 | On-Chip Memory | 48 |
| 4.5.1 | Memory Map and Register Block | 49 |
| 4.5.2 | RAM | 56 |

Section 5. Resets and Interrupts

| | | |
|-----|---|----|
| 5.1 | Contents | 57 |
| 5.2 | Introduction | 57 |
| 5.3 | Resets | 57 |
| 5.4 | System Configuration Options Register | 60 |
| 5.5 | Arm/Reset COP Timer Circuitry Register | 61 |
| 5.6 | Configuration Control Register | 62 |
| 5.7 | Highest Priority I-Bit Interrupt and Miscellaneous Register . . . | 63 |

Section 6. Parallel Input/Output (I/O) Ports

| | | |
|-----|--|----|
| 6.1 | Contents | 65 |
| 6.2 | Introduction | 65 |
| 6.3 | Port A Data Register | 66 |
| 6.4 | Pulse Accumulator Control Register | 67 |
| 6.5 | Port D Data Register | 68 |
| 6.6 | Port D Data Direction Register | 69 |

Section 7. Serial Communications Interface (SCI)

| | | |
|-------|--|----|
| 7.1 | Contents | 71 |
| 7.2 | Introduction | 71 |
| 7.3 | SCI Registers | 74 |
| 7.3.1 | Baud Rate Register | 74 |
| 7.3.2 | Serial Communications Control Register 1 | 77 |
| 7.3.3 | Serial Communications Control Register 2 | 78 |
| 7.3.4 | Serial Communication Status Register | 79 |
| 7.3.5 | Serial Communications Data Register | 80 |

Section 8. Serial Peripheral Interface (SPI)

| | | |
|-------|---|----|
| 8.1 | Contents | 81 |
| 8.2 | Introduction | 81 |
| 8.3 | SPI Registers | 83 |
| 8.3.1 | Serial Peripheral Control Register | 83 |
| 8.3.2 | Serial Peripheral Status Register | 85 |
| 8.3.3 | Serial Peripheral Data I/O Register | 86 |

Section 9. Timing System

| | | |
|--------|---|-----|
| 9.1 | Contents | 87 |
| 9.2 | Introduction | 88 |
| 9.3 | Timer Registers | 91 |
| 9.3.1 | Timer Compare Force Register | 91 |
| 9.3.2 | Output Compare 1 Mask Register | 92 |
| 9.3.3 | Output Compare 1 Data Register | 92 |
| 9.3.4 | Timer Count Register | 93 |
| 9.3.5 | Timer Input Capture Registers | 94 |
| 9.3.6 | Timer Output Compare Registers | 95 |
| 9.3.7 | Timer Input Capture 4/Output Compare 5 Register | 96 |
| 9.3.8 | Timer Control Register 1 | 97 |
| 9.3.9 | Timer Control Register 2 | 98 |
| 9.3.10 | Timer Interrupt Mask 1 Register | 99 |
| 9.3.11 | Timer Interrupt Flag 1 Register | 100 |
| 9.3.12 | Timer Interrupt Mask 2 Register | 100 |
| 9.3.13 | Timer Interrupt Flag Register 2 | 102 |
| 9.4 | Pulse Accumulator | 103 |
| 9.4.1 | Pulse Accumulator Control Register | 104 |
| 9.4.2 | Pulse Accumulator Counter Register | 106 |

List of Figures

| Figure | Title | Page |
|--------|--|------|
| 1-1 | MC68HC11ED0 Block Diagram | 17 |
| 2-1 | Pin Assignments for 44-Pin PLCC | 20 |
| 2-2 | Pin Assignments for 44-Pin QFP | 21 |
| 2-3 | Pin Assignments for 40-Pin DIP | 22 |
| 3-1 | Programming Model | 25 |
| 3-2 | Stacking Operations | 27 |
| 4-1 | Address and Data Demultiplexing | 46 |
| 4-2 | Highest Priority I-Bit Interrupt and Miscellaneous Register (HPRIO) | 47 |
| 4-3 | MC68HC11ED0 Memory Map | 49 |
| 4-4 | Register and Control Bit Assignments | 50 |
| 4-5 | RAM and Register Mapping Register (INIT) | 56 |
| 5-1 | System Configuration Options Register (OPTION) | 60 |
| 5-2 | Arm/Reset COP Timer Circuitry Register (COPRST) | 61 |
| 5-3 | System Configuration Register (CONFIG) | 62 |
| 5-4 | Highest Priority I-Bit Interrupt and Miscellaneous Register (HPRIO) | 63 |
| 6-1 | Port A Data Register (PORTA) | 66 |
| 6-2 | Pulse Accumulator Control Register (PACTL) | 67 |
| 6-3 | Port D Data Register (PORTD) | 68 |
| 6-4 | Port D Data Direction Register (DDRD) | 69 |
| 7-1 | SCI Transmitter Block Diagram | 72 |
| 7-2 | SCI Receiver Block Diagram | 73 |

List of Figures

| Figure | Title | Page |
|--------|---|------|
| 7-3 | Baud Rate Register (BAUD) | 74 |
| 7-4 | SCI Baud Rate Generator Clock Diagram | 76 |
| 7-5 | Serial Communications Control Register 1 (SCCR1) | 77 |
| 7-6 | Serial Communications Control Register 2 (SCCR2) | 78 |
| 7-7 | Serial Communications Status Register (SCSR) | 79 |
| 7-8 | Serial Communications Data Register (SCDR) | 80 |
| 8-1 | SPI Block Diagram | 82 |
| 8-2 | Serial Peripheral Control Register (SPCR) | 83 |
| 8-3 | SPI Transfer Format | 84 |
| 8-4 | Serial Peripheral Status Register (SPSR) | 85 |
| 8-5 | Serial Peripheral Data I/O Register (SPDR) | 86 |
| 9-1 | Timer Block Diagram | 90 |
| 9-2 | Timer Compare Force Register (CFORC) | 91 |
| 9-3 | Output Compare 1 Mask Register (OC1M) | 92 |
| 9-4 | Output Compare 1 Data Register (OC1D) | 92 |
| 9-5 | Timer Count Register (TCNT) | 93 |
| 9-6 | Timer Input Capture Register 1 (TIC1) | 94 |
| 9-7 | Timer Input Capture Register 2 (TIC2) | 94 |
| 9-8 | Timer Input Capture Register 3 (TIC3) | 94 |
| 9-9 | Timer Output Compare Register 1 (TOC1) | 95 |
| 9-10 | Timer Output Compare Register 2 (TOC2) | 95 |
| 9-11 | Timer Output Compare Register 3 (TOC3) | 95 |
| 9-12 | Timer Output Compare Register 4 (TOC4) | 96 |
| 9-13 | Timer Input Capture4/Output Compare 5 Register (TI4/O5) | 96 |
| 9-14 | Timer Control Register 1 (TCTL1) | 97 |
| 9-15 | Timer Control Register 2 (TCTL2) | 98 |
| 9-16 | Timer Interrupt Mask 1 Register (TMSK1) | 99 |
| 9-17 | Timer Interrupt Flag 1 Register (TFLG1) | 100 |
| 9-18 | Timer Interrupt Mask 2 Register (TMSK2) | 100 |
| 9-19 | Timer Interrupt Flag 2 Register (TFLG2) | 102 |
| 9-20 | Pulse Accumulator System Block Diagram | 103 |
| 9-21 | Pulse Accumulator Control Register (PACTL) | 104 |
| 9-22 | Pulse Accumulator Counter Register (PACNT) | 106 |

List of Tables

| Table | Title | Page |
|-------|--|------|
| 1-1 | Device Ordering Information | 18 |
| 3-1 | Reset Vector Comparison | 28 |
| 3-2 | Instruction Set | 35 |
| 5-1 | Interrupt and Reset Vector Assignments | 59 |
| 5-2 | COP Timer Rate Select | 61 |
| 5-3 | Highest Priority Interrupt Selection | 64 |
| 6-1 | Input/Output Ports | 65 |
| 7-1 | Prescaler Rates | 74 |
| 7-2 | Baud Rates | 75 |
| 8-1 | SPI Clock Rate Selects | 84 |
| 9-1 | Timer Summary | 89 |
| 9-2 | Timer Output Compare Actions | 97 |
| 9-3 | Timer Control Configuration | 98 |
| 9-4 | Timer Prescale | 101 |
| 9-5 | Pulse Accumulator Timing | 104 |
| 9-6 | RTI Rates (Period Length) | 105 |
| 9-7 | RTI Rates (Frequency) | 105 |

Section 1. General Description

1.1 Contents

| | | |
|-----|--------------------------------|----|
| 1.2 | Introduction | 15 |
| 1.3 | Features | 16 |
| 1.4 | Structure | 17 |
| 1.5 | Ordering Information | 18 |

1.2 Introduction

The MC68HC11ED0 is a low-cost member of the M68HC11 Family of microcontrollers (MCU). This MCU has a multiplexed address/data bus and is characterized by high speed and low-power consumption. The fully static design allows operation at frequencies from 3 MHz to dc.

Pin count is minimized for cost-sensitive applications. Because there is no on-chip read-only memory (ROM), this device is optimized for expanded-bus systems. On-chip serial peripheral interface (SPI) and serial communications interface (SCI) provide a convenient means of transferring data to and from internal random-access memory (RAM). Refer to [Figure 1-1](#).

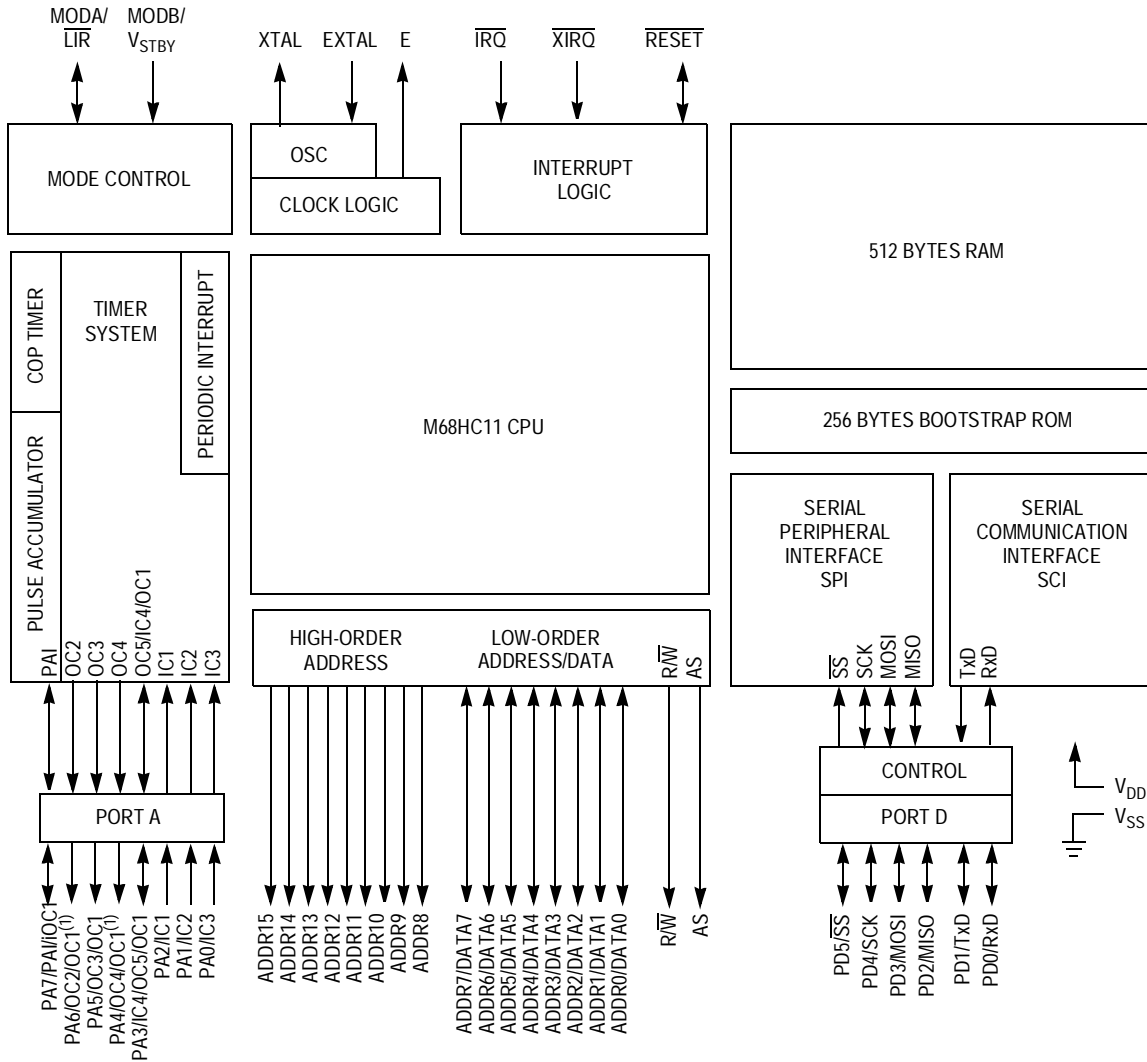
1.3 Features

Features include:

- M68HC11 CPU
- Power-saving stop and wait modes
- 512 bytes of RAM
- Multiplexed address and data buses
- Enhanced 16-bit timer with 4-stage programmable prescaler
 - Three input capture (IC) channels
 - Four output compare (OC) channels
 - One additional channel, selectable as fourth IC or fifth OC
- 8-bit pulse accumulator
- Real-time interrupt circuit
- Computer operating properly (COP) watchdog
- Clock monitor
- Enhanced asynchronous non-return-to-zero (NRZ) SCI
- Enhanced SPI
- Eight bidirectional input/output (I/O) lines
- Three input-only lines
- Three output-only lines (one output-only line in 40-pin package)
- Packaging options:
 - 44-pin plastic-leaded chip carrier (PLCC)
 - 44-pin quad flat pack (QFP)
 - 40-pin plastic dual in-line package (DIP)

1.4 Structure

See [Figure 1-1](#) for a block diagram of the MC68HC11ED0 MCU.



Note 1. Not bonded in 40-pin package

Figure 1-1. MC68HC11ED0 Block Diagram

1.5 Ordering Information

Table 1-1 provides ordering information for the MC68HC11ED0. Refer to **Section 2. Pin Assignments**.

Table 1-1. Device Ordering Information

| Package | Temperature | Description | Frequency | MC Order Number |
|-------------|-----------------|--|-----------|-----------------|
| 44-pin PLCC | -40°C to +85°C | No ROM/EPROM No EEPROM 512 bytes RAM | 2 MHz | MC68HC11ED0CFN2 |
| | | | 3 MHz | MC68HC11ED0CFN3 |
| | -40°C to +105°C | | 2 MHz | MC68HC11ED0VFN2 |
| | | | 3 MHz | MC68HC11ED0VFN3 |
| | -40°C to +125°C | | 2 MHz | MC68HC11ED0MFN2 |
| | | | 3 MHz | MC68HC11ED0MFN3 |
| 44-pin QFP | -40°C to +85°C | No ROM/EPROM No EEPROM 512 bytes RAM | 2 MHz | MC68HC11ED0CFU2 |
| | | | 3 MHz | MC68HC11ED0CFU3 |
| | -40°C to +105°C | | 2 MHz | MC68HC11ED0VFU2 |
| | | | 3 MHz | MC68HC11ED0VFU3 |
| | -40°C to +125°C | | 2 MHz | MC68HC11ED0MFU2 |
| | | | 3 MHz | MC68HC11ED0MFU3 |
| 44-pin DIP | -40°C to +85°C | No ROM/EPROM No EEPROM 512 bytes RAM | 2 MHz | MC68HC11ED0CP2 |
| | | | 3 MHz | MC68HC11ED0CP3 |
| | -40°C to +105°C | | 2 MHz | MC68HC11ED0VP2 |
| | | | 3 MHz | MC68HC11ED0VP3 |
| | -40°C to +125°C | | 2 MHz | MC68HC11ED0MP2 |
| | | | 3 MHz | MC68HC11ED0MP3 |

Section 2. Pin Assignments

2.1 Contents

| | | |
|-----|---|----|
| 2.2 | Introduction | 19 |
| 2.3 | 44-Pin Plastic-Leaded Chip Carrier (PLCC) | 20 |
| 2.4 | 44-Pin Quad Flat Pack (QFP) | 21 |
| 2.5 | 40-Pin Plastic Dual In-Line Package (DIP) | 22 |

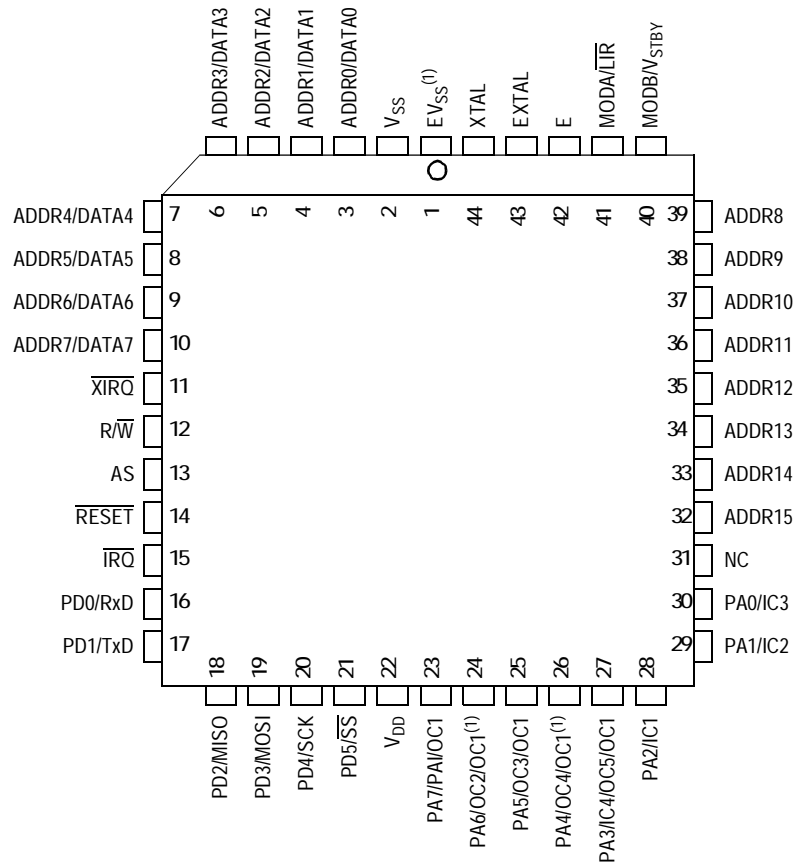
2.2 Introduction

The MC68HC11ED0 pin assignments are shown here for these packages:

- 44-pin plastic-leaded chip carrier (PLCC)
- 44-pin quad flat pack (QFP)
- 40-pin plastic dual in-line package (DIP)

2.3 44-Pin Plastic-Leaded Chip Carrier (PLCC)

Refer to [Figure 2-1](#) for the 44-pin PLCC pin assignments.

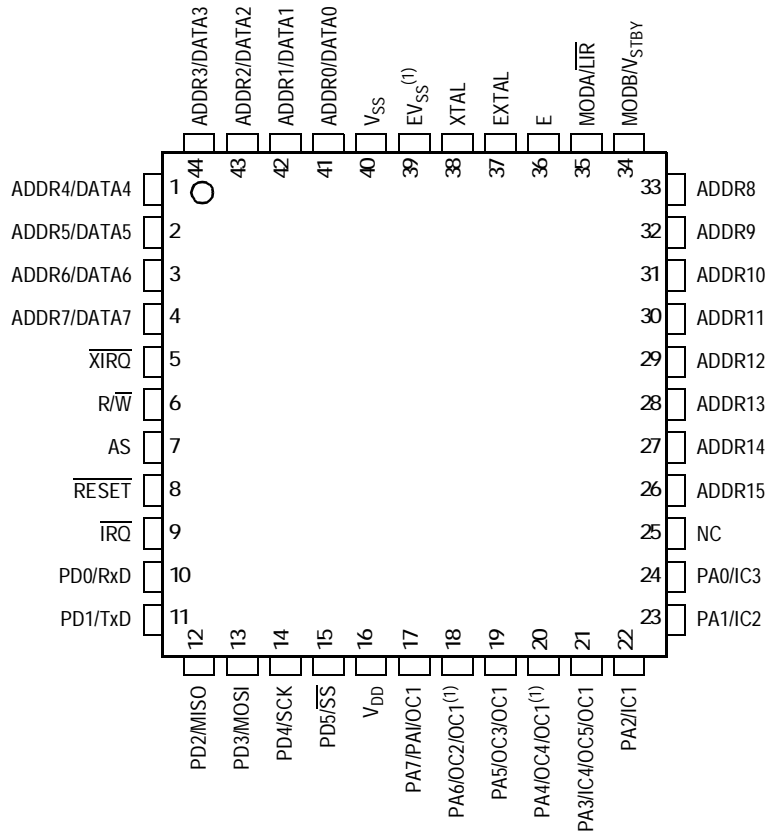


Note 1. Not bonded in 40-pin package

Figure 2-1. Pin Assignments for 44-Pin PLCC

2.4 44-Pin Quad Flat Pack (QFP)

Refer to [Figure 2-2](#) for the 44-pin QFP pin assignments.



Note 1. Not bonded in 40-pin package

Figure 2-2. Pin Assignments for 44-Pin QFP

2.5 40-Pin Plastic Dual In-Line Package (DIP)

Refer to [Figure 2-3](#) for the 40-pin DIP pin assignments.

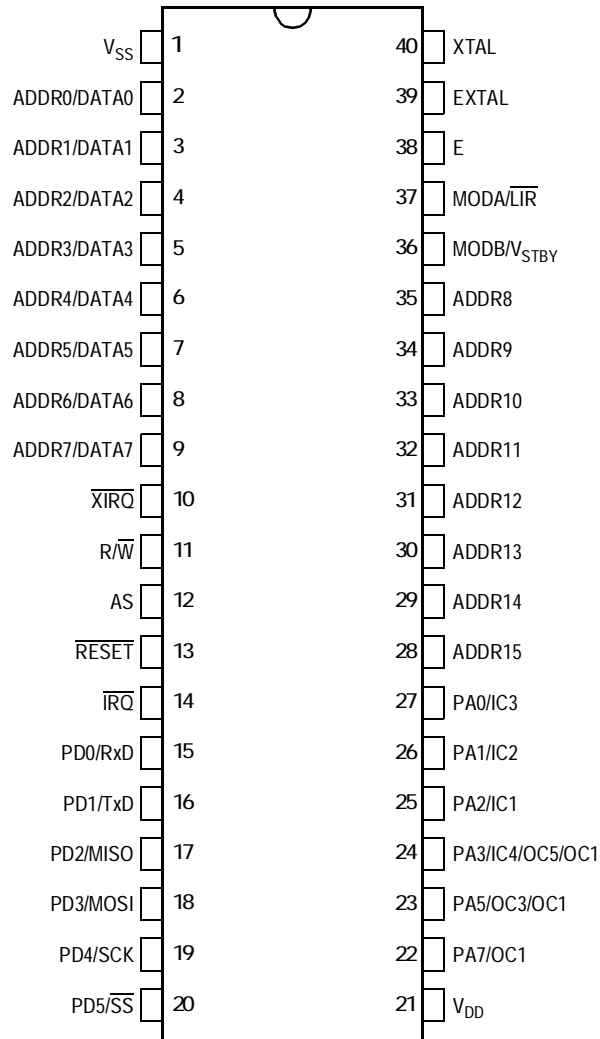


Figure 2-3. Pin Assignments for 40-Pin DIP

Section 3. Central Processor Unit (CPU)

3.1 Contents

| | | |
|---------|---|----|
| 3.2 | Introduction | 24 |
| 3.3 | CPU Registers | 24 |
| 3.3.1 | Accumulators A, B, and D | 25 |
| 3.3.2 | Index Register X (IX) | 26 |
| 3.3.3 | Index Register Y (IY) | 26 |
| 3.3.4 | Stack Pointer (SP) | 26 |
| 3.3.5 | Program Counter (PC) | 28 |
| 3.3.6 | Condition Code Register (CCR) | 29 |
| 3.3.6.1 | Carry/Borrow (C) | 29 |
| 3.3.6.2 | Overflow (V) | 29 |
| 3.3.6.3 | Zero (Z) | 29 |
| 3.3.6.4 | Negative (N) | 30 |
| 3.3.6.5 | I-Interrupt Mask (I) | 30 |
| 3.3.6.6 | Half Carry (H) | 30 |
| 3.3.6.7 | X-Interrupt Mask (X) | 30 |
| 3.3.6.8 | STOP Disable (S) | 31 |
| 3.4 | Data Types | 31 |
| 3.5 | Opcodes and Operands | 31 |
| 3.6 | Addressing Modes | 32 |
| 3.6.1 | Immediate | 32 |
| 3.6.2 | Direct | 33 |
| 3.6.3 | Extended | 33 |
| 3.6.4 | Indexed | 33 |
| 3.6.5 | Inherent | 33 |
| 3.6.6 | Relative | 34 |
| 3.7 | Instruction Set | 34 |

3.2 Introduction

This section presents information on M68HC11:

- Central processor unit (CPU) architecture
- Data types
- Addressing modes
- Instruction set
- Special operations such as subroutine calls and interrupts

The CPU is designed to treat all peripheral, input/output (I/O), and memory locations identically as addresses in the 64-Kbyte memory map. This is referred to as memory-mapped I/O. I/O has no instructions separate from those used by memory. This architecture also allows accessing an operand from an external memory location with no execution time penalty.

3.3 CPU Registers

M68HC11 CPU registers are an integral part of the CPU and are not addressed as if they were memory locations. The seven registers, discussed in the following paragraphs, are shown in [Figure 3-1](#).

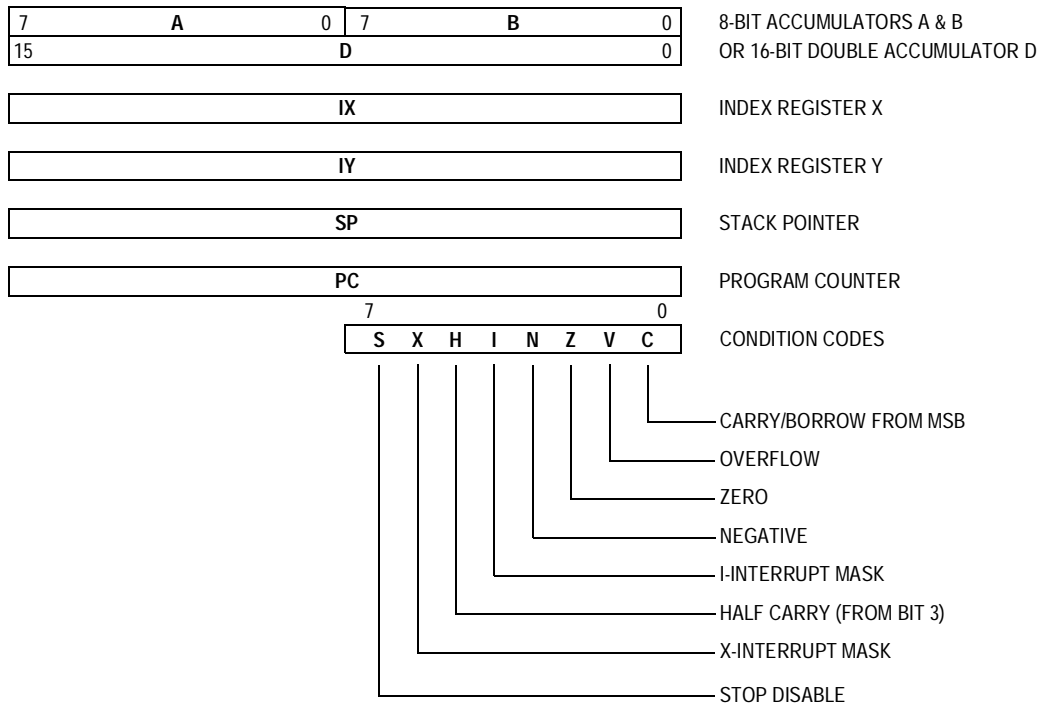


Figure 3-1. Programming Model

3.3.1 Accumulators A, B, and D

Accumulators A and B are general-purpose 8-bit registers that hold operands and results of arithmetic calculations or data manipulations. For some instructions, these two accumulators are treated as a single double-byte (16-bit) accumulator called accumulator D. Although most instructions can use accumulators A or B interchangeably, these exceptions apply:

- The ABX and ABY instructions add the contents of 8-bit accumulator B to the contents of 16-bit register X or Y, but there are no equivalent instructions that use A instead of B.
- The TAP and TPA instructions transfer data from accumulator A to the condition code register or from the condition code register to accumulator A. However, there are no equivalent instructions that use B rather than A.

- The decimal adjust accumulator A (DAA) instruction is used after binary-coded decimal (BCD) arithmetic operations, but there is no equivalent BCD instruction to adjust accumulator B.
- The add, subtract, and compare instructions associated with both A and B (ABA, SBA, and CBA) only operate in one direction, making it important to plan ahead to ensure that the correct operand is in the correct accumulator.

3.3.2 Index Register X (IX)

The IX register provides a 16-bit indexing value that can be added to the 8-bit offset provided in an instruction to create an effective address. The IX register can also be used as a counter or as a temporary storage register.

3.3.3 Index Register Y (IY)

The 16-bit IY register performs an indexed mode function similar to that of the IX register. However, most instructions using the IY register require an extra byte of machine code and an extra cycle of execution time because of the way the opcode map is implemented. Refer to [3.5 Opcodes and Operands](#) for further information.

3.3.4 Stack Pointer (SP)

The M68HC11 CPU has an automatic program stack. This stack can be located anywhere in the address space and can be any size up to the amount of memory available in the system. Normally, the SP is initialized by one of the first instructions in an application program. The stack is configured as a data structure that grows downward from high memory to low memory. Each time a new byte is pushed onto the stack, the SP is decremented. Each time a byte is pulled from the stack, the SP is incremented. At any given time, the SP holds the 16-bit address of the next free location in the stack. [Figure 3-2](#) is a summary of SP operations.

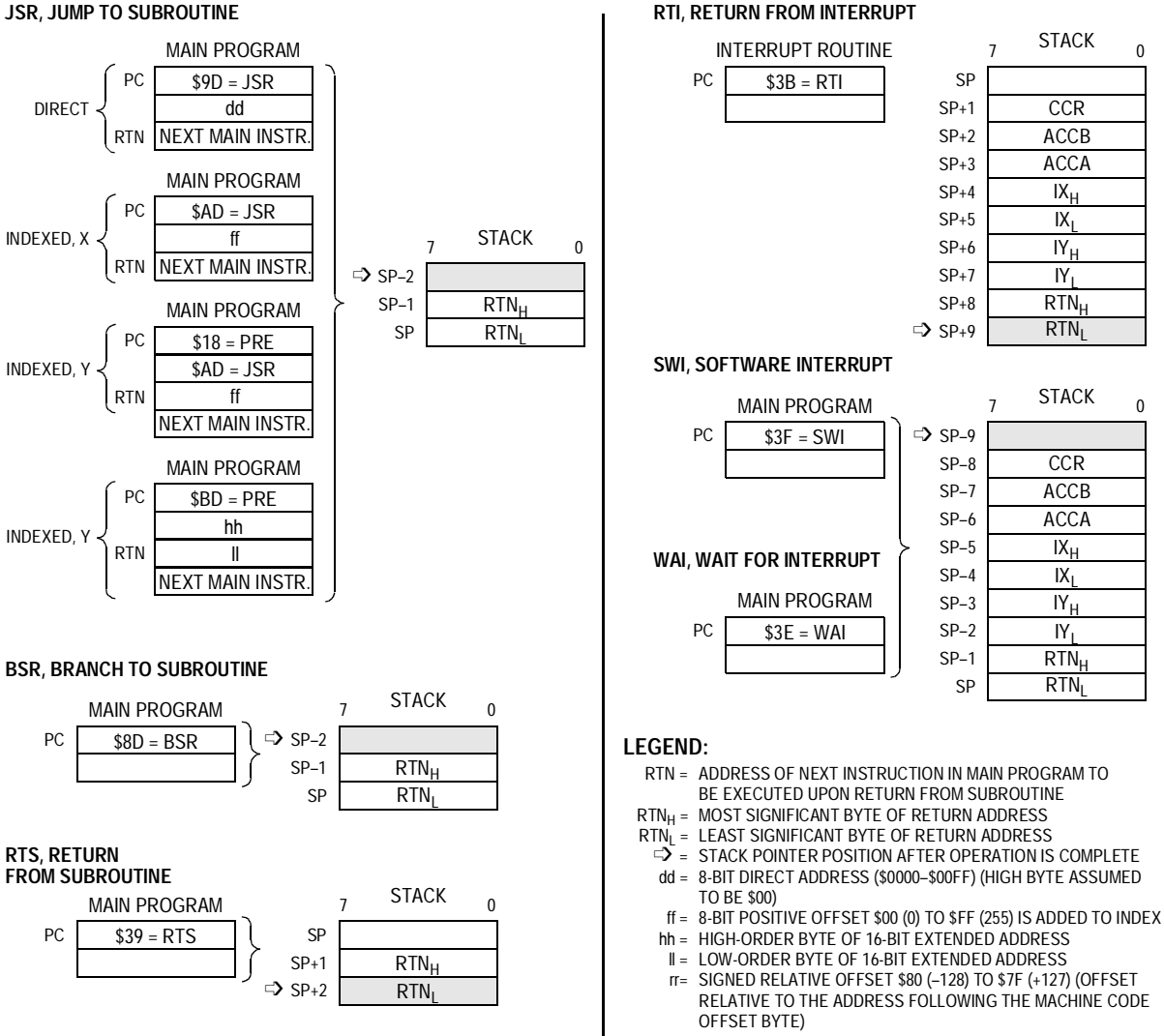


Figure 3-2. Stacking Operations

When a subroutine is called by a jump-to-subroutine (JSR) or branch-to-subroutine (BSR) instruction, the address of the instruction after the JSR or BSR is automatically pushed onto the stack, least significant byte first. When the subroutine is finished, a return-from-subroutine (RTS) instruction is executed. The RTS pulls the previously stacked return address from the stack and loads it into the program counter. Execution then continues at this recovered return address.

When an interrupt is recognized, the current instruction finishes normally, the return address (the current value in the program counter) is pushed onto the stack, all of the CPU registers are pushed onto the stack, and execution continues at the address specified by the vector for the interrupt.

At the end of the interrupt service routine, a return-from interrupt (RTI) instruction is executed. The RTI instruction causes the saved registers to be pulled off the stack in reverse order. Program execution resumes at the return address.

Certain instructions push and pull the A and B accumulators and the X and Y index registers and are often used to preserve program context. For example, pushing accumulator A onto the stack when entering a subroutine that uses accumulator A and then pulling accumulator A off the stack just before leaving the subroutine ensures that the contents of a register will be the same after returning from the subroutine as it was before starting the subroutine.

3.3.5 Program Counter (PC)

The program counter, a 16-bit register, contains the address of the next instruction to be executed. After reset, the program counter is initialized from one of six possible vectors, depending on operating mode and the cause of reset. See [Table 3-1](#).

Table 3-1. Reset Vector Comparison

| Mode | POR or $\overline{\text{RESET}}$ Pin | Clock Monitor | COP Watchdog |
|--------------|--------------------------------------|---------------|--------------|
| Normal | \$FFFE, \$FFFF | \$FFFC, D | \$FFFA, B |
| Test or boot | \$BFFE, \$BFFF | \$BFFC, D | \$BFFA, B |

3.3.6 Condition Code Register (CCR)

This 8-bit register contains:

- Five condition code indicators (C, V, Z, N, and H),
- Two interrupt masking bits ($\overline{\text{IRQ}}$ and $\overline{\text{XIRQ}}$)
- A stop disable bit (S)

In the M68HC11 CPU, condition codes are updated automatically by most instructions. For example, load accumulator A (LDAA) and store accumulator A (STAA) instructions automatically set or clear the N, Z, and V condition code flags. Pushes, pulls, add B to X (ABX), add B to Y (ABY), and transfer/exchange instructions do not affect the condition codes. Refer to [Table 3-2](#), which shows what condition codes are affected by a particular instruction.

3.3.6.1 Carry/Borrow (C)

The C bit is set if the arithmetic logic unit (ALU) performs a carry or borrow during an arithmetic operation. The C bit also acts as an error flag for multiply and divide operations. Shift and rotate instructions operate with and through the carry bit to facilitate multiple-word shift operations.

3.3.6.2 Overflow (V)

The overflow bit is set if an operation causes an arithmetic overflow. Otherwise, the V bit is cleared.

3.3.6.3 Zero (Z)

The Z bit is set if the result of an arithmetic, logic, or data manipulation operation is 0. Otherwise, the Z bit is cleared. Compare instructions do an internal implied subtraction and the condition codes, including Z, reflect the results of that subtraction. A few operations (INX, DEX, INY, and DEY) affect the Z bit and no other condition flags. For these operations, only = and \neq conditions can be determined.

3.3.6.4 Negative (N)

The N bit is set if the result of an arithmetic, logic, or data manipulation operation is negative (MSB = 1). Otherwise, the N bit is cleared. A result is said to be negative if its most significant bit (MSB) is a 1. A quick way to test whether the contents of a memory location has the MSB set is to load it into an accumulator and then check the status of the N bit.

3.3.6.5 I-Interrupt Mask (I)

The interrupt request (IRQ) mask (I bit) is a global mask that disables all maskable interrupt sources. While the I bit is set, interrupts can become pending, but the operation of the CPU continues uninterrupted until the I bit is cleared. After any reset, the I bit is set by default and can be cleared only by a software instruction. When an interrupt is recognized, the I bit is set after the registers are stacked, but before the interrupt vector is fetched. After the interrupt has been serviced, a return-from-interrupt instruction is normally executed, restoring the registers to the values that were present before the interrupt occurred. Normally, the I bit is 0 after a return from interrupt is executed. Although the I bit can be cleared within an interrupt service routine, "nesting" interrupts in this way should be done only when there is a clear understanding of latency and of the arbitration mechanism. Refer to [Section 5. Resets and Interrupts](#).

3.3.6.6 Half Carry (H)

The H bit is set when a carry occurs between bits 3 and 4 of the arithmetic logic unit during an ADD, ABA, or ADC instruction. Otherwise, the H bit is cleared. Half carry is used during BCD operations.

3.3.6.7 X-Interrupt Mask (X)

The XIRQ mask (X) bit disables interrupts from the $\overline{\text{XIRQ}}$ pin. After any reset, X is set by default and must be cleared by a software instruction. When an $\overline{\text{XIRQ}}$ interrupt is recognized, the X and I bits are set after the registers are stacked, but before the interrupt vector is fetched. After the interrupt has been serviced, an RTI instruction is normally executed, causing the registers to be restored to the values that were present

before the interrupt occurred. The X interrupt mask bit is set only by hardware ($\overline{\text{RESET}}$ or $\overline{\text{XIRQ}}$ acknowledge). X is cleared only by program instruction (TAP, where the associated bit of A is 0; or RTI, where bit 6 of the value loaded into the CCR from the stack has been cleared). There is no hardware action for clearing X.

3.3.6.8 STOP Disable (S)

Setting the STOP disable (S) bit prevents the STOP instruction from putting the M68HC11 into a low-power stop condition. If the STOP instruction is encountered by the CPU while the S bit is set, it is treated as a no-operation (NOP) instruction, and processing continues to the next instruction. S is set by reset; STOP is disabled by default.

3.4 Data Types

The M68HC11 CPU supports four data types:

1. Bit data
2. 8-bit and 16-bit signed and unsigned integers
3. 16-bit unsigned fractions
4. 16-bit addresses

A byte is eight bits wide and can be accessed at any byte location. A word is composed of two consecutive bytes with the most significant byte at the lower value address. Because the M68HC11 is an 8-bit CPU, there are no special requirements for alignment of instructions or operands.

3.5 Opcodes and Operands

The M68HC11 Family of microcontrollers uses 8-bit opcodes. Each opcode identifies a particular instruction and associated addressing mode to the CPU. Several opcodes are required to provide each instruction with a range of addressing capabilities. Only 256 opcodes

would be available if the range of values were restricted to the number able to be expressed in 8-bit binary numbers.

A 4-page opcode map has been implemented to expand the number of instructions. An additional byte, called a prebyte, directs the processor from page 0 of the opcode map to one of the other three pages. As its name implies, the additional byte precedes the opcode.

A complete instruction consists of a prebyte, if any, an opcode, and zero, one, two, or three operands. The operands contain information the CPU needs for executing the instruction. Complete instructions can be from one to five bytes long.

3.6 Addressing Modes

Six addressing modes can be used to access memory:

- Immediate
- Direct
- Extended
- Indexed
- Inherent
- Relative

These modes are detailed in the following paragraphs. All modes except inherent mode use an effective address. The effective address is the memory address from which the argument is fetched or stored or the address from which execution is to proceed. The effective address can be specified within an instruction, or it can be calculated.

3.6.1 Immediate

In the immediate addressing mode, an argument is contained in the byte(s) immediately following the opcode. The number of bytes following the opcode matches the size of the register or memory location being operated on. There are 2-, 3-, and 4- (if prebyte is required) byte

immediate instructions. The effective address is the address of the byte following the instruction.

3.6.2 Direct

In the direct addressing mode, the low-order byte of the operand address is contained in a single byte following the opcode, and the high-order byte of the address is assumed to be \$00. Addresses \$00–\$FF are thus accessed directly, using 2-byte instructions. Execution time is reduced by eliminating the additional memory access required for the high-order address byte. In most applications, this 256-byte area is reserved for frequently referenced data. In M68HC11 MCUs, the memory map can be configured for combinations of internal registers, RAM, or external memory to occupy these addresses.

3.6.3 Extended

In the extended addressing mode, the effective address of the argument is contained in two bytes following the opcode byte. These are 3-byte instructions (or 4-byte instructions if a prebyte is required). One or two bytes are needed for the opcode and two for the effective address.

3.6.4 Indexed

In the indexed addressing mode, an 8-bit unsigned offset contained in the instruction is added to the value contained in an index register (IX or IY). The sum is the effective address. This addressing mode allows referencing any memory location in the 64-Kbyte address space. These are 2- to 5-byte instructions, depending on whether a prebyte is required.

3.6.5 Inherent

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations that use only the index registers or accumulators, as well as control instructions with no arguments, are included in this addressing mode. These are 1- or 2-byte instructions.

3.6.6 Relative

The relative addressing mode is used only for branch instructions. If the branch condition is true, an 8-bit signed offset included in the instruction is added to the contents of the program counter to form the effective branch address. Otherwise, control proceeds to the next instruction. These are usually 2-byte instructions.

3.7 Instruction Set

Refer to [Table 3-2](#), which shows all the M68HC11 instructions in all possible addressing modes. For each instruction, the table shows the operand construction, the number of machine code bytes, and execution time in CPU E-clock cycles.

Table 3-2. Instruction Set (Sheet 1 of 7)


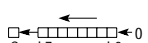

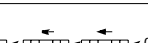
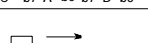
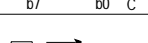
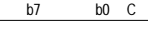
| Mnemonic | Operation | Description | Addressing Mode | Instruction | | | Condition Codes | | | | | | | | | |
|------------------|--------------------------|---|-----------------------|-------------------------------------|---------|--------|-----------------|---|---|---|---|---|---|---|---|---|
| | | | | Opcode | Operand | Cycles | S | X | H | I | N | Z | V | C | | |
| ABA | Add Accumulators | $A + B \Rightarrow A$ | INH | | 1B | — | 2 | — | — | Δ | — | Δ | Δ | Δ | Δ | |
| ABX | Add B to X | $IX + (00 : B) \Rightarrow IX$ | INH | | 3A | — | 3 | — | — | — | — | — | — | — | — | |
| ABY | Add B to Y | $IY + (00 : B) \Rightarrow IY$ | INH | 18 | 3A | — | 4 | — | — | — | — | — | — | — | — | |
| ADCA (opr) | Add with Carry to A | $A + M + C \Rightarrow A$ | A A A A A | IMM DIR EXT IND,X IND,Y | 18 | 89 | ii | 2 | — | — | Δ | — | Δ | Δ | Δ | Δ |
| | | | | | | 99 | dd | 3 | | | | | | | | |
| | | | | | | B9 | hh ll | 4 | | | | | | | | |
| | | | | | | A9 | ff | 4 | | | | | | | | |
| | | | | | | A9 | ff | 5 | | | | | | | | |
| ADCB (opr) | Add with Carry to B | $B + M + C \Rightarrow B$ | B B B B B | IMM DIR EXT IND,X IND,Y | 18 | C9 | ii | 2 | — | — | Δ | — | Δ | Δ | Δ | Δ |
| | | | | | | D9 | dd | 3 | | | | | | | | |
| | | | | | | F9 | hh ll | 4 | | | | | | | | |
| | | | | | | E9 | ff | 4 | | | | | | | | |
| | | | | | | E9 | ff | 5 | | | | | | | | |
| ADDA (opr) | Add Memory to A | $A + M \Rightarrow A$ | A A A A A | IMM DIR EXT IND,X IND,Y | 18 | 8B | ii | 2 | — | — | Δ | — | Δ | Δ | Δ | Δ |
| | | | | | | 9B | dd | 3 | | | | | | | | |
| | | | | | | BB | hh ll | 4 | | | | | | | | |
| | | | | | | AB | ff | 4 | | | | | | | | |
| | | | | | | AB | ff | 5 | | | | | | | | |
| ADDB (opr) | Add Memory to B | $B + M \Rightarrow B$ | B B B B B | IMM DIR EXT IND,X IND,Y | 18 | CB | ii | 2 | — | — | Δ | — | Δ | Δ | Δ | Δ |
| | | | | | | DB | dd | 3 | | | | | | | | |
| | | | | | | FB | hh ll | 4 | | | | | | | | |
| | | | | | | EB | ff | 4 | | | | | | | | |
| | | | | | | EB | ff | 5 | | | | | | | | |
| ADDD (opr) | Add 16-Bit to D | $D + (M : M + 1) \Rightarrow D$ | | IMM DIR EXT IND,X IND,Y | 18 | C3 | jj kk | 4 | — | — | — | — | Δ | Δ | Δ | Δ |
| | | | | | | D3 | dd | 5 | | | | | | | | |
| | | | | | | F3 | hh ll | 6 | | | | | | | | |
| | | | | | | E3 | ff | 6 | | | | | | | | |
| | | | | | | E3 | ff | 7 | | | | | | | | |
| ANDA (opr) | AND A with Memory | $A \cdot M \Rightarrow A$ | A A A A A | IMM DIR EXT IND,X IND,Y | 18 | 84 | ii | 2 | — | — | — | — | Δ | Δ | 0 | — |
| | | | | | | 94 | dd | 3 | | | | | | | | |
| | | | | | | B4 | hh ll | 4 | | | | | | | | |
| | | | | | | A4 | ff | 4 | | | | | | | | |
| | | | | | | A4 | ff | 5 | | | | | | | | |
| ANDB (opr) | AND B with Memory | $B \cdot M \Rightarrow B$ | B B B B B | IMM DIR EXT IND,X IND,Y | 18 | C4 | ii | 2 | — | — | — | — | Δ | Δ | 0 | — |
| | | | | | | D4 | dd | 3 | | | | | | | | |
| | | | | | | F4 | hh ll | 4 | | | | | | | | |
| | | | | | | E4 | ff | 4 | | | | | | | | |
| | | | | | | E4 | ff | 5 | | | | | | | | |
| ASL (opr) | Arithmetic Shift Left |  | | EXT IND,X IND,Y | 18 | 78 | hh ll | 6 | — | — | — | — | Δ | Δ | Δ | Δ |
| | | | | | | 68 | ff | 6 | | | | | | | | |
| | | | | | | 68 | ff | 7 | | | | | | | | |
| ASLA | Arithmetic Shift Left A |  | A | INH | | 48 | — | 2 | — | — | — | — | Δ | Δ | Δ | Δ |
| ASLB | Arithmetic Shift Left B |  | B | INH | | 58 | — | 2 | — | — | — | — | Δ | Δ | Δ | Δ |
| ASLD | Arithmetic Shift Left D |  | | INH | | 05 | — | 3 | — | — | — | — | Δ | Δ | Δ | Δ |
| ASR | Arithmetic Shift Right |  | | EXT IND,X IND,Y | 18 | 77 | hh ll | 6 | — | — | — | — | Δ | Δ | Δ | Δ |
| | | | | | | 67 | ff | 6 | | | | | | | | |
| | | | | | | 67 | ff | 7 | | | | | | | | |
| ASRA | Arithmetic Shift Right A |  | A | INH | | 47 | — | 2 | — | — | — | — | Δ | Δ | Δ | Δ |
| ASRB | Arithmetic Shift Right B |  | B | INH | | 57 | — | 2 | — | — | — | — | Δ | Δ | Δ | Δ |
| BCC (rel) | Branch if Carry Clear | ? C = 0 | | REL | | 24 | rr | 3 | — | — | — | — | — | — | — | — |
| BCLR (opr) (msk) | Clear Bit(s) | $M \cdot (mm) \Rightarrow M$ | | DIR IND,X IND,Y | 18 | 15 | dd mm | 6 | — | — | — | — | Δ | Δ | 0 | — |
| | | | | | | 1D | ff mm | 7 | | | | | | | | |
| | | | | | | 1D | ff mm | 8 | | | | | | | | |

Table 3-2. Instruction Set (Sheet 2 of 7)

| Mnemonic | Operation | Description | Addressing Mode | Instruction | | | Condition Codes | | | | | | | |
|------------------------|---------------------------|--------------------------------|-----------------|-------------|----------|--------|-----------------|---|---|---|----------|----------|----------|----------|
| | | | | Opcode | Operand | Cycles | S | X | H | I | N | Z | V | C |
| BCS (rel) | Branch if Carry Set | ? C = 1 | REL | 25 | rr | 3 | — | — | — | — | — | — | — | — |
| BEQ (rel) | Branch if = Zero | ? Z = 1 | REL | 27 | rr | 3 | — | — | — | — | — | — | — | — |
| BGE (rel) | Branch if Δ Zero | ? N \oplus V = 0 | REL | 2C | rr | 3 | — | — | — | — | — | — | — | — |
| BGT (rel) | Branch if > Zero | ? Z + (N \oplus V) = 0 | REL | 2E | 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 | — | — | — | — | — | — | — | — |
| BITA (opr) | Bit(s) Test A with Memory | A • M | A IMM | 85 | ii | 2 | — | — | — | — | Δ | Δ | 0 | — |
| | | | A DIR | 95 | dd | 3 | | | | | | | | |
| | | | A EXT | B5 | hh ll | 4 | | | | | | | | |
| | | | A IND,X | A5 | ff | 4 | | | | | | | | |
| | | | A IND,Y | A5 | ff | 5 | | | | | | | | |
| BITB (opr) | Bit(s) Test B with Memory | B • M | B IMM | C5 | ii | 2 | — | — | — | — | Δ | Δ | 0 | — |
| | | | B DIR | D5 | dd | 3 | | | | | | | | |
| | | | B EXT | F5 | hh ll | 4 | | | | | | | | |
| | | | B IND,X | E5 | ff | 4 | | | | | | | | |
| | | | B IND,Y | E5 | ff | 5 | | | | | | | | |
| BLE (rel) | Branch if Δ Zero | ? Z + (N \oplus V) = 1 | REL | 2F | rr | 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 | — | — | — | — | — | — | — | — |
| BLT (rel) | Branch if < Zero | ? N \oplus V = 1 | REL | 2D | rr | 3 | — | — | — | — | — | — | — | — |
| BMI (rel) | Branch if Minus | ? N = 1 | REL | 2B | rr | 3 | — | — | — | — | — | — | — | — |
| BNE (rel) | Branch if not = Zero | ? Z = 0 | REL | 26 | rr | 3 | — | — | — | — | — | — | — | — |
| BPL (rel) | Branch if Plus | ? N = 0 | REL | 2A | rr | 3 | — | — | — | — | — | — | — | — |
| BRA (rel) | Branch Always | ? 1 = 1 | REL | 20 | rr | 3 | — | — | — | — | — | — | — | — |
| BRCLR(opr) (msk) (rel) | Branch if Bit(s) Clear | ? M • mm = 0 | DIR | 13 | dd mm rr | 6 | — | — | — | — | — | — | — | — |
| | | | IND,X | 1F | ff mm rr | 7 | | | | | | | | |
| | | | IND,Y | 1F | ff mm rr | 8 | | | | | | | | |
| BRN (rel) | Branch Never | ? 1 = 0 | REL | 21 | rr | 3 | — | — | — | — | — | — | — | — |
| BRSET(opr) (msk) (rel) | Branch if Bit(s) Set | ? (M) • mm = 0 | DIR | 12 | dd mm rr | 6 | — | — | — | — | — | — | — | — |
| | | | IND,X | 1E | ff mm rr | 7 | | | | | | | | |
| | | | IND,Y | 1E | ff mm rr | 8 | | | | | | | | |
| BSET (opr) (msk) | Set Bit(s) | M + mm \Rightarrow M | DIR | 14 | dd mm | 6 | — | — | — | — | Δ | Δ | 0 | — |
| | | | IND,X | 1C | ff mm | 7 | | | | | | | | |
| | | | IND,Y | 1C | ff mm | 8 | | | | | | | | |
| BSR (rel) | Branch to Subroutine | See Figure 3-2 | REL | 8D | rr | 6 | — | — | — | — | — | — | — | — |
| BVC (rel) | Branch if Overflow Clear | ? V = 0 | REL | 28 | rr | 3 | — | — | — | — | — | — | — | — |
| BVS (rel) | Branch if Overflow Set | ? V = 1 | REL | 29 | rr | 3 | — | — | — | — | — | — | — | — |
| CBA | Compare A to B | A – B | INH | 11 | — | 2 | — | — | — | — | Δ | Δ | Δ | Δ |
| CLC | Clear Carry Bit | 0 \Rightarrow C | INH | 0C | — | 2 | — | — | — | — | — | — | — | 0 |
| CLI | Clear Interrupt Mask | 0 \Rightarrow I | INH | 0E | — | 2 | — | — | — | 0 | — | — | — | — |
| CLR (opr) | Clear Memory Byte | 0 \Rightarrow M | EXT | 7F | hh ll | 6 | — | — | — | — | 0 | 1 | 0 | 0 |
| | | | IND,X | 6F | ff | 6 | | | | | | | | |
| | | | IND,Y | 6F | ff | 7 | | | | | | | | |
| CLRA | Clear Accumulator A | 0 \Rightarrow A | A INH | 4F | — | 2 | — | — | — | — | 0 | 1 | 0 | 0 |
| CLRB | Clear Accumulator B | 0 \Rightarrow B | B INH | 5F | — | 2 | — | — | — | — | 0 | 1 | 0 | 0 |
| CLV | Clear Overflow Flag | 0 \Rightarrow V | INH | 0A | — | 2 | — | — | — | — | — | — | 0 | — |
| CMPA (opr) | Compare A to Memory | A – M | A IMM | 81 | ii | 2 | — | — | — | — | Δ | Δ | Δ | Δ |
| | | | A DIR | 91 | dd | 3 | | | | | | | | |
| | | | A EXT | B1 | hh ll | 4 | | | | | | | | |
| | | | A IND,X | A1 | ff | 4 | | | | | | | | |
| | | | A IND,Y | A1 | ff | 5 | | | | | | | | |

Table 3-2. Instruction Set (Sheet 3 of 7)

| Mnemonic | Operation | Description | Addressing Mode | Instruction | | | Condition Codes | | | | | | | |
|------------|-----------------------------|--------------------|-----------------|-------------|---------|--------|-----------------|---|---|---|---|---|---|---|
| | | | | Opcode | Operand | Cycles | S | X | H | I | N | Z | V | C |
| CMPB (opr) | Compare B to Memory | B – M | B IMM | C1 | ii | 2 | — | — | — | — | Δ | Δ | Δ | Δ |
| | | | B DIR | D1 | dd | 3 | | | | | | | | |
| | | | B EXT | F1 | hh ll | 4 | | | | | | | | |
| | | | B IND,X | E1 | ff | 4 | | | | | | | | |
| | | | B IND,Y | E1 | ff | 5 | | | | | | | | |
| COM (opr) | Ones Complement Memory Byte | \$FF – M ⇒ M | EXT | 73 | hh ll | 6 | — | — | — | — | Δ | Δ | 0 | 1 |
| | | | IND,X | 63 | ff | 6 | | | | | | | | |
| | | | IND,Y | 63 | ff | 7 | | | | | | | | |
| COMA | Ones Complement A | \$FF – A ⇒ A | A INH | 43 | — | 2 | — | — | — | — | Δ | Δ | 0 | 1 |
| COMB | Ones Complement B | \$FF – B ⇒ B | B INH | 53 | — | 2 | — | — | — | — | Δ | Δ | 0 | 1 |
| CPD (opr) | Compare D to Memory 16-Bit | D – M : M + 1 | IMM | 1A 83 | jj kk | 5 | — | — | — | — | Δ | Δ | Δ | Δ |
| | | | DIR | 1A 93 | dd | 6 | | | | | | | | |
| | | | EXT | 1A B3 | hh ll | 7 | | | | | | | | |
| | | | IND,X | 1A A3 | ff | 7 | | | | | | | | |
| | | | IND,Y | CD A3 | ff | 7 | | | | | | | | |
| CPX (opr) | Compare X to Memory 16-Bit | IX – M : M + 1 | IMM | 8C | jj kk | 4 | — | — | — | — | Δ | Δ | Δ | Δ |
| | | | DIR | 9C | dd | 5 | | | | | | | | |
| | | | EXT | BC | hh ll | 6 | | | | | | | | |
| | | | IND,X | AC | ff | 6 | | | | | | | | |
| | | | IND,Y | CD AC | ff | 7 | | | | | | | | |
| CPY (opr) | Compare Y to Memory 16-Bit | IY – M : M + 1 | IMM | 18 8C | jj kk | 5 | — | — | — | — | Δ | Δ | Δ | Δ |
| | | | DIR | 18 9C | dd | 6 | | | | | | | | |
| | | | EXT | 18 BC | hh ll | 7 | | | | | | | | |
| | | | IND,X | 1A AC | ff | 7 | | | | | | | | |
| | | | IND,Y | 18 AC | ff | 7 | | | | | | | | |
| DAA | Decimal Adjust A | Adjust Sum to BCD | INH | 19 | — | 2 | — | — | — | — | Δ | Δ | Δ | Δ |
| DEC (opr) | Decrement Memory Byte | M – 1 ⇒ M | EXT | 7A | hh ll | 6 | — | — | — | — | Δ | Δ | Δ | — |
| | | | IND,X | 6A | ff | 6 | | | | | | | | |
| | | | IND,Y | 18 6A | ff | 7 | | | | | | | | |
| DECA | Decrement Accumulator A | A – 1 ⇒ A | A INH | 4A | — | 2 | — | — | — | — | Δ | Δ | Δ | — |
| DECB | Decrement Accumulator B | B – 1 ⇒ B | B INH | 5A | — | 2 | — | — | — | — | Δ | Δ | Δ | — |
| DES | Decrement Stack Pointer | SP – 1 ⇒ SP | INH | 34 | — | 3 | — | — | — | — | — | — | — | — |
| DEX | Decrement Index Register X | IX – 1 ⇒ IX | INH | 09 | — | 3 | — | — | — | — | — | Δ | — | — |
| DEY | Decrement Index Register Y | IY – 1 ⇒ IY | INH | 18 09 | — | 4 | — | — | — | — | — | Δ | — | — |
| EORA (opr) | Exclusive OR A with Memory | A ⊕ M ⇒ A | A IMM | 88 | ii | 2 | — | — | — | — | Δ | Δ | 0 | — |
| | | | A DIR | 98 | dd | 3 | | | | | | | | |
| | | | A EXT | B8 | hh ll | 4 | | | | | | | | |
| | | | A IND,X | A8 | ff | 4 | | | | | | | | |
| | | | A IND,Y | 18 A8 | ff | 5 | | | | | | | | |
| EORB (opr) | Exclusive OR B with Memory | B ⊕ M ⇒ B | B IMM | C8 | ii | 2 | — | — | — | — | Δ | Δ | 0 | — |
| | | | B DIR | D8 | dd | 3 | | | | | | | | |
| | | | B EXT | F8 | hh ll | 4 | | | | | | | | |
| | | | B IND,X | E8 | ff | 4 | | | | | | | | |
| | | | B IND,Y | 18 E8 | ff | 5 | | | | | | | | |
| FDIV | Fractional Divide 16 by 16 | D / IX ⇒ IX; r ⇒ D | INH | 03 | — | 41 | — | — | — | — | — | Δ | Δ | Δ |
| IDIV | Integer Divide 16 by 16 | D / IX ⇒ IX; r ⇒ D | INH | 02 | — | 41 | — | — | — | — | — | Δ | 0 | Δ |
| INC (opr) | Increment Memory Byte | M + 1 ⇒ M | EXT | 7C | hh ll | 6 | — | — | — | — | Δ | Δ | Δ | — |
| | | | IND,X | 6C | ff | 6 | | | | | | | | |
| | | | IND,Y | 18 6C | ff | 7 | | | | | | | | |
| INCA | Increment Accumulator A | A + 1 ⇒ A | A INH | 4C | — | 2 | — | — | — | — | Δ | Δ | Δ | — |

Table 3-2. Instruction Set (Sheet 4 of 7)

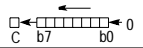
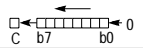
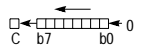
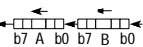
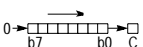
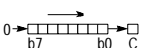
| Mnemonic | Operation | Description | Addressing Mode | Instruction | | | Condition Codes | | | | | | | |
|------------|----------------------------|---|---|---|----------------------------------|-----------------------|-----------------|---|---|---|---|---|---|---|
| | | | | Opcode | Operand | Cycles | S | X | H | I | N | Z | V | C |
| INCB | Increment Accumulator B | $B + 1 \Rightarrow B$ | B INH | 5C | — | 2 | — | — | — | — | Δ | Δ | Δ | — |
| INS | Increment Stack Pointer | $SP + 1 \Rightarrow SP$ | INH | 31 | — | 3 | — | — | — | — | — | — | — | — |
| INX | Increment Index Register X | $IX + 1 \Rightarrow IX$ | INH | 08 | — | 3 | — | — | — | — | — | Δ | — | — |
| INY | Increment Index Register Y | $IY + 1 \Rightarrow IY$ | INH | 18 08 | — | 4 | — | — | — | — | — | Δ | — | — |
| JMP (opr) | Jump | See Figure 3-2 | EXT IND,X IND,Y | 7E 6E 6E | hh ll ff ff | 3 3 4 | — | — | — | — | — | — | — | — |
| JSR (opr) | Jump to Subroutine | See Figure 3-2 | DIR EXT IND,X IND,Y | 9D BD AD AD | dd hh ll ff ff | 5 6 6 7 | — | — | — | — | — | — | — | — |
| LDA (opr) | Load Accumulator A | $M \Rightarrow A$ | A IMM A DIR A EXT A IND,X A IND,Y | 86 96 B6 A6 A6 | ii dd hh ll ff ff | 2 3 4 4 5 | — | — | — | — | Δ | Δ | 0 | — |
| LDAB (opr) | Load Accumulator B | $M \Rightarrow B$ | B IMM B DIR B EXT B IND,X B IND,Y | C6 D6 F6 E6 E6 | ii dd hh ll ff ff | 2 3 4 4 5 | — | — | — | — | Δ | Δ | 0 | — |
| LDD (opr) | Load Double Accumulator D | $M \Rightarrow A, M + 1 \Rightarrow B$ | IMM DIR EXT IND,X IND,Y | CC DC FC EC EC | jj kk dd hh ll ff ff | 3 4 5 5 6 | — | — | — | — | Δ | Δ | 0 | — |
| LDS (opr) | Load Stack Pointer | $M : M + 1 \Rightarrow SP$ | IMM DIR EXT IND,X IND,Y | 8E 9E BE AE AE | jj kk dd hh ll ff ff | 3 4 5 5 6 | — | — | — | — | Δ | Δ | 0 | — |
| LDX (opr) | Load Index Register X | $M : M + 1 \Rightarrow IX$ | IMM DIR EXT IND,X IND,Y | CE DE FE EE EE | jj kk dd hh ll ff ff | 3 4 5 5 6 | — | — | — | — | Δ | Δ | 0 | — |
| LDY (opr) | Load Index Register Y | $M : M + 1 \Rightarrow IY$ | IMM DIR EXT IND,X IND,Y | 18 CE 18 DE 18 FE 1A EE 18 EE | jj kk dd hh ll ff ff | 4 5 6 6 6 | — | — | — | — | Δ | Δ | 0 | — |
| LSL (opr) | Logical Shift Left |  | EXT IND,X IND,Y | 78 68 68 | hh ll ff ff | 6 6 7 | — | — | — | — | Δ | Δ | Δ | Δ |
| LSLA | Logical Shift Left A |  | A INH | 48 | — | 2 | — | — | — | — | Δ | Δ | Δ | Δ |
| LSLB | Logical Shift Left B |  | B INH | 58 | — | 2 | — | — | — | — | Δ | Δ | Δ | Δ |
| LSLD | Logical Shift Left Double |  | INH | 05 | — | 3 | — | — | — | — | Δ | Δ | Δ | Δ |
| LSR (opr) | Logical Shift Right |  | EXT IND,X IND,Y | 74 64 64 | hh ll ff ff | 6 6 7 | — | — | — | — | 0 | Δ | Δ | Δ |
| LSRA | Logical Shift Right A |  | A INH | 44 | — | 2 | — | — | — | — | 0 | Δ | Δ | Δ |

Table 3-2. Instruction Set (Sheet 5 of 7)

| Mnemonic | Operation | Description | Addressing Mode | Instruction | | | Condition Codes | | | | | | | | | |
|------------|------------------------------|--|-----------------|-------------------------------------|---------|--------|-----------------|---|---|---|---|---|---|---|---|---|
| | | | | Opcode | Operand | Cycles | S | X | H | I | N | Z | V | C | | |
| LSRB | Logical Shift Right B | | B | INH | 54 | — | 2 | — | — | — | — | 0 | Δ | Δ | Δ | |
| LSRD | Logical Shift Right Double | | | INH | 04 | — | 3 | — | — | — | — | 0 | Δ | Δ | Δ | |
| MUL | Multiply 8 by 8 | $A * B \Rightarrow D$ | | INH | 3D | — | 10 | — | — | — | — | — | — | — | Δ | |
| NEG (opr) | Two's Complement Memory Byte | $0 - M \Rightarrow M$ | | EXT IND,X IND,Y | 70 | hh ll | 6 | — | — | — | — | Δ | Δ | Δ | Δ | |
| | | | | | 60 | ff | 6 | — | — | — | — | — | — | — | — | — |
| NEGA | Two's Complement A | $0 - A \Rightarrow A$ | A | INH | 40 | — | 2 | — | — | — | — | Δ | Δ | Δ | Δ | |
| NEGB | Two's Complement B | $0 - B \Rightarrow B$ | B | INH | 50 | — | 2 | — | — | — | — | Δ | Δ | Δ | Δ | |
| NOP | No operation | No Operation | | INH | 01 | — | 2 | — | — | — | — | — | — | — | — | |
| ORAA (opr) | OR Accumulator A (Inclusive) | $A + M \Rightarrow A$ | A | IMM DIR EXT IND,X IND,Y | 8A | ii | 2 | — | — | — | — | Δ | Δ | 0 | — | |
| | | | | | 9A | dd | 3 | — | — | — | — | — | — | — | — | — |
| | | | | | BA | hh ll | 4 | — | — | — | — | — | — | — | — | — |
| | | | | | AA | ff | 4 | — | — | — | — | — | — | — | — | — |
| | | | | | AA | ff | 5 | — | — | — | — | — | — | — | — | — |
| ORAB (opr) | OR Accumulator B (Inclusive) | $B + M \Rightarrow B$ | B | IMM DIR EXT IND,X IND,Y | CA | ii | 2 | — | — | — | — | Δ | Δ | 0 | — | |
| | | | | | DA | dd | 3 | — | — | — | — | — | — | — | — | |
| | | | | | FA | hh ll | 4 | — | — | — | — | — | — | — | — | — |
| | | | | | EA | ff | 4 | — | — | — | — | — | — | — | — | — |
| | | | | | EA | ff | 5 | — | — | — | — | — | — | — | — | — |
| PSHA | Push A onto Stack | $A \Rightarrow \text{Stk}, SP = SP - 1$ | A | INH | 36 | — | 3 | — | — | — | — | — | — | — | — | |
| PSHB | Push B onto Stack | $B \Rightarrow \text{Stk}, SP = SP - 1$ | B | INH | 37 | — | 3 | — | — | — | — | — | — | — | — | |
| PSHX | Push X onto Stack (Lo First) | $IX \Rightarrow \text{Stk}, SP = SP - 2$ | | INH | 3C | — | 4 | — | — | — | — | — | — | — | — | |
| PSHY | Push Y onto Stack (Lo First) | $IY \Rightarrow \text{Stk}, SP = SP - 2$ | | INH | 18 3C | — | 5 | — | — | — | — | — | — | — | — | |
| PULA | Pull A from Stack | $SP = SP + 1, A \Leftarrow \text{Stk}$ | A | INH | 32 | — | 4 | — | — | — | — | — | — | — | — | |
| PULB | Pull B from Stack | $SP = SP + 1, B \Leftarrow \text{Stk}$ | B | INH | 33 | — | 4 | — | — | — | — | — | — | — | — | |
| PULX | Pull X From Stack (Hi First) | $SP = SP + 2, IX \Leftarrow \text{Stk}$ | | INH | 38 | — | 5 | — | — | — | — | — | — | — | — | |
| PULY | Pull Y from Stack (Hi First) | $SP = SP + 2, IY \Leftarrow \text{Stk}$ | | INH | 18 38 | — | 6 | — | — | — | — | — | — | — | — | |
| ROL (opr) | Rotate Left | | | EXT IND,X IND,Y | 79 | hh ll | 6 | — | — | — | — | Δ | Δ | Δ | Δ | |
| | | | | | 69 | ff | 6 | — | — | — | — | — | — | — | — | |
| | | | | | 69 | ff | 7 | — | — | — | — | — | — | — | — | — |
| ROLA | Rotate Left A | | A | INH | 49 | — | 2 | — | — | — | — | Δ | Δ | Δ | Δ | |
| ROLB | Rotate Left B | | B | INH | 59 | — | 2 | — | — | — | — | Δ | Δ | Δ | Δ | |
| ROR (opr) | Rotate Right | | | EXT IND,X IND,Y | 76 | hh ll | 6 | — | — | — | — | Δ | Δ | Δ | Δ | |
| | | | | | 66 | ff | 6 | — | — | — | — | — | — | — | — | |
| | | | | | 66 | ff | 7 | — | — | — | — | — | — | — | — | — |
| RORA | Rotate Right A | | A | INH | 46 | — | 2 | — | — | — | — | Δ | Δ | Δ | Δ | |
| RORB | Rotate Right B | | B | INH | 56 | — | 2 | — | — | — | — | Δ | Δ | Δ | Δ | |
| RTI | Return from Interrupt | See Figure 3-2 | | INH | 3B | — | 12 | Δ | ↓ | Δ | Δ | Δ | Δ | Δ | Δ | |

Table 3-2. Instruction Set (Sheet 6 of 7)

| Mnemonic | Operation | Description | Addressing Mode | Instruction | | | Condition Codes | | | | | | | | | |
|------------|----------------------------|--|-----------------|-------------|---------|--------|-----------------|--------------|----------|----------|----------|----------|----------|----------|----------|---|
| | | | | Opcode | Operand | Cycles | S | X | H | I | N | Z | V | C | | |
| RTS | Return from Subroutine | See Figure 3-2 | INH | 39 | — | 5 | — | — | — | — | — | — | — | — | | |
| SBA | Subtract B from A | $A - B \Rightarrow A$ | INH | 10 | — | 2 | — | — | — | — | Δ | Δ | Δ | Δ | | |
| SBCA (opr) | Subtract with Carry from A | $A - M - C \Rightarrow A$ | A | IMM | 82 | ii | 2 | — | — | — | — | Δ | Δ | Δ | Δ | |
| | | | | DIR | 92 | dd | 3 | | | | | | | | | |
| | | | | EXT | B2 | hh ll | 4 | | | | | | | | | |
| | | | | IND,X | A2 | ff | 4 | | | | | | | | | |
| | | | | IND,Y | A2 | ff | 5 | | | | | | | | | |
| SBCB (opr) | Subtract with Carry from B | $B - M - C \Rightarrow B$ | B | IMM | C2 | ii | 2 | — | — | — | — | Δ | Δ | Δ | Δ | |
| | | | | DIR | D2 | dd | 3 | | | | | | | | | |
| | | | | EXT | F2 | hh ll | 4 | | | | | | | | | |
| | | | | IND,X | E2 | ff | 4 | | | | | | | | | |
| | | | | IND,Y | E2 | ff | 5 | | | | | | | | | |
| SEC | Set Carry | $1 \Rightarrow C$ | INH | 0D | — | 2 | — | — | — | — | — | — | — | 1 | | |
| SEI | Set Interrupt Mask | $1 \Rightarrow I$ | INH | 0F | — | 2 | — | — | — | 1 | — | — | — | — | | |
| SEV | Set Overflow Flag | $1 \Rightarrow V$ | INH | 0B | — | 2 | — | — | — | — | — | — | 1 | — | | |
| STAA (opr) | Store Accumulator A | $A \Rightarrow M$ | A | DIR | 97 | dd | 3 | — | — | — | — | Δ | Δ | 0 | — | |
| | | | | EXT | B7 | hh ll | 4 | | | | | | | | | |
| | | | | IND,X | A7 | ff | 4 | | | | | | | | | |
| | | | | IND,Y | A7 | ff | 5 | | | | | | | | | |
| STAB (opr) | Store Accumulator B | $B \Rightarrow M$ | B | DIR | D7 | dd | 3 | — | — | — | — | Δ | Δ | 0 | — | |
| | | | | EXT | F7 | hh ll | 4 | | | | | | | | | |
| | | | | IND,X | E7 | ff | 4 | | | | | | | | | |
| | | | | IND,Y | E7 | ff | 5 | | | | | | | | | |
| STD (opr) | Store Accumulator D | $A \Rightarrow M, B \Rightarrow M + 1$ | | DIR | DD | dd | 4 | — | — | — | — | Δ | Δ | 0 | — | |
| | | | | EXT | FD | hh ll | 5 | | | | | | | | | |
| | | | | IND,X | ED | ff | 5 | | | | | | | | | |
| | | | | IND,Y | ED | ff | 6 | | | | | | | | | |
| STOP | Stop Internal Clocks | — | INH | CF | — | 2 | — | — | — | — | — | — | — | — | | |
| STS (opr) | Store Stack Pointer | $SP \Rightarrow M : M + 1$ | | DIR | 9F | dd | 4 | — | — | — | — | Δ | Δ | 0 | — | |
| | | | | EXT | BF | hh ll | 5 | | | | | | | | | |
| | | | | IND,X | AF | ff | 5 | | | | | | | | | |
| | | | | IND,Y | AF | ff | 6 | | | | | | | | | |
| STX (opr) | Store Index Register X | $IX \Rightarrow M : M + 1$ | | DIR | DF | dd | 4 | — | — | — | — | Δ | Δ | 0 | — | |
| | | | | EXT | FF | hh ll | 5 | | | | | | | | | |
| | | | | IND,X | EF | ff | 5 | | | | | | | | | |
| | | | | IND,Y | EF | ff | 6 | | | | | | | | | |
| STY (opr) | Store Index Register Y | $IY \Rightarrow M : M + 1$ | | DIR | 18 | DD | 5 | — | — | — | — | Δ | Δ | 0 | — | |
| | | | | EXT | 18 | FF | hh ll | | | | | | | | | 6 |
| | | | | IND,X | 1A | EF | ff | | | | | | | | | 6 |
| | | | | IND,Y | 18 | EF | ff | | | | | | | | | 6 |
| SUBA (opr) | Subtract Memory from A | $A - M \Rightarrow A$ | A | IMM | 80 | ii | 2 | — | — | — | — | Δ | Δ | Δ | Δ | |
| | | | | DIR | 90 | dd | 3 | | | | | | | | | |
| | | | | EXT | B0 | hh ll | 4 | | | | | | | | | |
| | | | | IND,X | A0 | ff | 4 | | | | | | | | | |
| | | | | IND,Y | A0 | ff | 5 | | | | | | | | | |
| SUBB (opr) | Subtract Memory from B | $B - M \Rightarrow B$ | A | IMM | C0 | ii | 2 | — | — | — | — | Δ | Δ | Δ | Δ | |
| | | | | DIR | D0 | dd | 3 | | | | | | | | | |
| | | | | EXT | F0 | hh ll | 4 | | | | | | | | | |
| | | | | IND,X | E0 | ff | 4 | | | | | | | | | |
| | | | | IND,Y | E0 | ff | 5 | | | | | | | | | |
| SUBD (opr) | Subtract Memory from D | $D - M : M + 1 \Rightarrow D$ | | IMM | 83 | jj kk | 4 | — | — | — | — | Δ | Δ | Δ | Δ | |
| | | | | DIR | 93 | dd | 5 | | | | | | | | | |
| | | | | EXT | B3 | hh ll | 6 | | | | | | | | | |
| | | | | IND,X | A3 | ff | 6 | | | | | | | | | |
| | | | | IND,Y | A3 | ff | 7 | | | | | | | | | |
| SWI | Software Interrupt | See Figure 3-2 | INH | 3F | — | 14 | — | — | — | 1 | — | — | — | — | | |
| TAB | Transfer A to B | $A \Rightarrow B$ | INH | 16 | — | 2 | — | — | — | — | Δ | Δ | 0 | — | | |
| TAP | Transfer A to CC Register | $A \Rightarrow CCR$ | INH | 06 | — | 2 | Δ | \downarrow | Δ | Δ | Δ | Δ | Δ | Δ | | |
| TBA | Transfer B to A | $B \Rightarrow A$ | INH | 17 | — | 2 | — | — | — | — | Δ | Δ | 0 | — | | |

Table 3-2. Instruction Set (Sheet 7 of 7)

| Mnemonic | Operation | Description | Addressing Mode | Instruction | | | Condition Codes | | | | | | | |
|-----------|-----------------------------|--------------------|-----------------------|----------------|-------------------|-------------|-----------------|---|---|---|---|---|---|---|
| | | | | Opcode | Operand | Cycles | S | X | H | I | N | Z | V | C |
| TEST | TEST (Only in Test Modes) | Address Bus Counts | INH | 00 | — | * | — | — | — | — | — | — | — | — |
| TPA | Transfer CC Register to A | CCR ⇒ A | INH | 07 | — | 2 | — | — | — | — | — | — | — | — |
| TST (opr) | Test for Zero or Minus | M – 0 | EXT IND,X IND,Y | 7D 6D 6D | hh ll ff ff | 6 6 7 | — | — | — | — | Δ | Δ | 0 | 0 |
| TSTA | Test A for Zero or Minus | A – 0 | A INH | 4D | — | 2 | — | — | — | — | Δ | Δ | 0 | 0 |
| TSTB | Test B for Zero or Minus | B – 0 | B INH | 5D | — | 2 | — | — | — | — | Δ | Δ | 0 | 0 |
| TSX | Transfer Stack Pointer to X | SP + 1 ⇒ IX | INH | 30 | — | 3 | — | — | — | — | — | — | — | — |
| TSY | Transfer Stack Pointer to Y | SP + 1 ⇒ IY | INH | 18 30 | — | 4 | — | — | — | — | — | — | — | — |
| TXS | Transfer X to Stack Pointer | IX – 1 ⇒ SP | INH | 35 | — | 3 | — | — | — | — | — | — | — | — |
| TYS | Transfer Y to Stack Pointer | IY – 1 ⇒ SP | INH | 18 35 | — | 4 | — | — | — | — | — | — | — | — |
| WAI | Wait for Interrupt | Stack Regs & WAIT | INH | 3E | — | ** | — | — | — | — | — | — | — | — |
| XGDY | Exchange D with X | IX ⇒ D, D ⇒ IX | INH | 8F | — | 3 | — | — | — | — | — | — | — | — |
| XGDY | Exchange D with Y | IY ⇒ D, D ⇒ IY | INH | 18 8F | — | 4 | — | — | — | — | — | — | — | — |

Cycle

* Infinity or until reset occurs

** 12 cycles are used beginning with the opcode fetch. A wait state is entered which remains in effect for an integer number of MPU E-clock cycles (n) until an interrupt is recognized. Finally, two additional cycles are used to fetch the appropriate interrupt vector (14 + n total).

Operands

- dd = 8-bit direct address (\$0000–\$00FF) (high byte assumed to be \$00)
- ff = 8-bit positive offset \$00 (0) to \$FF (255) (is added to index)
- hh = High-order byte of 16-bit extended address
- ii = One byte of immediate data
- jj = High-order byte of 16-bit immediate data
- kk = Low-order byte of 16-bit immediate data
- ll = Low-order byte of 16-bit extended address
- mm = 8-bit mask (set bits to be affected)
- rr = Signed relative offset \$80 (–128) to \$7F (+127)
(offset relative to address following machine code offset byte))

Operators

- () Contents of register shown inside parentheses
- ← Is transferred to
- ↑ Is pulled from stack
- ↓ Is pushed onto stack
- Boolean AND
- + Arithmetic addition symbol except where used as inclusive-OR symbol in Boolean formula
- ⊕ Exclusive-OR
- * Multiply
- :
- Arithmetic subtraction symbol or negation symbol (two's complement)

Condition Codes

- Bit not changed
- 0 Bit always cleared
- 1 Bit always set
- Δ Bit cleared or set, depending on operation
- ↓ Bit can be cleared, cannot become set

Section 4. Operating Modes and On-Chip Memory

4.1 Contents

| | | |
|-------|---|----|
| 4.2 | Introduction | 43 |
| 4.3 | Operating Modes | 44 |
| 4.3.1 | Bootstrap Mode | 44 |
| 4.3.2 | Special Test Mode | 44 |
| 4.3.3 | Expanded Operating Mode | 45 |
| 4.4 | Mode Selection | 46 |
| 4.5 | On-Chip Memory | 48 |
| 4.5.1 | Memory Map and Register Block | 49 |
| 4.5.2 | RAM | 56 |

4.2 Introduction

The MC68HC11ED0 microcontroller (MCU) has three modes of operation. For expanded and special test modes, there are no reset or interrupt vectors contained in on-chip resources. An external memory must be used to provide vectors at locations \$FFC0–\$FFFF. In bootstrap mode, a small on-chip read-only memory (ROM) becomes present in the memory map and provides the vectors for this mode. Refer to [Figure 4-3](#).

4.3 Operating Modes

This subsection describes the three memory modes:

- Bootstrap mode
- Special test mode
- Expanded operating mode

Refer to [Figure 4-3](#).

4.3.1 Bootstrap Mode

Bootstrap mode allows special-purpose programs to be entered into internal random-access memory (RAM). The MCU contains 256 bytes of bootstrap ROM which is enabled and present in the memory map only when the device is in bootstrap mode. The bootstrap ROM contains a small program which initializes the serial communications interface (SCI) and allows the user to download exactly 256 bytes of code into on-chip RAM. After receiving the character for address \$01FF, control passes to the loaded program at \$0100. Vectors are present on chip and located at \$BFC0–\$BFFF.

4.3.2 Special Test Mode

Special test mode is used primarily for factory testing. In this operating mode, vectors must be provided externally at \$BFC0–\$BFFF.

4.3.3 Expanded Operating Mode

In expanded operating mode, the MCU has a 64-Kbyte address range and, using the expansion bus, can access external resources within the 64-Kbyte space. This space includes:

- On-chip memory addresses
- Addressing capabilities for external peripheral and memory devices

In expanded operating mode, high-order address bits are output on ADDR[15:8] pins, low-order address bits and the data bus are multiplexed on ADDR/DATA[7:0]. Refer to [Figure 1-1. MC68HC11ED0 Block Diagram](#).

The read/write ($\overline{R/\overline{W}}$) and address strobe (AS) signals allow the low-order address and the 8-bit data bus to be time-multiplexed on the same pins.

- During the first half of each bus cycle, address information is present.
- During the second half of each bus cycle, the pins become the bidirectional data bus.

AS is an active-high latch enable signal for an external address latch. Address information is allowed through the transparent latch while AS is high and is latched when AS drives low. The address, $\overline{R/\overline{W}}$, and AS signals are active and valid for all bus cycles including accesses to internal memory locations.

The E clock is used to enable external devices to drive data onto the internal data bus during the second half of a read bus cycle (E clock high). $\overline{R/\overline{W}}$ controls the direction of data transfers. $\overline{R/\overline{W}}$ drives low when data is being written to the external data bus. $\overline{R/\overline{W}}$ will remain low during consecutive data bus write cycles, such as when a double-byte store occurs. Refer to [Figure 4-1](#) for an example of address and data multiplexing.

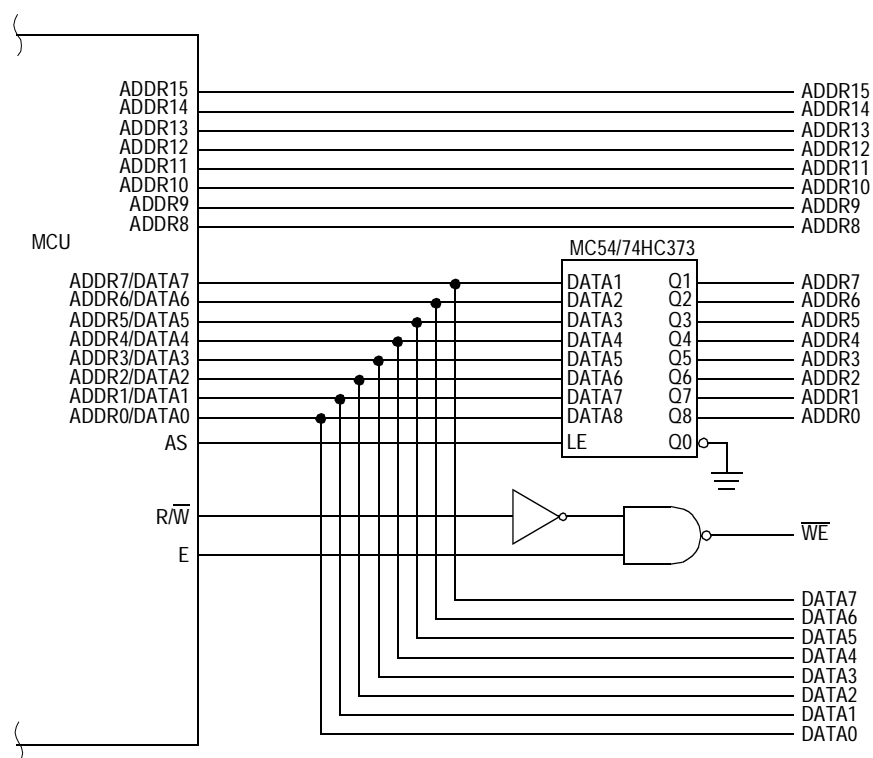


Figure 4-1. Address and Data Demultiplexing

4.4 Mode Selection

Operating modes are selected by a combination of logic levels applied to two input pins (MODA and MODB) during reset. The logic level present (at the rising edge of reset) on these inputs is reflected in bits in the highest priority I-bit interrupt and miscellaneous (HPRIO) register. After reset, the operating mode may be changed as shown in [Figure 4-2](#).

Address: \$003C

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---------------|----------------------|---------------------|--------------------|----------------------|-------|-------|-------|-------|
| Read: | RBOOT ⁽¹⁾ | SMOD ⁽¹⁾ | MDA ⁽¹⁾ | IRVNE ⁽¹⁾ | PSEL3 | PSEL2 | PSEL1 | PSEL0 |
| Write: | | | | | | | | |
| Resets: | | | | | | | | |
| Expanded: | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| Bootstrap: | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| Special Test: | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |

1. The reset values depend on the mode selected at power-up.

Figure 4-2. Highest Priority I-Bit Interrupt and Miscellaneous Register (HPRIO)

RBOOT — Read Bootstrap ROM Bit

Valid only when SMOD is set (bootstrap or special test mode); can be written only in special modes

0 = Bootloader ROM disabled and not in map

1 = Bootloader ROM enabled and in map at \$BE00–\$BFFF

SMOD and MDA — Special Mode Select and Mode Select A Bits

The initial value of SMOD is the inverse of the logic level present on the MODB pin at the rising edge of reset. The initial value of MDA equals the logic level present on the MODA pin at the rising edge of reset. These two bits can be read at any time. They can be written any time in special modes. MDA can be written only once in normal modes. SMOD cannot be set once it has been cleared.

| Input | | Mode | Latched at Reset | |
|-------|------|--|------------------|-----|
| MODB | MODA | | SMOD | MDA |
| 1 | 0 | High-impedance state ADDR/DATA (CPU held in reset) | 0 | 0 |
| 1 | 1 | Expanded | 0 | 1 |
| 0 | 0 | Bootstrap | 1 | 0 |
| 0 | 1 | Special test | 1 | 1 |

CAUTION: *Unlike other M68HC11 Family devices, the MC68HC11ED0 will not function in single-chip operating mode. If MODA is pulled low and MODB is pulled high at the rising edge of reset (the condition that causes most*

M68HC11 devices to enter single-chip mode) the CPU will remain in reset until the \overline{RESET} pin is pulled low then released with appropriate logic levels applied to MODA and MODB.

IRVNE — Internal Read Visibility/Not E Bit

IRVNE can be written once in any mode. In special test mode, IRVNE is reset to 1. In all other modes, IRVNE is reset to 0.

In expanded test modes, IRVNE determines whether internal read visibility (IRV) is on or off.

0 = No internal read visibility on external bus

1 = Data from internal reads driven out the external data bus

In bootstrap mode, IRVNE determines whether the E clock drives out from the chip.

0 = E driven out from the chip

1 = E pin driven low

| Mode | IRVNE Out of Reset | E Clock Out of Reset | IRV Out of Reset | IRVNE Affects Only | IRVNE Can Be Written |
|--------------|--------------------|----------------------|------------------|--------------------|----------------------|
| Expanded | 0 | On | Off | IRV | Once |
| Bootstrap | 0 | On | Off | E | Once |
| Special test | 1 | On | On | IRV | Once |

PSEL[3:0] — Priority Select Bits

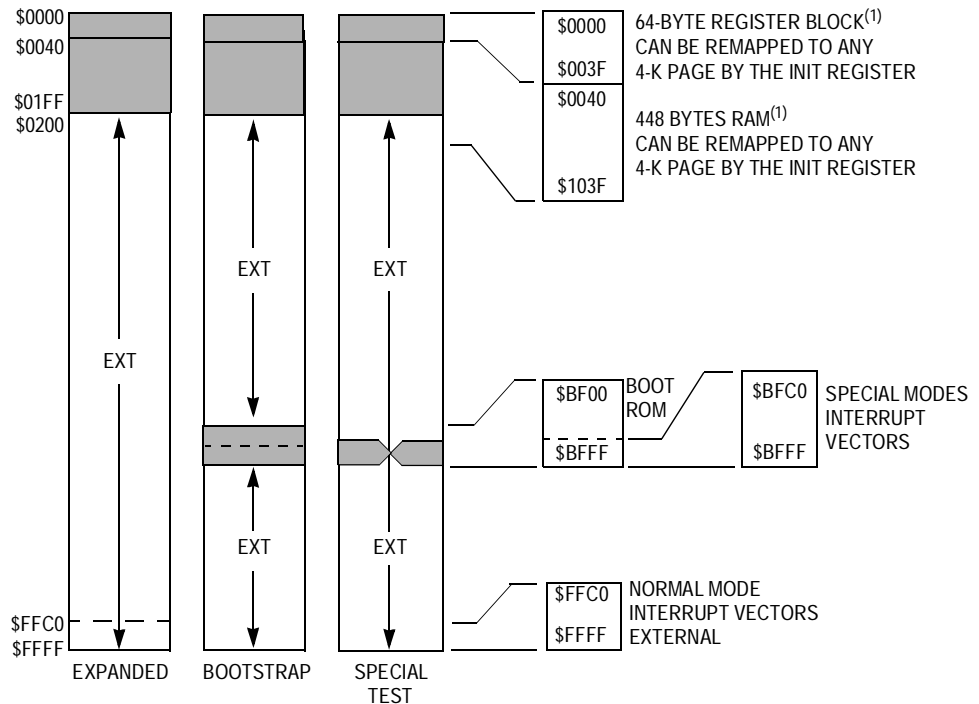
Refer to [Section 5. Resets and Interrupts](#).

4.5 On-Chip Memory

The MC68HC11ED0 contains 512 bytes of on-chip static RAM. There is no on-chip ROM. Since the MC68HC11ED0 is intended for expanded mode applications only, reset and interrupt vectors are not contained in on-chip resources. An external memory must provide these at locations \$FFC0–\$FFFF. Refer to [Figure 4-3](#).

4.5.1 Memory Map and Register Block

The INIT register controls the location of the register block and RAM in the 64-Kbyte central processor unit (CPU) address space. The 64-Kbyte register block originates at \$0000 after reset and can be placed at any 4-Kbyte boundary (\$x000) by writing an appropriate value to the INIT register. Since the RAM also begins at \$0000 after reset, 64 bytes are overlaid by the register block. Registers are a higher priority resource than RAM. Therefore, the RAM which is overlaid by registers is inaccessible. Either the registers or the RAM must be remapped to gain access to all 512 bytes of the RAM. Refer to [Figure 4-3](#) and [Figure 4-4](#).



Note 1. To access the full 512 bytes of RAM, remap either the register block or the RAM to any 4-K (\$x000) boundary.

Figure 4-3. MC68HC11ED0 Memory Map

Operating Modes and On-Chip Memory

| Addr. | Register Name | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
|--------|---|--------|--------|--------|--------|--------|--------|--------|-------|-------|
| \$0000 | Port A Data Register (PORTA) See page 66. | Read: | PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| | | Write: | | | | | | | | |
| | | Reset: | U | 0 | 0 | 0 | U | U | U | U |
| \$0001 | Reserved | R | R | R | R | R | R | R | R | |
| ↓ | | | | | | | | | | |
| \$0007 | Reserved | R | R | R | R | R | R | R | R | |
| \$0008 | Port D Data Register (PORTD) See page 68. | Read: | | | PD5 | PD4 | PD3 | PD2 | PD1 | PD0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | U | U | U | U | U | U |
| \$0009 | Port D Data Direction Register (DDRD) See page 69. | Read: | | | DDD5 | DDD4 | DDD3 | DDD2 | DDD1 | DDD0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$000A | Reserved | R | R | R | R | R | R | R | R | |
| \$000B | Timer Compare Force Register (CFORC) See page 91. | Read: | FOC1 | FOC2 | FOC3 | FOC4 | FOC5 | | | |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$000C | Output Compare 1 Mask Register (OC1M) See page 92. | Read: | OC1M7 | OC1M6 | OC1M5 | OC1M4 | OC1M3 | | | |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$000D | Output Compare 1 Data Register (OC1D) See page 92. | Read: | OC1D7 | OC1D6 | OC1D5 | OC1D4 | OC1D3 | | | |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$000E | Timer Count Register High (TCNT) See page 93. | Read: | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented
 R = Reserved
 U = Unaffected

Figure 4-4. Register and Control Bit Assignments (Sheet 1 of 6)

| Addr. | Register Name | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
|--------|---|--------|---------------------|--------|--------|--------|--------|--------|-------|-------|
| \$000F | Timer Count Register Low (TCNT) See page 93. | Read: | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0010 | Timer Input Capture Register 1 High (TIC1) See page 94. | Read: | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| \$0011 | Timer Input Capture Register 1 Low (TIC1) See page 94. | Read: | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| \$0012 | Timer Input Capture Register 2 High (TIC2) See page 94. | Read: | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| \$0013 | Timer Input Capture Register 2 Low (TIC2) See page 94. | Read: | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| \$0014 | Timer Input Capture Register 3 High (TIC3) See page 94. | Read: | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| \$0015 | Timer Input Capture Register 3 Low (TIC3) See page 94. | Read: | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| \$0016 | Timer Output Compare Register 1 High (TOC1) See page 95. | Read: | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$0017 | Timer Output Compare Register 1 Low (TOC1) See page 95. | Read: | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

= Unimplemented
 R = Reserved
 U = Unaffected

Figure 4-4. Register and Control Bit Assignments (Sheet 2 of 6)

Operating Modes and On-Chip Memory

| Addr. | Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|---|--------|--------|--------|--------|--------|--------|--------|-------|-------|
| \$0018 | Timer Output Compare Register 2 High (TOC2) See page 95. | Read: | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$0019 | Timer Output Compare Register 2 Low (TOC2) See page 95. | Read: | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$001A | Timer Output Compare Register 3 High (TOC3) See page 95. | Read: | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$001B | Timer Output Compare Register 3 Low (TOC3) See page 95. | Read: | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$001C | Timer Output Compare Register 4 High (TOC4) See page 96. | Read: | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$001D | Timer Output Compare Register 4 Low (TOC4) See page 96. | Read: | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$001E | Timer Input Capture4/Output Compare 5 High (TI4/O5) See page 96. | Read: | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$001F | Timer Input Capture4/Output Compare 5 Low (TI4/O5) See page 96. | Read: | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| \$0020 | Timer Control Register 1 (TCTL1) See page 97. | Read: | OM2 | OL2 | OM3 | OL3 | OM4 | OL4 | OM5 | OL5 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented
 R = Reserved
 U = Unaffected

Figure 4-4. Register and Control Bit Assignments (Sheet 3 of 6)

| Addr. | Register Name | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
|--------|---|--------|---------------------|-------|-------|-------|--------|-------|-------|-------|
| \$0021 | Timer Control Register 2 (TCTL2) See page 98. | Read: | EDG4B | EDG4A | EDG1B | EDG1A | EDG2B | EDG2A | EDG3B | EDG3A |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0022 | Timer Interrupt Mask 1 Register (TMSK1) See page 99. | Read: | OC1I | OC2I | OC3I | OC4I | I4/O5I | IC1I | IC2I | IC3I |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0023 | Timer Interrupt Flag 1 Register (TFLG1) See page 100. | Read: | OC1F | OC2F | OC3F | OC4F | I4/O5F | IC1F | IC2F | IC3F |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0024 | Timer Interrupt Mask 2 Register (TMSK2) See page 100. | Read: | TOI | RTII | PAOVI | PAII | | | PR1 | PR0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0025 | Timer Interrupt Flag 2 Register (TFLG2) See page 102. | Read: | TOF | RTIF | PAOVF | PAIF | | | | |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0026 | Pulse Accumulator Control Register (PACTL) See page 104. | Read: | DDRA7 | PAEN | PAMOD | PEDGE | DDRA3 | I4/O5 | RTR1 | RTR0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$0027 | Pulse Accumulator Counter Register (PACNT) See page 106. | Read: | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| \$0028 | Serial Peripheral Control Register (SPCR) See page 83. | Read: | SPIE | SPE | DWOM | MSTR | CPOL | CPHA | SPR1 | SPR0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| \$0029 | Serial Peripheral Status Register (SPSR) See page 85. | Read: | SPIF | WCOL | | MODF | | | | |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

= Unimplemented
 = Reserved
U = Unaffected

Figure 4-4. Register and Control Bit Assignments (Sheet 4 of 6)

Operating Modes and On-Chip Memory

| Addr. | Register Name | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
|--------|--|--------|---------------------------|-------|---------------------|--------------------|-------|-------|--------------------|--------------------|
| \$002A | SPI Data Register I/O (SPDR) See page 86. | Read: | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | Write: | | | | | | | | |
| | | Reset: | Indeterminate after reset | | | | | | | |
| \$002B | Baud Rate Register (BAUD) See page 74. | Read: | TCLR | | SCP1 | SCP0 | RCKB | SCR2 | SCR1 | SCR0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | U | U | U |
| \$002C | SCI Control Register 1 (SCCR1) See page 77. | Read: | R8 | T8 | | M | WAKE | | | |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$002D | SCI Control Register 2 (SCCR2) See page 78. | Read: | TIE | TCIE | RIE | ILIE | TE | RE | RWU | SBK |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$002E | SCI Status Register (SCSR) See page 79. | Read: | TDRE | TC | RDRF | IDLE | OR | NF | FE | |
| | | Write: | | | | | | | | |
| | | Reset: | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$002F | SCI Data Register (SCDR) See page 80. | Read: | R7/T7 | R6/T6 | R5/T5 | R4/T4 | R3/T3 | R2/T2 | R1/T1 | R0/T0 |
| | | Write: | | | | | | | | |
| | | Reset: | Unaffected by reset | | | | | | | |
| \$0030 | Reserved | R | R | R | R | R | R | R | R | |
| ↓ | | | | | | | | | | |
| \$0038 | Reserved | R | R | R | R | R | R | R | R | |
| \$0039 | System Configuration Options Register (OPTION) See page 60. | Read: | | | IRQE ⁽¹⁾ | DLY ⁽¹⁾ | CME | | CR1 ⁽¹⁾ | CR0 ⁽¹⁾ |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

Note 1. Can be written only once in the first 64 cycles out of reset in normal modes or at any time in special modes.


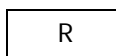
 = Unimplemented  = Reserved U = Unaffected

Figure 4-4. Register and Control Bit Assignments (Sheet 5 of 6)

| Addr. | Register Name | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 | |
|---|--|--------|-------------------|----------------|----------------|----------------------|-------|-------|-------|-------|
| \$003A | Arm/Reset COP Timer Circuitry Register (COPRST) See page 61. | Read: | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$003B | Reserved | R | R | R | R | R | R | R | R | |
| \$003C | Highest Priority I-Bit Interrupt and Miscellaneous Register (HPRIO) See page 63. | Read: | RBOOT | SMOD | MDA | IRVNE ⁽¹⁾ | PSEL3 | PSEL2 | PSEL1 | PSEL0 |
| | | Write: | | | | | | | | |
| | | Reset: | Note 1 | Note 1 | Note 1 | U | 0 | 1 | 0 | 1 |
| Note 1. RBOOT, SMOD, and MDA reset depends on power-up initialization mode and can be written only in special mode. | | | | | | | | | | |
| \$003D | RAM and Register Mapping Register (INIT) See page 56. | Read: | RAM3 | RAM2 | RAM1 | RAM0 | REG3 | REG2 | REG1 | REG0 |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$003E | Reserved | R | R | R | R | R | R | R | R | |
| \$003F | System Configuration Register (CONFIG) See page 62. | Read: | | | | | | NOCOP | | |
| | | Write: | | | | | | | | |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | ■ = Unimplemented | □ R = Reserved | U = Unaffected | | | | | |

Figure 4-4. Register and Control Bit Assignments (Sheet 6 of 6)

4.5.2 RAM

The MC68HC11ED0 has 512 bytes of on-chip static RAM. The RAM can be mapped to any 4-Kbyte boundary. Upon reset, the RAM is mapped at \$0000–\$01FF. The register block also begins at \$0000 and overlaps the RAM space. Since registers have priority over RAM, this causes 64 bytes of RAM to be lost. However, the user can map either the RAM or the register block to any 4-Kbyte boundary (\$x000) and access the full 512 bytes of RAM. Remapping is accomplished by writing appropriate values to the INIT register.

When power is removed from the MCU, RAM contents may be preserved using the MODB/ V_{STBY} pin. A 4-volt nominal power source applied to this pin protects all 512 bytes of RAM.

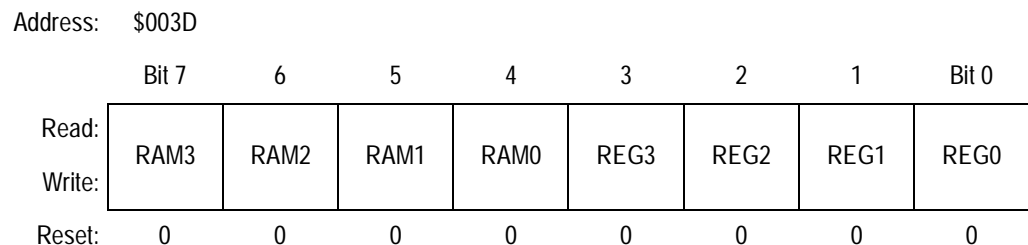


Figure 4-5. RAM and Register Mapping Register (INIT)

NOTE: *INIT can be written only once in the first 64-cycles out of reset in normal modes or at any time in special modes.*

RAM[3:0] — Internal RAM Map Position Bits

These bits determine the upper four bits of the RAM address. At reset RAM is mapped to \$0000 and includes the register block. Refer to [Figure 4-3](#).

REG[3:0] — 128-Byte Register Block Map Position Bits

These bits determine the upper four bits of the register space address. At reset registers are mapped to \$0000 and overwrite the first 64 bytes of RAM. Refer to [Figure 4-3](#).

Section 5. Resets and Interrupts

5.1 Contents

| | | |
|-----|--|----|
| 5.2 | Introduction | 57 |
| 5.3 | Resets | 57 |
| 5.4 | System Configuration Options Register | 60 |
| 5.5 | Arm/Reset COP Timer Circuitry Register | 61 |
| 5.6 | Configuration Control Register | 62 |
| 5.7 | Highest Priority I-Bit Interrupt and Miscellaneous Register | 63 |

5.2 Introduction

This section describes the MC68HC11ED0 reset and interrupt structure.

5.3 Resets

The MC68HC11ED0 has three reset vectors and 18 interrupt vectors. The reset vectors are:

- $\overline{\text{RESET}}$ or power-on reset
- Clock monitor fail
- Computer operating properly (COP) failure

The 18 interrupt vectors service 22 interrupt sources (three non-maskable, 19 maskable). The three non-maskable interrupt sources are:

- $\overline{\text{XIRQ}}$ pin (X-bit interrupt)
- Illegal opcode trap
- Software interrupt

On-chip peripheral systems generate maskable interrupts, which are recognized only if the global interrupt mask bit (I) in the condition code register (CCR) is clear. Maskable interrupts are prioritized according to a default arrangement; however, any one source can be elevated to the highest maskable priority position by a software-accessible control register (HPRIO). HPRIO can be written at any time, provided bit I in the CCR is set.

Nineteen interrupt sources in the MC68HC11ED0 are subject to masking by the global interrupt mask bit (bit I in the CCR). In addition to the global I bit, all of these sources, except the external interrupt ($\overline{\text{IRQ}}$) pin, are controlled by local enable bits in the control registers. Most interrupt sources in M68HC11 devices have separate interrupt vectors; therefore there is usually no need for software to poll control registers to determine the cause of an interrupt.

For some interrupt sources, such as the serial communications interface (SCI) interrupts, the flags are automatically cleared during the normal course of responding to the interrupt requests. For example, the RDRF flag in the SCI system is cleared by the automatic clearing mechanism invoked by a read of the SCI status register while RDRF is set, followed by a read of the SCI data register. The normal response to an RDRF interrupt request would be to read the SCI status register to check for receive errors, then to read the received data from the SCI data register. These two steps satisfy the automatic clearing mechanism without requiring any special instructions.

Refer to [Table 5-1](#) for interrupt and reset vector assignments.

Table 5-1. Interrupt and Reset Vector Assignments

| Vector Address | Interrupt Source | CCR Mask Bit | Local Mask | Priority (1 = High) |
|---------------------------------|--|--------------|-----------------------------------|----------------------------|
| \$FFC0, \$FFC1 – \$FFD4, \$FFD5 | Reserved | — | — | — |
| \$FFD6, \$FFD7 | SCI serial system: <ul style="list-style-type: none"> • SCI receive data register full • SCI receiver overrun • SCI transmit data register empty • SCI transmit complete • SCI idle line detect | I | RIE RIE TIE TCIE ILIE | 19 20 21 22 23 |
| \$FFD8, \$FFD9 | SPI serial transfer complete | I | SPIE | 18 |
| \$FFDA, \$FFDB | Pulse accumulator input edge | I | PAII | 17 |
| \$FFDC, \$FFDD | Pulse accumulator overflow | I | PAOVI | 16 |
| \$FFDE, \$FFDF | Timer overflow | I | TOI | 15 |
| \$FFE0, \$FFE1 | Timer input capture 4/output compare 5 | I | I4/O5I | 14 |
| \$FFE2, \$FFE3 | Timer output compare 4 | I | OC4I | 13 |
| \$FFE4, \$FFE5 | Timer output compare 3 | I | OC3I | 12 |
| \$FFE6, \$FFE7 | Timer output compare 2 | I | OC2I | 11 |
| \$FFE8, \$FFE9 | Timer output compare 1 | I | OC1I | 10 |
| \$FFEA, \$FFEB | Timer input capture 3 | I | IC3I | 9 |
| \$FFEC, \$FFED | Timer input capture 2 | I | IC2I | 8 |
| \$FFEE, \$FFEF | Timer input capture 1 | I | IC1I | 7 |
| \$FFF0, \$FFF1 | Real-time interrupt | I | RTII | 6 |
| \$FFF2, \$FFF3 | $\overline{\text{IRQ}}$ (external pin) | I | None | 5 |
| \$FFF4, \$FFF5 | $\overline{\text{XIRQ}}$ pin | X | None | 4 |
| \$FFF6, \$FFF7 | Software interrupt | None | None | Note 1 |
| \$FFF8, \$FFF9 | Illegal opcode trap | None | None | Note 1 |
| \$FFFA, \$FFFB | COP failure | None | NOCOP | 3 |
| \$FFFC, \$FFFD | Clock monitor fail | None | CME | 2 |
| \$FFFE, \$FFFF | $\overline{\text{RESET}}$ | None | None | 1 |

1. Same level as an instruction

5.4 System Configuration Options Register

Refer to [Figure 5-1](#) for a description of the system configuration options register (OPTION)

Address: \$0039

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|---|---------------------|--------------------|-----|---|--------------------|--------------------|
| Read: | | | IRQE ⁽¹⁾ | DLY ⁽¹⁾ | CME | | CR1 ⁽¹⁾ | CR0 ⁽¹⁾ |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

Note 1. Can be written only once in the first 64 cycles out of reset in normal modes or at any time

in special modes.


 = Unimplemented

Figure 5-1. System Configuration Options Register (OPTION)

Bits [7:6] — Unimplemented

Always read as 0

IRQE — $\overline{\text{IRQ}}$ Select Edge Sensitive Only Bit

0 = Low level recognition

1 = Falling edge recognition

DLY — Enable Oscillator Startup Delay on Exit from Stop Mode Bit

0 = No stabilization delay on exit from stop mode

1 = Stabilization delay enabled on exit from stop mode

CME — Clock Monitor Enable Bit

0 = Clock monitor disabled; slow clocks can be used

1 = Slow or stopped clocks cause clock failure reset.

Bit 2 — Unimplemented

Always reads 0

CR[1:0] — COP Timer Rate Select Bit

Refer to description of the NOCOP bit in [5.6 Configuration Control Register](#).

Table 5-2. COP Timer Rate Select

| CR[1:0] | Rate Selected | XTAL = 4.0 MHz Timeout – 0 ms, + 32.8 ms | XTAL = 8.0 MHz Timeout – 0 ms, + 16.4 ms | XTAL = 12.0 MHz Timeout – 0 ms, + 10.9 ms |
|---------|-----------------|---|---|--|
| 0 0 | $2^{15} \div E$ | 32.768 ms | 16.384 ms | 10.923 ms |
| 0 1 | $2^{17} \div E$ | 131.07 ms | 65.536 ms | 43.691 ms |
| 1 0 | $2^{19} \div E$ | 524.29 ms | 262.14 ms | 174.76 ms |
| 1 1 | $2^{21} \div E$ | 2.1 s | 1.049 s | 699.05 ms |
| | E = | 1.0 MHz | 2.0 MHz | 3.0 MHz |

5.5 Arm/Reset COP Timer Circuitry Register

Refer to [Figure 5-2](#) for a description of the arm/reset COP timer circuitry register (COPRST).

Address: \$003A

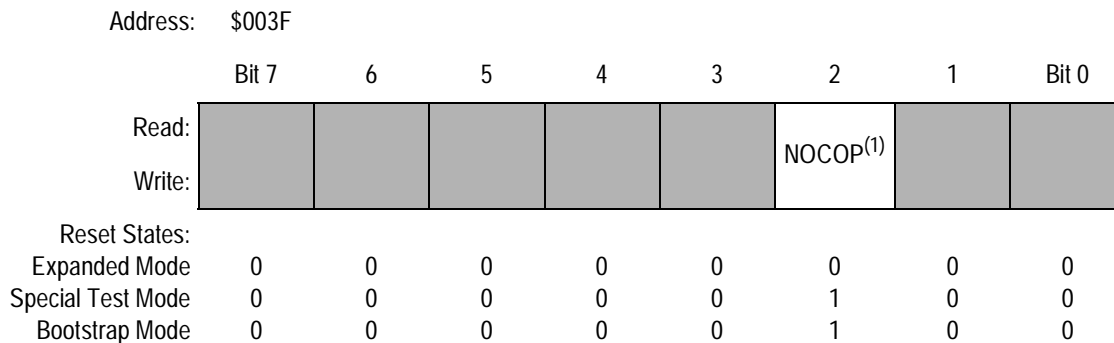
| | | | | | | | | |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Read: | Bit 7 | Bit 6 | Bit 6 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 5-2. Arm/Reset COP Timer Circuitry Register (COPRST)

Write \$55 to COPRST to arm COP watchdog clearing mechanism. Write \$AA to COPRST to reset COP watchdog. Refer to description of the NOCOP bit in [5.6 Configuration Control Register](#).

5.6 Configuration Control Register

In many M68HC11 devices the configuration control register (CONFIG) is used to define various system functions. In the MC68HC11ED0, CONFIG controls only one microcontroller (MCU) function. The NOCOP bit disables the COP watchdog circuit when it is set. Refer to [Table 5-2](#) and [Figure 5-3](#).



1. NOCOP must be written during the first 64 cycles after reset in normal modes (SMOD = 0) or at any time in special modes (SMOD = 1).


 = Unimplemented

Figure 5-3. System Configuration Register (CONFIG)

Bits [7:3] — Unimplemented

Always read as 0

NOCOP — COP System Disable Bit

Resets to programmed value

0 = COP enabled (forces reset on timeout)

1 = COP disabled (does not force reset on timeout)

Bits [1:0] — Unimplemented

Always read as 0

5.7 Highest Priority I-Bit Interrupt and Miscellaneous Register

Address: \$003C

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|----------------------|---------------------|--------------------|-------|-------|-------|-------|-------|
| Read: | RBOOT ⁽¹⁾ | SMOD ⁽¹⁾ | MDA ⁽¹⁾ | IRVNE | PSEL3 | PSEL2 | PSEL1 | PSEL0 |
| Write: | | | | | | | | |
| Reset: | U | U | U | U | 0 | 1 | 0 | 1 |

1. RBOOT, SMOD, and MDA reset depends on power-up initialization mode and can only be written only in special mode.

U = Undefined

Figure 5-4. Highest Priority I-Bit Interrupt and Miscellaneous Register (HPRIO)

RBOOT — Read Bootstrap ROM Bit

Refer to [Section 4. Operating Modes and On-Chip Memory](#).

SMOD — Special Mode Select Bit

Refer to [Section 4. Operating Modes and On-Chip Memory](#).

MDA — Mode Select A Bit

Refer to [Section 4. Operating Modes and On-Chip Memory](#).

IRVNE — Internal Read Visibility/Not E Bit

Refer to [Section 4. Operating Modes and On-Chip Memory](#).

PSEL[3:0] — Priority Select Bits

Can be written only while the I bit in the CCR is set (interrupts disabled). These bits select one interrupt source to be elevated above all other I bit related sources. See [Table 5-3](#).

Table 5-3. Highest Priority Interrupt Selection

| PSEL[3:0] | Interrupt Source Promoted |
|-----------|--|
| 0 0 0 0 | Timer overflow |
| 0 0 0 1 | Pulse accumulator overflow |
| 0 0 1 0 | Pulse accumulator input edge |
| 0 0 1 1 | SPI serial transfer complete |
| 0 1 0 0 | SCI serial system |
| 0 1 0 1 | Reserved (default to $\overline{\text{IRQ}}$) |
| 0 1 1 0 | $\overline{\text{IRQ}}$ (external pin) |
| 0 1 1 1 | Real-time interrupt |
| 1 0 0 0 | Timer input capture 1 |
| 1 0 0 1 | Timer input capture 2 |
| 1 0 1 0 | Timer input capture 3 |
| 1 0 1 1 | Timer output compare 1 |
| 1 1 0 0 | Timer output compare 2 |
| 1 1 0 1 | Timer output compare 3 |
| 1 1 1 0 | Timer output compare 4 |
| 1 1 1 1 | Timer input capture 4/output compare 5 |

Section 6. Parallel Input/Output (I/O) Ports

6.1 Contents

| | | |
|-----|--|----|
| 6.2 | Introduction | 65 |
| 6.3 | Port A Data Register | 66 |
| 6.4 | Pulse Accumulator Control Register | 67 |
| 6.5 | Port D Data Register | 68 |
| 6.6 | Port D Data Direction Register | 69 |

6.2 Introduction

The MC68HC11ED0 has up to 14 input/output (I/O) lines. The address/data bus of this microcontroller (MCU) is multiplexed and has no I/O ports associated with it. **Table 6-1** provides a summary of the configuration and features of each port.

Table 6-1. Input/Output Ports

| Port | Input Pins | Output Pins | Bidirectional Pins | Shared Functions |
|--------|------------|-------------|--------------------|---|
| Port A | 3 | 3 | 2 | Timer |
| Port D | — | — | 6 | Serial communications interface (SCI) and serial peripheral interface (SPI) |

NOTE: Do not confuse pin function with the electrical state of the pin at reset. All general-purpose I/O pins configured as inputs at reset are in a high-impedance state and the contents of port data registers is undefined. In port descriptions, a U indicates this condition. The pin function is mode dependent.

6.3 Port A Data Register

Refer to [Figure 6-1](#) for a description of the port A data register (PORTA).

Address: \$0000

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---------------------|-------|-----|-----|-----|---------|-----|-----|-------|
| Read: | PA7 | PA6 | PA5 | PA4 | PA3 | PA2 | PA1 | PA0 |
| Write: | | | | | | | | |
| Reset: | U | 0 | 0 | 0 | U | U | U | U |
| Alternate Function: | PAI | OC2 | OC3 | OC4 | IC4/OC5 | IC1 | IC2 | IC3 |
| And/or: | OC1 | OC1 | OC1 | OC1 | OC1 | — | — | — |

U = Undefined

Figure 6-1. Port A Data Register (PORTA)

To enable PA3 as fourth input capture, set I4/O5 bit in the pulse accumulator control register (PACTL). Otherwise, PA3 is configured as a fifth output compare out of reset, with bit I4/O5 being cleared. If the DDRA3 bit in PACTL is set (configuring PA3 as an output), and IC4 is enabled, writes to PA3 cause edges on the pin to result in input captures. Writing to I4/O5 has no effect when the I4/O5 register is acting as IC4. PA7 drives the pulse accumulator input but also can be configured for general-purpose I/O or output compare. DDRA7 bit in PACTL configures PA7 for either input or output.

NOTE: *Even when PA7 is configured as an output, the pin still drives the pulse accumulator input.*

6.4 Pulse Accumulator Control Register

Refer to [Figure 6-2](#) for a description of the pulse accumulator control register (PACTL).

Address: \$0026

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|------|-------|-------|-------|-------|------|-------|
| Read: | DDRA7 | PAEN | PAMOD | PEDGE | DDRA3 | I4/O5 | RTR1 | RTR0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 6-2. Pulse Accumulator Control Register (PACTL)

DDRA7 — Data Direction for Port A Bit 7

- 0 = Input
- 1 = Output

PAEN — Pulse Accumulator System Enable Bit

Refer to [9.4 Pulse Accumulator](#).

PAMOD — Pulse Accumulator Mode Bit

Refer to [9.4 Pulse Accumulator](#).

PEDGE — Pulse Accumulator Edge Control Bit

Refer to [9.4 Pulse Accumulator](#).

DDRA3 — Data Direction for Port A Bit 3

This bit is overridden if an output compare function is configured to control the PA3 pin.

- 0 = Input
- 1 = Output

I4/O5 — Input Capture 4/Output Compare 5 Bit

Refer to [Section 9. Timing System](#).

RTR[1:0] — Real-Time Interrupt (RTI) Rate Select Bits

Refer to [9.4 Pulse Accumulator](#).

6.5 Port D Data Register

Refer to [Figure 6-3](#) for a description of the port D data register (PORTD).

Address: \$0008

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|---------------------|-------|---|-----------------|-----|------|------|-----|-------|
| Read: | | | PD5 | PD4 | PD3 | PD2 | PD1 | PD0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | U | U | U | U | U | U |
| Alternate Function: | — | — | \overline{SS} | SCK | MOSI | MISO | TxD | RxD |

= Unimplemented U = Undefined

Figure 6-3. Port D Data Register (PORTD)

6.6 Port D Data Direction Register

Refer to [Figure 6-4](#) for a description of the port D data direction register (DDRD)

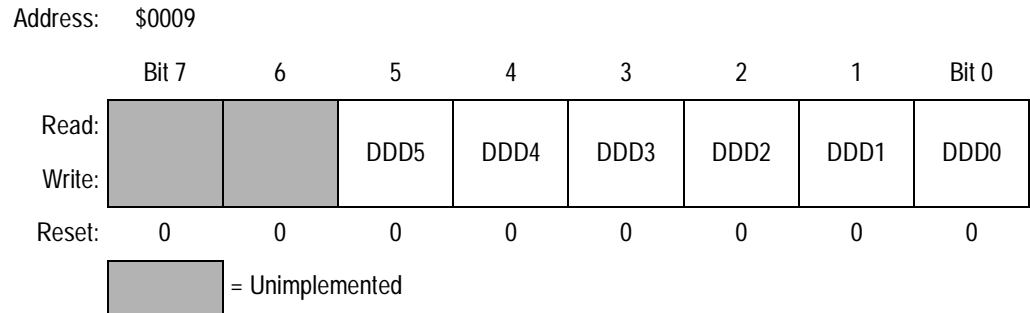


Figure 6-4. Port D Data Direction Register (DDRD)

Bits [7:6] — Unimplemented

Always read 0

DDD[5:0] — Port D Data Direction Bits

0 = Input

1 = Output

NOTE: When the serial peripheral interface (SPI) system is in slave mode, DDD5 has no meaning or effect. When the SPI system is in master mode, DDD5 determines whether bit 5 of PORTD is an error detect input (DDD5 = 0) or a general-purpose output (DDD5 = 1). If the SPI system is enabled and expects any of bits [4:2] to be an input, that bit will be an input regardless of the state of the associated DDR bit. If any of bits [4:2] are expected to be outputs, that bit will be an output **only** if the associated DDR bit is set.

Section 7. Serial Communications Interface (SCI)

7.1 Contents

| | | |
|-------|--|----|
| 7.2 | Introduction | 71 |
| 7.3 | SCI Registers | 74 |
| 7.3.1 | Baud Rate Register | 74 |
| 7.3.2 | Serial Communications Control Register 1 | 77 |
| 7.3.3 | Serial Communications Control Register 2 | 78 |
| 7.3.4 | Serial Communication Status Register | 79 |
| 7.3.5 | Serial Communications Data Register | 80 |

7.2 Introduction

The serial communications interface (SCI) is a universal asynchronous receiver transmitter (UART), one of two independent serial input/output (I/O) subsystems in the MC68HC11ED0. It has a standard non-return to zero (NRZ) format (one start bit, eight or nine data bits, and one stop bit) and several baud rates available. The SCI transmitter and receiver are independent, but use the same data format and bit rate.

Refer to [Figure 7-1](#) and [Figure 7-2](#).

Serial Communications Interface (SCI)

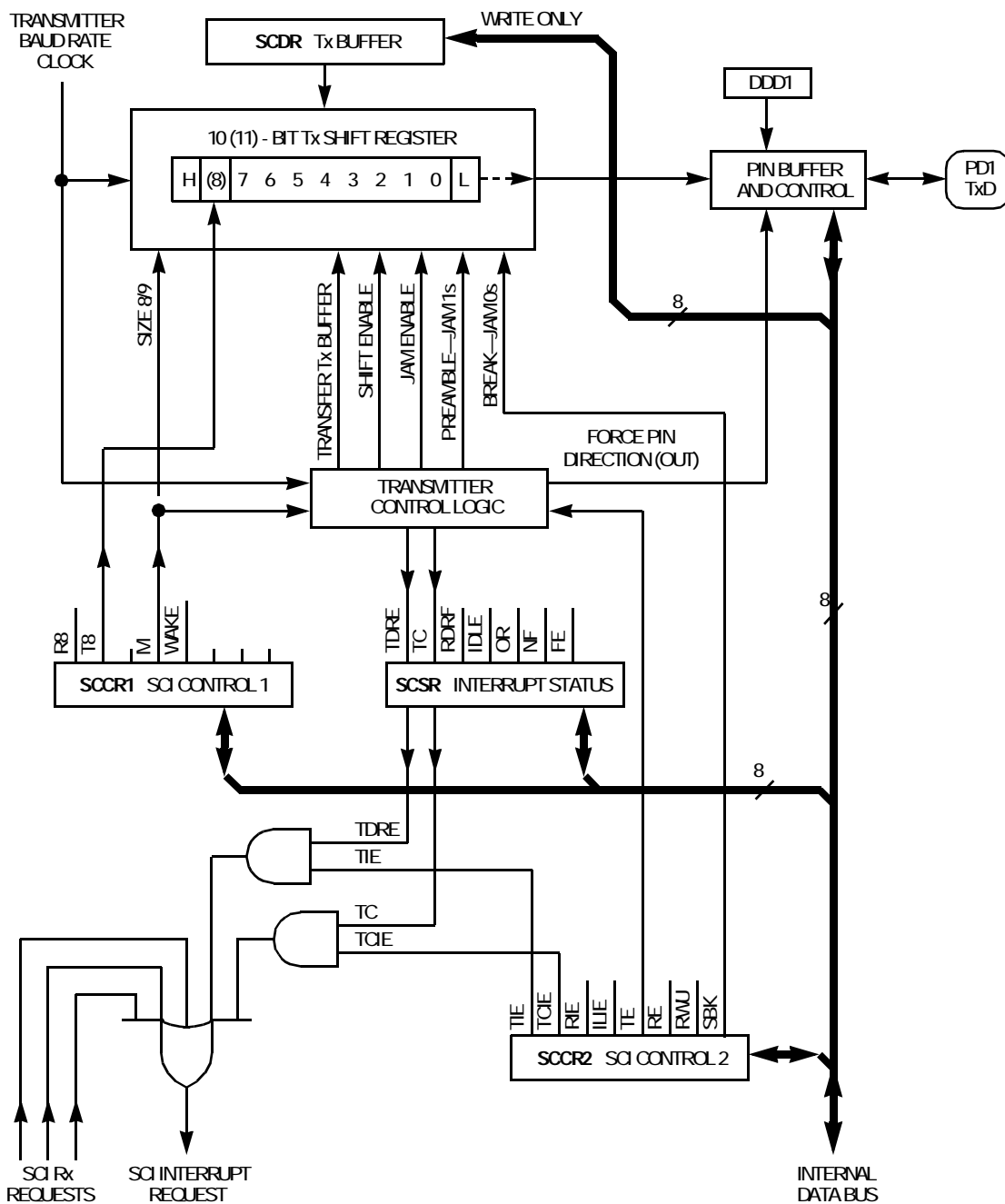


Figure 7-1. SCI Transmitter Block Diagram

7.3 SCI Registers

This subsection describes the SCI registers.

7.3.1 Baud Rate Register

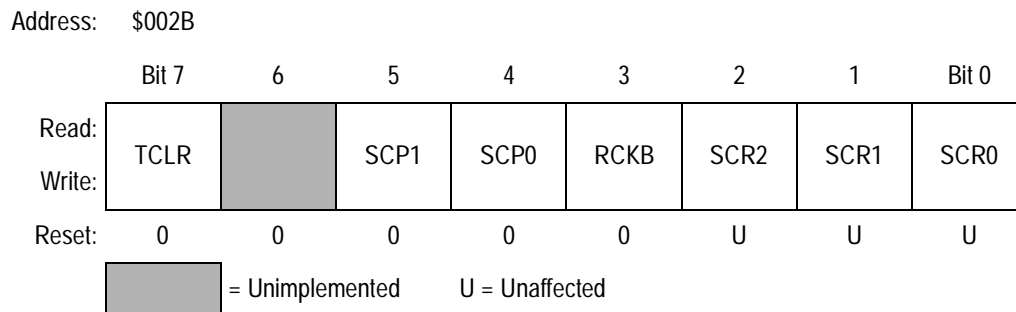


Figure 7-3. Baud Rate Register (BAUD)

TCLR — Clear Baud Rate Counter Bit

TCLR can be set only in test modes.

1 = Clear baud rate counter chain for testing purposes

0 = Normal SCI operation

SCP[1:0] — SCI Baud Rate Prescaler Select Bits

Refer to [Table 7-1](#) for the prescaler rates. The shaded boxes contain the prescaler rates used in the [Table 7-2](#).

Table 7-1. Prescaler Rates

| SCP[1:0] | Divide E Clock By | Crystal Frequency in MHz | | |
|----------|----------------------|--------------------------|-------------------|--------------------|
| | | 4.0 MHz (Baud) | 8.0 MHz (Baud) | 12.0 MHz (Baud) |
| 0 0 | 1 | 62.50 k | 125.0 k | 187.5 k |
| 0 1 | 3 | 20.83 k | 41.67 k | 62.5 k |
| 1 0 | 4 | 15.625 k | 31.25 k | 46.88 k |
| 1 1 | 13 | 4800 | 9600 | 14.4 k |

RCKB — SCI Baud Rate Clock Check Bit

RCKB can be set only in test modes.

1 = Exclusive-OR of the RT clock driven out TxD pin for testing purposes

0 = Normal SCI operation

SCR[2:0] — SCI Baud Rate Select Bits

These bits select receiver and transmitter bit rates based on output from the baud rate prescaler stage. Refer to [Table 7-2](#) and [Figure 7-4](#).

Table 7-2. Baud Rates

| SCR[2:0] | Divide Prescaler By | Baud Rate (Prescaler output from Table 7-1) | | |
|----------|---------------------|---|------|--------|
| | | 4800 | 9600 | 14.4 k |
| 0 0 0 | 1 | 4800 | 9600 | 14.4 k |
| 0 0 1 | 2 | 2400 | 4800 | 7200 |
| 0 1 0 | 4 | 1200 | 2400 | 3600 |
| 0 1 1 | 8 | 600 | 1200 | 1800 |
| 1 0 0 | 16 | 300 | 600 | 1200 |
| 1 0 1 | 32 | 150 | 300 | 450 |
| 1 1 0 | 64 | 75 | 150 | 225 |
| 1 1 1 | 128 | 37.5 | 75 | 112.5 |

Serial Communications Interface (SCI)

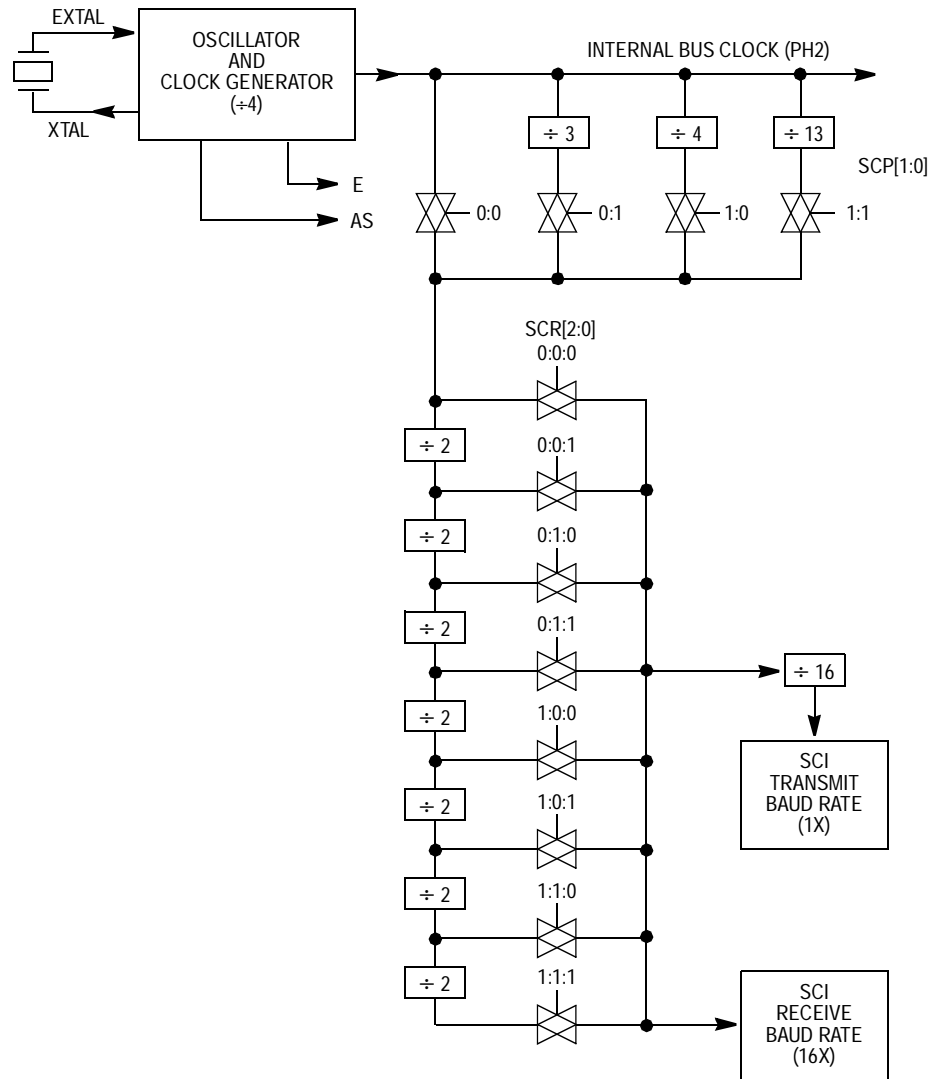


Figure 7-4. SCI Baud Rate Generator Clock Diagram

7.3.2 Serial Communications Control Register 1

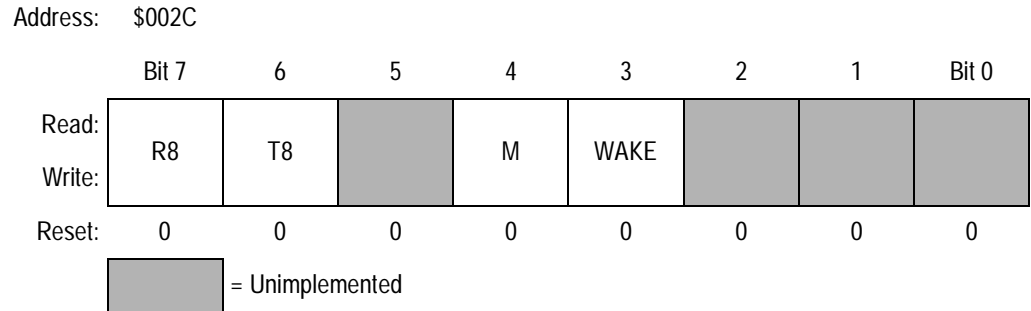


Figure 7-5. Serial Communications Control Register 1 (SCCR1)

R8 — Receive Data Bit 8

If M bit is set, R8 stores the ninth bit in the receive data character.

T8 — Transmit Data Bit 8

If M bit is set, T8 stores the ninth bit in the transmit data character.

Bit 5 — Unimplemented

Always reads 0

M — Mode Bit (select character format)

1 = Start bit, 8 data bits, 1 stop bit

1 = Start bit, 9 data bits, 1 stop bit

WAKE — Wakeup by Address Mark/Idle Bit

0 = Wakeup by IDLE line recognition

1 = Wakeup by address mark (most significant data bit set)

Bits [2:0] — Unimplemented

Always read 0

7.3.3 Serial Communications Control Register 2

Address: \$002D

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|------|-----|------|----|----|-----|-------|
| Read: | TIE | TCIE | RIE | ILIE | TE | RE | RWU | SBK |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 7-6. Serial Communications Control Register 2 (SCCR2)

TIE — Transmit Interrupt Enable Bit

0 = TDRE interrupts disabled

1 = SCI interrupt requested when TDRE status flag is set

TCIE — Transmit Complete Interrupt Enable Bit

0 = TC interrupts disabled

1 = SCI interrupt requested when TC status flag is set

RIE — Receiver Interrupt Enable Bit

0 = RDRF and OR interrupts disabled

1 = SCI interrupt requested when RDRF flag or the OR status flag is set

ILIE — Idle-Line Interrupt Enable Bit

0 = IDLE interrupts disabled

1 = SCI interrupt requested when IDLE status flag is set

TE — Transmitter Enable Bit

0 = Transmitter disabled

1 = Transmitter enabled

RE — Receiver Enable Bit

0 = Receiver disabled

1 = Receiver enabled

RWU — Receiver Wakeup Control Bit

0 = Normal SCI receiver

1 = Wakeup enabled and receiver interrupts inhibited

SBK — Send Break Bit

0 = Break generator off

1 = Break codes generated as long as SBK = 1

7.3.4 Serial Communication Status Register

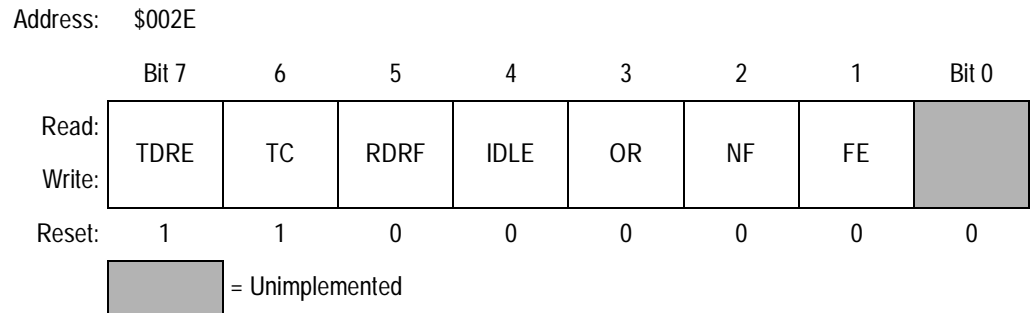


Figure 7-7. Serial Communications Status Register (SCSR)

TDRE — Transmit Data Register Empty Flag

This flag is set when SCDR is empty. Clear the TDRE flag by reading SCSR and then writing to SCDR.

- 0 = SCDR busy
- 1 = SCDR empty

TC — Transmit Complete Flag

This flag is set when the transmitter is idle (no data, preamble, or break transmission in progress). Clear the TC flag by reading SCSR and then writing to SCDR.

- 0 = Transmitter busy
- 1 = Transmitter idle

RDRF — Receive Data Register Full Flag

This flag is set if a received character is ready to be read from SCDR. Clear the RDRF flag by reading SCSR and then reading SCDR.

- 0 = SCDR empty
- 1 = SCDR full

IDLE — Idle Line Detected Flag

This flag is set if the RxD line is idle. Once cleared, IDLE is not set again until the RxD line has been active and becomes idle again. The IDLE flag is inhibited when RWU = 1. Clear IDLE by reading SCSR and then reading SCDR.

- 0 = RxD line active
- 1 = RxD line idle

Serial Communications Interface (SCI)

OR — Overrun Error Flag

OR is set if a new character is received before a previously received character is read from SCDR. Clear the OR flag by reading SCSR and then reading SCDR.

- 0 = No overrun
- 1 = Overrun detected

NF — Noise Error Flag

NF is set if majority sample logic detects anything other than a unanimous decision. Clear NF by reading SCSR and then reading SCDR.

- 0 = Unanimous decision
- 1 = Noise detected

FE — Framing Error Flag

FE is set when a 0 is detected where a stop bit was expected. Clear the FE flag by reading SCSR and then reading SCDR.

- 0 = Stop bit detected
- 1 = Zero detected

Bit 0 — Unimplemented

Always reads 0

7.3.5 Serial Communications Data Register

Address: \$002F

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|---------------------|-------|-------|-------|-------|-------|-------|-------|
| Read: | R7/T7 | R6/T6 | R5/T5 | R4/T4 | R3/T3 | R2/T2 | R1/T1 | R0/T0 |
| Write: | | | | | | | | |
| Reset: | Unaffected by reset | | | | | | | |

Figure 7-8. Serial Communications Data Register (SCDR)

NOTE: *SCI receive and transmit data are double buffered. Reads of SCDR access the receive data buffer and writes access the transmit data buffer.*

Section 8. Serial Peripheral Interface (SPI)

8.1 Contents

| | | |
|-------|---|----|
| 8.2 | Introduction | 81 |
| 8.3 | SPI Registers | 83 |
| 8.3.1 | Serial Peripheral Control Register | 83 |
| 8.3.2 | Serial Peripheral Status Register | 85 |
| 8.3.3 | Serial Peripheral Data I/O Register | 86 |

8.2 Introduction

The serial peripheral interface (SPI) allows the microcontroller unit (MCU) to communicate synchronously with peripheral devices and other microprocessors. When configured as a master, data transfer rates can be as high as one-half the E clock rate (1.5 Mbits per second for a 3-MHz bus frequency). When configured as a slave, data transfers can be as fast as the E-clock rate (3 Mbits per second for a 3-MHz bus frequency).

When the SPI is enabled, all pins that are defined by the configuration as inputs are inputs regardless of the state of the DDR bits of those pins. All pins that are defined as outputs will be outputs only if the DDR bits for those pins are set to 1. Any SPI output whose corresponding DDR bit is cleared to 0 can be used as a general-purpose input. If the SPI system is in master mode and DDRD bit 5 is set to 1, the port D bit 5 pin becomes a general-purpose output instead of the \overline{SS} input to the SPI system. The MODF mode error flag function for which \overline{SS} was used becomes disabled to avoid interference between the general-purpose output function and the SPI system.

Refer to [Figure 8-1](#), which shows the SPI block diagram.

Serial Peripheral Interface (SPI)

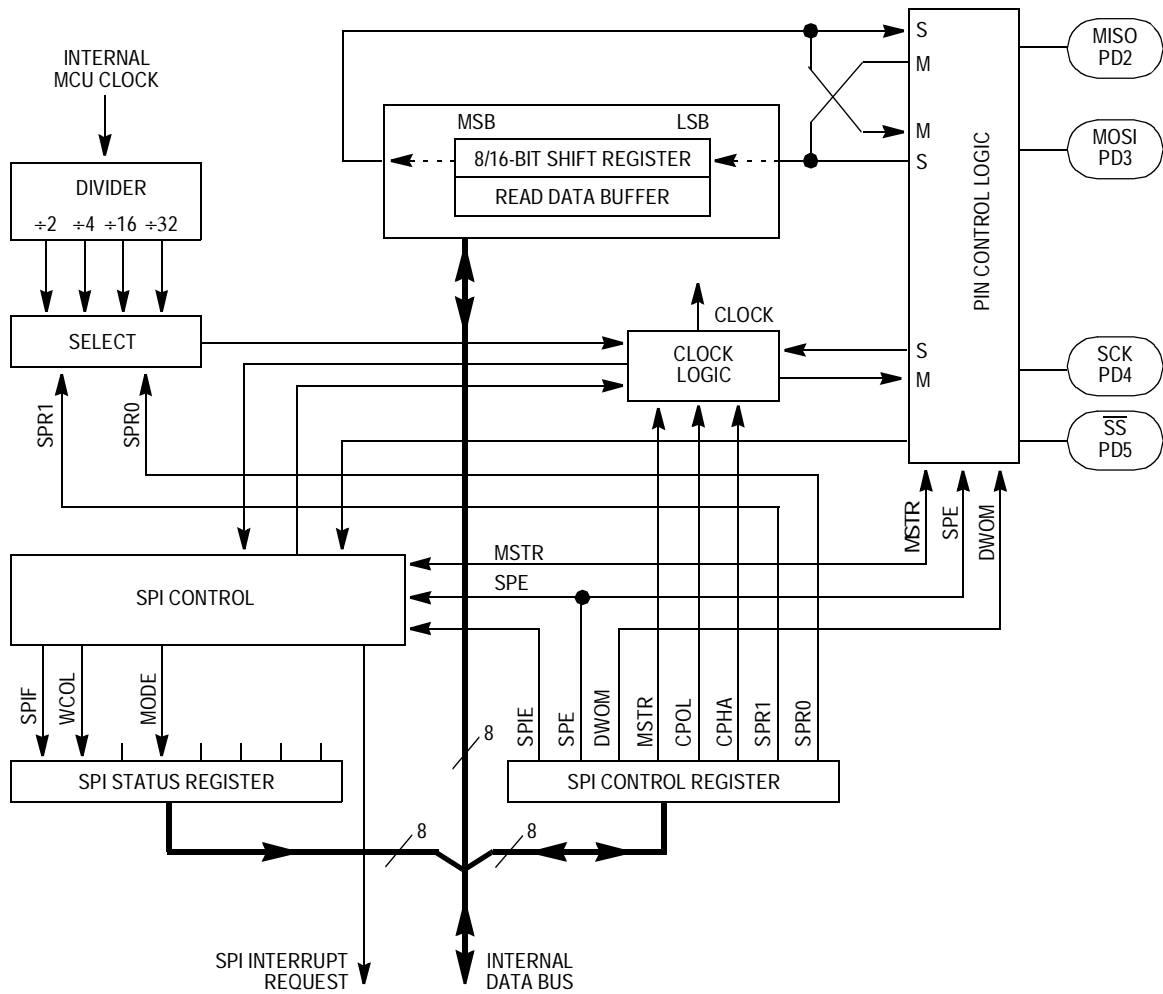


Figure 8-1. SPI Block Diagram

8.3 SPI Registers

This subsection describes the SPI registers.

8.3.1 Serial Peripheral Control Register

Address: \$0028

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-----|------|------|------|------|------|-------|
| Read: | SPIE | SPE | DWOM | MSTR | CPOL | CPHA | SPR1 | SPR0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 1 | U | U |

U = Unaffected

Figure 8-2. Serial Peripheral Control Register (SPCR)

SPIE — Serial Peripheral Interrupt Enable Bit

0 = SPI interrupts disabled

1 = SPI interrupts enabled

SPE — Serial Peripheral System Enable Bit

0 = SPI off

1 = SPI on

DWOM — Port D Wired-OR Mode Option Bit for Port D Pins PD[5:2]

0 = Normal CMOS outputs

1 = Open-drain outputs

MSTR — Master Mode Select Bit

0 = Slave mode

1 = Master mode

CPOL and CPHA — Clock Polarity Bit and Clock Phase Bit

Refer to [Figure 8-3](#).

Serial Peripheral Interface (SPI)

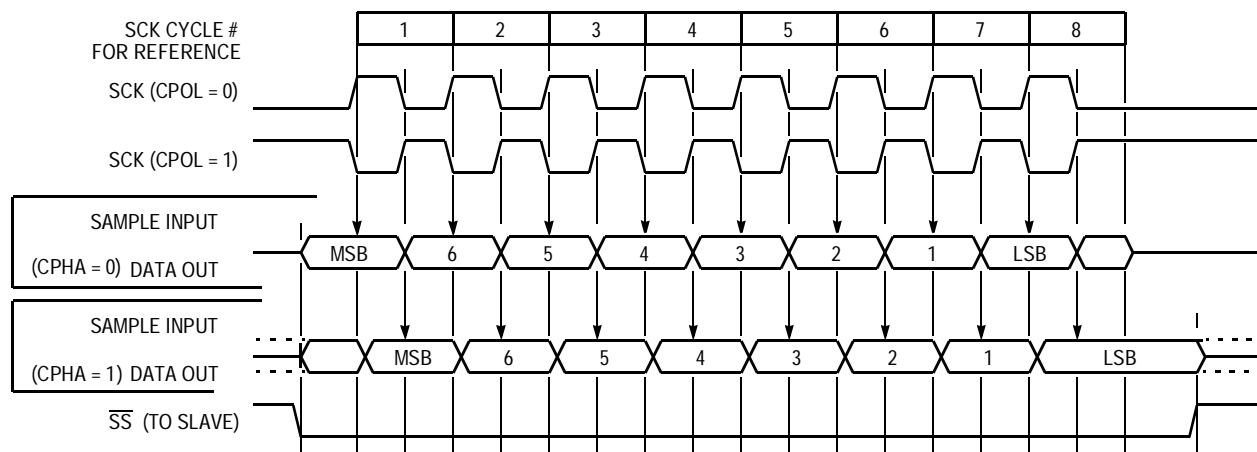


Figure 8-3. SPI Transfer Format

SPR[1:0] — SPI Clock Rate Select Bits

Refer to [Table 8-1](#).

Table 8-1. SPI Clock Rate Selects

| SPR[1:0] | Divide E Clock By | Frequency at E = 1 MHz | Frequency at E = 2 MHz | Frequency at E = 3 MHz |
|----------|-------------------|------------------------|------------------------|------------------------|
| 0 0 | 2 | 500 kHz | 1.0 MHz | 1.5 MHz |
| 0 1 | 4 | 250 kHz | 500 kHz | 750 kHz |
| 1 0 | 16 | 125 kHz | 125 kHz | 375 kHz |
| 1 1 | 32 | 62.5 kHz | 62.5 kHz | 187.5 kHz |

8.3.2 Serial Peripheral Status Register

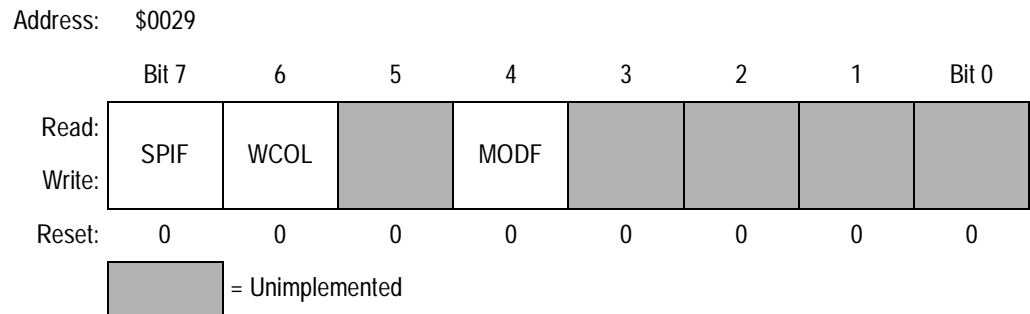


Figure 8-4. Serial Peripheral Status Register (SPSR)

SPIF — SPI Transfer Complete Flag

This flag is set when an SPI transfer is complete (after eight SCK cycles in a data transfer). Clear this flag by reading SPSR, then access SPDR.

- 0 = No SPI transfer complete or SPI transfer still in progress
- 1 = SPI transfer complete

WCOL — Write Collision Error Flag

This flag is set if the MCU tries to write data into SPDR while an SPI data transfer is in progress. Clear this flag by reading SPSR, then access SPDR.

- 0 = No write collision error
- 1 = SPDR written while SPI transfer in progress

Bit 5 — Unimplemented

Always reads 0

MODF — Mode Fault Bit (mode fault terminates SPI operation)

Set when \overline{SS} is pulled low while MSTR = 1. Cleared by SPSR read followed by SPCR write.

- 0 = No mode fault error
- 1 = \overline{SS} pulled low in master mode

Bits [3:0] — Unimplemented

Always read 0

Serial Peripheral Interface (SPI)

8.3.3 Serial Peripheral Data I/O Register

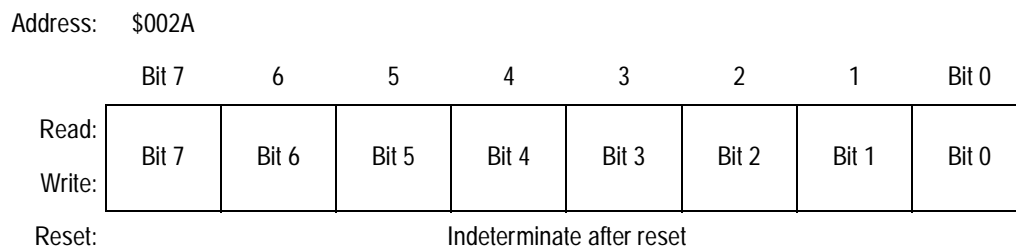


Figure 8-5. Serial Peripheral Data I/O Register (SPDR)

SPI is double buffered in and single buffered out.

Section 9. Timing System

9.1 Contents

| | | |
|--------|---|-----|
| 9.2 | Introduction | 88 |
| 9.3 | Timer Registers | 91 |
| 9.3.1 | Timer Compare Force Register | 91 |
| 9.3.2 | Output Compare 1 Mask Register | 92 |
| 9.3.3 | Output Compare 1 Data Register | 92 |
| 9.3.4 | Timer Count Register | 93 |
| 9.3.5 | Timer Input Capture Registers | 94 |
| 9.3.6 | Timer Output Compare Registers | 95 |
| 9.3.7 | Timer Input Capture 4/Output Compare 5 Register | 96 |
| 9.3.8 | Timer Control Register 1 | 97 |
| 9.3.9 | Timer Control Register 2 | 98 |
| 9.3.10 | Timer Interrupt Mask 1 Register | 99 |
| 9.3.11 | Timer Interrupt Flag 1 Register | 100 |
| 9.3.12 | Timer Interrupt Mask 2 Register | 100 |
| 9.3.13 | Timer Interrupt Flag Register 2 | 102 |
| 9.4 | Pulse Accumulator | 103 |
| 9.4.1 | Pulse Accumulator Control Register | 104 |
| 9.4.2 | Pulse Accumulator Counter Register | 106 |

9.2 Introduction

The timing system is based on a free-running 16-bit counter with a four-stage programmable prescaler. A timer overflow function allows software to extend the system's timing capability beyond the counter's 16-bit range.

The timer has:

- Three channels for input capture
- Four channels for output compare
- One channel that can be configured as a fourth input capture or a fifth output compare

In addition, the timing system includes pulse accumulator and real-time interrupt (RTI) functions, as well as a clock monitor function, which can be used to detect clock failures that are not detected by the computer operating properly (COP. Refer to [9.4 Pulse Accumulator](#) for further information about these functions.

[Table 9-1](#) provides a summary of the crystal-related frequencies and periods. A block diagram of the timer system is shown in [Figure 9-1](#).

Table 9-1. Timer Summary

| Control Bits | Common System Frequencies | | | Definition |
|-----------------------------------|---|--------------------------|----------------------------|--------------------------------|
| | 4.0 MHz | 8.0 MHz | 12.0 MHz | XTAL |
| | 1.0 MHz | 2.0 MHz | 3.0 MHz | E |
| PR[1:0] | Main Timer Count Rates (Period Length) | | | |
| 0 0 1 count — overflow — | 1000 ns 65.536 ms | 500 ns 32.768 ms | 333 ns 21.845 ms | $1 \div E$ $2^{16} \div E$ |
| 0 1 1 count — overflow — | 4.0 μ s 262.14 ms | 2.0 μ s 131.07 ms | 1.333 μ s 87.381 ms | $4 \div E$ $2^{18} \div E$ |
| 1 0 1 count — overflow — | 8.0 μ s 524.28 ms | 4.0 μ s 262.14 ms | 2.667 μ s 174.76 ms | $8 \div E$ $2^{19} \div E$ |
| 1 1 1 count — overflow — | 16.0 μ s 1.049 s | 8.0 μ s 524.29 ms | 5.333 μ s 349.52 ms | $16 \div E$ $2^{20} \div E$ |
| RTR[1:0] | Periodic (RTI) Interrupt Rates (Period Length) | | | |
| 0 0 | 8.192 ms | 4.096 ms | 2.731 ms | $2^{13} \div E$ |
| 0 1 | 16.384 ms | 8.192 ms | 5.461 ms | $2^{14} \div E$ |
| 1 0 | 32.768 ms | 16.384 ms | 10.923 ms | $2^{15} \div E$ |
| 1 1 | 65.536 ms | 32.768 ms | 21.845 ms | $2^{16} \div E$ |
| CR[1:0] | COP Watchdog Timeout Rates (Period Length) | | | |
| 0 0 | 32.768 ms | 16.384 ms | 10.923 ms | $2^{15} \div E$ |
| 0 1 | 131.072 ms | 65.536 ms | 43.691 ms | $2^{17} \div E$ |
| 1 0 | 524.288 ms | 262.14 ms | 174.76 ms | $2^{19} \div E$ |
| 1 1 | 2.098 s | 1.049 s | 699.05 ms | $2^{21} \div E$ |
| Timeout tolerance (-0 ms/+...) | 32.8 ms | 16.4 ms | 10.9 ms | $2^{15} \div E$ |

Timing System

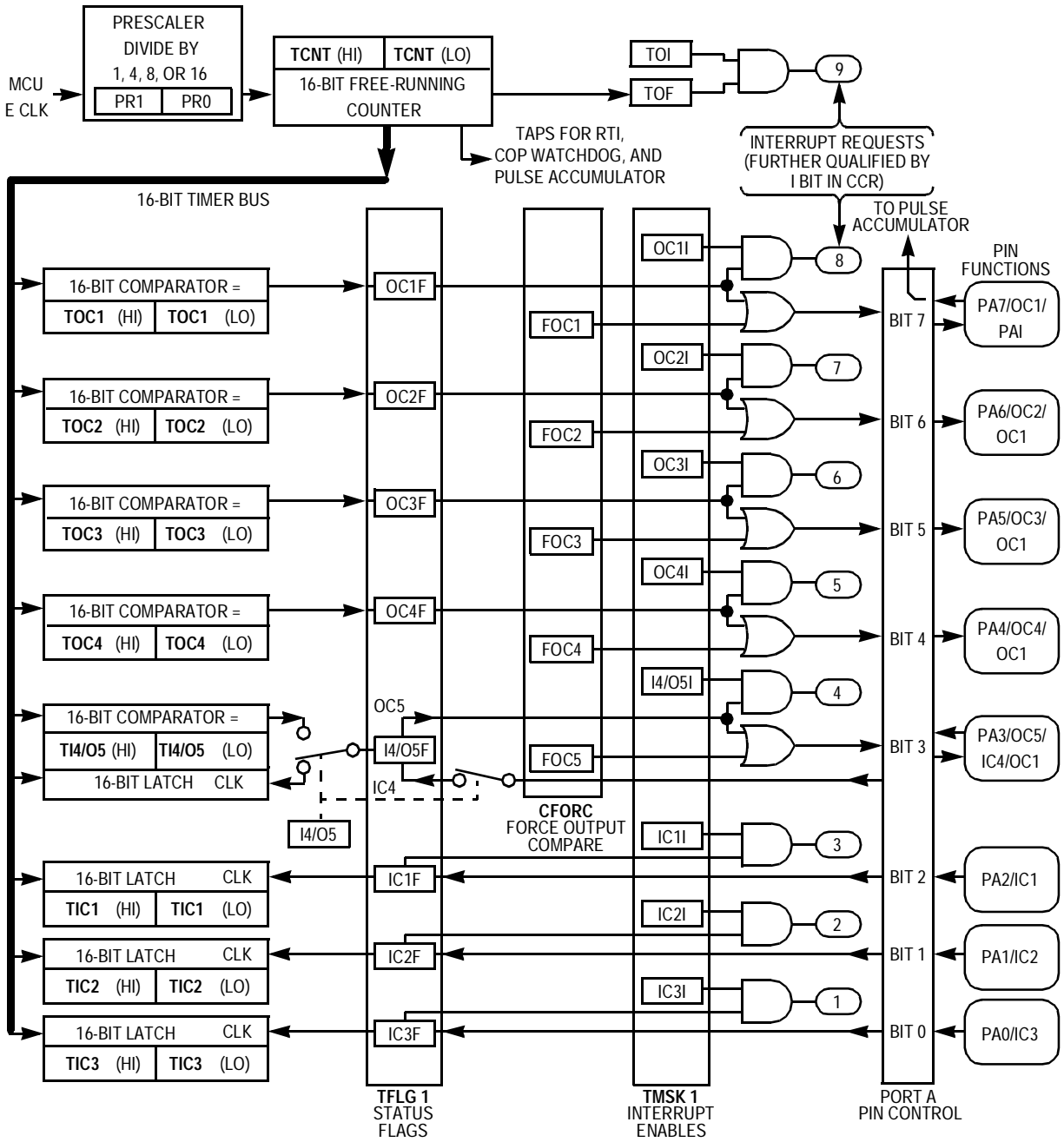


Figure 9-1. Timer Block Diagram

9.3 Timer Registers

This subsection provides a description of the registers associated with the timer system.

9.3.1 Timer Compare Force Register

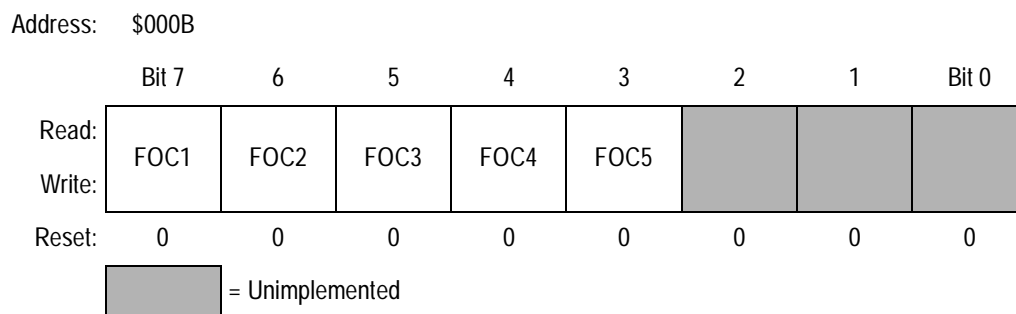


Figure 9-2. Timer Compare Force Register (CFORC)

FOC[5:1] — Force Output Compare Bits

Write 1s to force compare(s)

0 = Not affected

1 = Output x action occurs

Bits [2:0] — Unimplemented

Always read 0

9.3.2 Output Compare 1 Mask Register

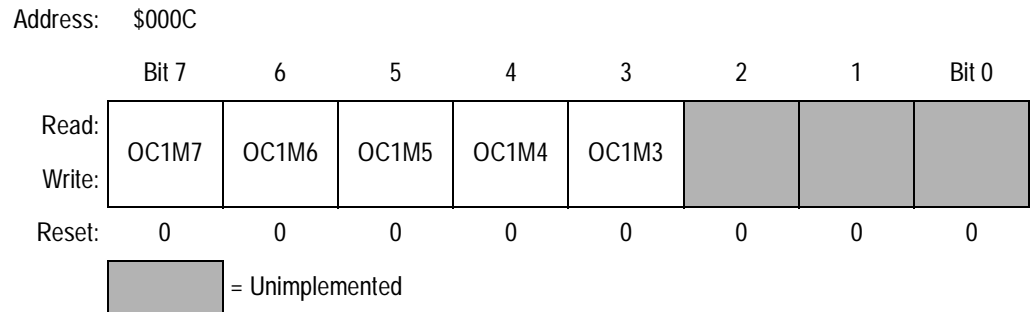


Figure 9-3. Output Compare 1 Mask Register (OC1M)

OC1M[7:3] — Output Compare 1 Mask Bits

Set bit(s) to enable OC1 to control corresponding pin(s) of port A

Bits [2:0] — Unimplemented

Always read as 0

9.3.3 Output Compare 1 Data Register

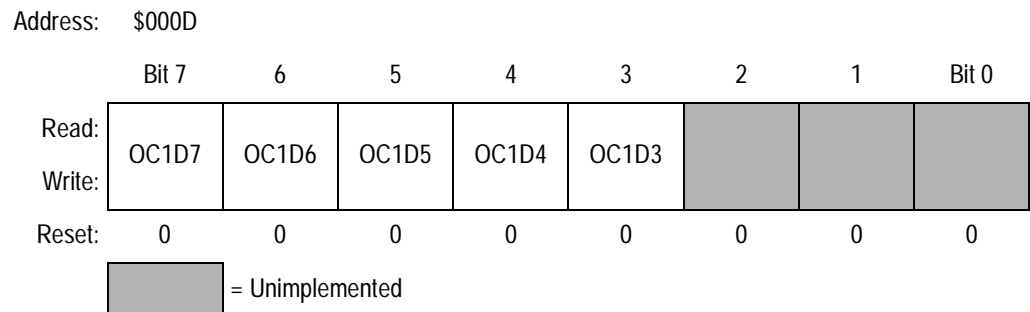


Figure 9-4. Output Compare 1 Data Register (OC1D)

OC1D[7:3] — Output Compare 1 Data Bits

If OC1M_x is set, data in OC1D_x is output to port A bit x on successful OC1 compares.

Bits [2:0] — Unimplemented

Always read 0

9.3.4 Timer Count Register

Address: \$000E — High

| | | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|-------|-------|
| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Read: | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Address: \$000F — Low

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


 = Unimplemented

Figure 9-5. Timer Count Register (TCNT)

The timer count register (TCNT) is read only in normal modes.

9.3.5 Timer Input Capture Registers

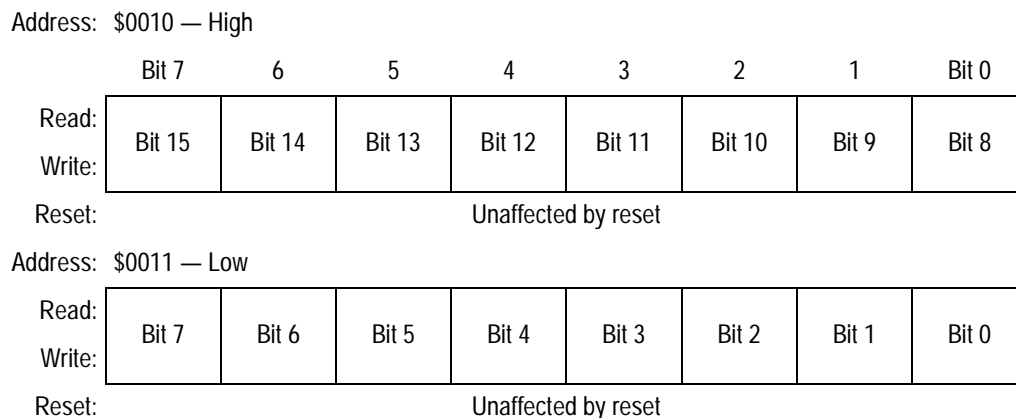


Figure 9-6. Timer Input Capture Register 1 (TIC1)

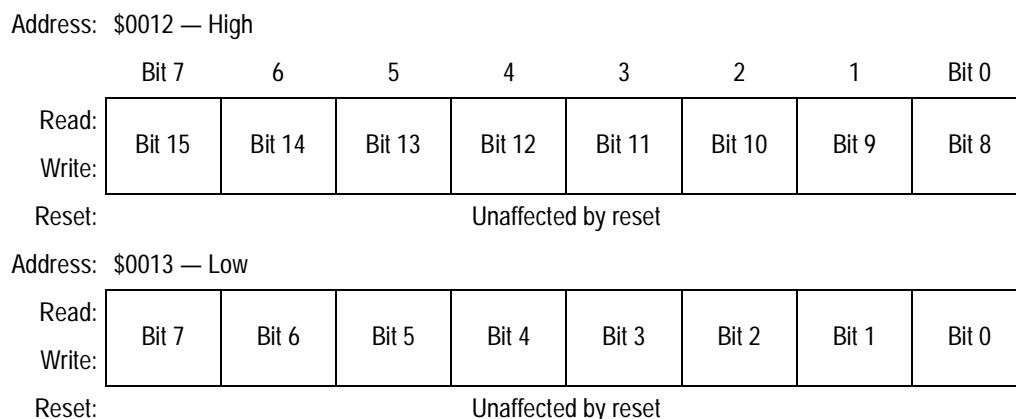


Figure 9-7. Timer Input Capture Register 2 (TIC2)

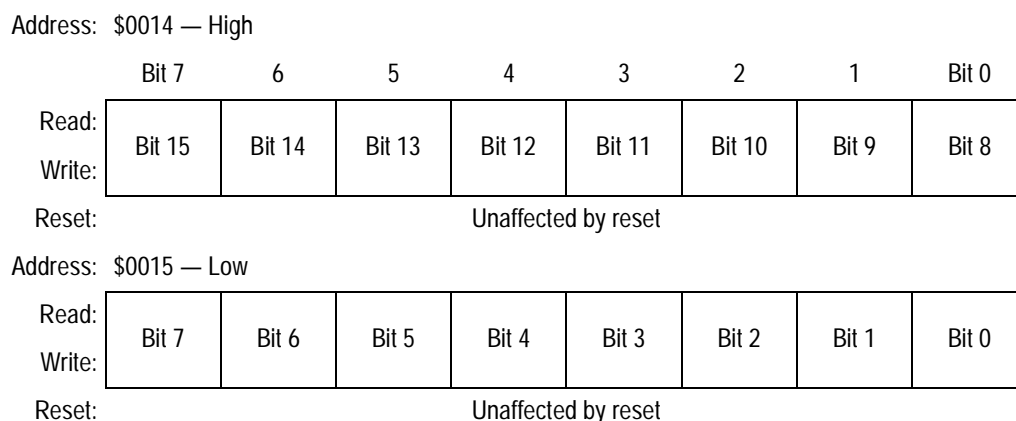


Figure 9-8. Timer Input Capture Register 3 (TIC3)

9.3.6 Timer Output Compare Registers

Address: \$0016 — High

| | | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|-------|-------|
| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Read: | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| Write: | | | | | | | | |
| Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Address: \$0017 — Low

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

Figure 9-9. Timer Output Compare Register 1 (TOC1)

Address: \$0018 — High

| | | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|-------|-------|
| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Read: | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| Write: | | | | | | | | |
| Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Address: \$0019 — Low

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

Figure 9-10. Timer Output Compare Register 2 (TOC2)

Address: \$001A — High

| | | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|-------|-------|
| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Read: | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| Write: | | | | | | | | |
| Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Address: \$001B — Low

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

Figure 9-11. Timer Output Compare Register 3 (TOC3)

Address: \$001C — High

| | | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|-------|-------|
| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Read: | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| Write: | | | | | | | | |
| Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Address: \$001D — Low

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

Figure 9-12. Timer Output Compare Register 4 (TOC4)

9.3.7 Timer Input Capture 4/Output Compare 5 Register

Address: \$001E — High

| | | | | | | | | |
|--------|--------|--------|--------|--------|--------|--------|-------|-------|
| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
| Read: | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| Write: | | | | | | | | |
| Reset: | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Address: \$001F — Low

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

Figure 9-13. Timer Input Capture4/Output Compare 5 Register (TI4/O5)

This is a shared register and is either input capture 4 or output compare 5 depending on the state of bit I4/O5 in PACTL. Writes to TI4/O5 have no effect when this register is configured as input capture 4.

9.3.8 Timer Control Register 1

Address: \$0020

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-----|-----|-----|-----|-----|-----|-------|
| Read: | OM2 | OL2 | OM3 | OL3 | OM4 | OL4 | OM5 | OL5 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 9-14. Timer Control Register 1 (TCTL1)

OM[5:2] — Output Mode

OL[5:2] — Output Level

See [Table 9-2](#).

Table 9-2. Timer Output Compare Actions

| OMx | OLx | Action Taken on Successful Compare |
|-----|-----|--|
| 0 | 0 | Timer disconnected from output pin logic |
| 0 | 1 | Toggle OCx output line |
| 1 | 0 | Clear OCx output line to 0 |
| 1 | 1 | Set OCx output line to 1 |

9.3.9 Timer Control Register 2

Address: \$0021

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|
| Read: | EDG4B | EDG4A | EDG1B | EDG1A | EDG2B | EDG2A | EDG3B | EDG3A |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 9-15. Timer Control Register 2 (TCTL2)

Table 9-3. Timer Control Configuration

| EDGxB | EDGxA | Configuration |
|-------|-------|-------------------------------|
| 0 | 0 | Capture disabled |
| 0 | 1 | Capture on rising edges only |
| 1 | 0 | Capture on falling edges only |
| 1 | 1 | Capture on any edge |

9.3.10 Timer Interrupt Mask 1 Register

Address: \$0022

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|------|------|------|--------|------|------|-------|
| Read: | OC1I | OC2I | OC3I | OC4I | I4/O5I | IC1I | IC2I | IC3I |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 9-16. Timer Interrupt Mask 1 Register (TMSK1)

OC1I–OC4I — Output Compare Interrupt Enable Bits

If the OCxF flag bit is set while the OCxI enable bit is set, a hardware interrupt sequence is requested.

I4/OC5I — Input Capture 4 or Output Compare 5 Interrupt Enable Bit

When I4/O5 in PACTL is 1, I4/O5I is the input capture 4 interrupt bit. When I4/O5 in PACTL is 0, I4/O5I is the output compare 5 interrupt control bit.

IC1I–IC3I — Input Capture Interrupt Enable Bits

If the ICxF flag bit is set while the ICxI enable bit is set, a hardware interrupt sequence is requested.

NOTE: Control bits in TMSK1 correspond bit for bit with flag bits in TFLG1. Ones in TMSK1 enable the corresponding interrupt sources.

9.3.11 Timer Interrupt Flag 1 Register

Address: \$0023

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|------|------|------|--------|------|------|-------|
| Read: | OC1F | OC2F | OC3F | OC4F | I4/O5F | IC1F | IC2F | IC3F |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 9-17. Timer Interrupt Flag 1 Register (TFLG1)

Clear flags by writing a 1 to the corresponding bit position(s).

OC1F–OC4F — Output Compare x Flags

Set each time the counter matches output compare x value

I4/O5F — Input Capture 4/Output Compare 5 Flag

Set by IC4 or OC5, depending on which function was enabled by I4/O5 of PACTL

IC1F–IC3F — Input Capture x Flag

Set each time a selected active edge is detected on the ICx input line

9.3.12 Timer Interrupt Mask 2 Register

Address: \$0024

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|------|-------|------|---|---|-----|-------|
| Read: | TOI | RTII | PAOVI | PAII | | | PR1 | PR0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |


 = Unimplemented

Figure 9-18. Timer Interrupt Mask 2 Register (TMSK2)

TOI — Timer Overflow Interrupt Enable Bit

0 = Timer overflow interrupt disabled

1 = Timer overflow interrupt enabled

RTII — Real-Time Interrupt Enable Bit

0 = RTIF interrupts disabled

1 = Interrupt requested when RTIF is set to 1

PAOVI — Pulse Accumulator Overflow Interrupt Enable Bit

Refer to [9.4 Pulse Accumulator](#).

PAII — Pulse Accumulator Interrupt Enable Bit

Refer to [9.4 Pulse Accumulator](#).

Bits [3:2] — Unimplemented

Always read as 0

PR[1:0] — Timer Prescaler Select Bits

In normal modes, PR1 and PR0 can be written only once, and the writes must occur within 64 cycles after reset. Refer to [Table 9-1](#) and [Table 9-4](#) for specific timing values.

Table 9-4. Timer Prescale

| PR[1:0] | Prescaler |
|---------|-----------|
| 0 0 | ÷ 1 |
| 0 1 | ÷ 4 |
| 1 0 | ÷ 8 |
| 1 1 | ÷ 16 |

NOTE: Control bits [7:4] in TMSK2 correspond bit for bit with flag bits [7:4] in TFLG2. Logic 1s in TMSK2 enable the corresponding interrupt sources.

9.3.13 Timer Interrupt Flag Register 2

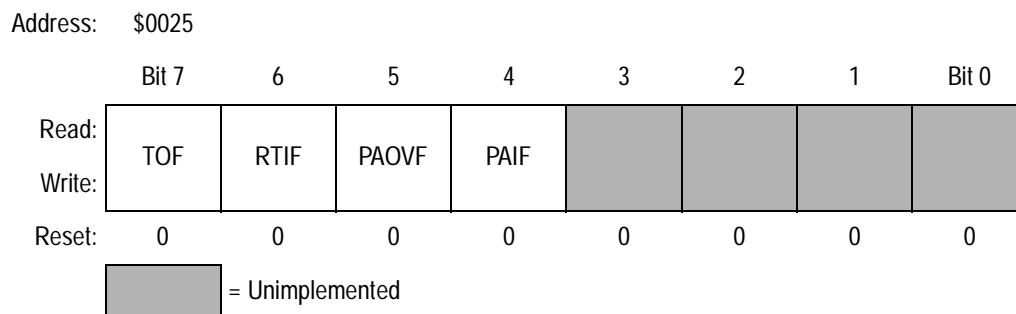


Figure 9-19. Timer Interrupt Flag 2 Register (TFLG2)

Clear flags by writing a 1 to the corresponding bit position(s).

TOF — Timer Overflow Interrupt Flag

Set when TCNT changes from \$FFFF to \$0000

RTIF — Real-Time (Periodic) Interrupt Flag

Set periodically. Refer to the description of bits RTR[1:0] in [Figure 9-21](#).

PAOVF — Pulse Accumulator Overflow Flag

Refer to [9.4 Pulse Accumulator](#).

PAIF — Pulse Accumulator Input Edge Flag

Refer to [9.4 Pulse Accumulator](#).

Bits [3:0] — Unimplemented

Always read 0

Table 9-5. Pulse Accumulator Timing

| | | Common XTAL Frequencies | | |
|--------------------------------|------------------|-------------------------|--------------|----------------|
| | | 4.0 MHz | 8.0 MHz | 12.0 MHz |
| CPU Clock | (E) | 1.0 MHz | 2.0 MHz | 3.0 MHz |
| Cycle Time | $(1 \div E)$ | 1000 ns | 500 ns | 333 ns |
| Pulse Accumulator (Gated Mode) | | | | |
| 1 Count | $(2^6 \div E)$ | 64.0 μ s | 32.0 μ s | 21.330 μ s |
| Overflow | $2^{14} \div E)$ | 16.384 ms | 8.192 ms | 5.491 ms |

Pulse accumulator control bits are also located within two timer registers, TMSK2 and TFLG2, as described in [9.4.1 Pulse Accumulator Control Register](#).

9.4.1 Pulse Accumulator Control Register

Address: \$0026

| | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|-------|------|-------|-------|-------|-------|------|-------|
| Read: | DDRA7 | PAEN | PAMOD | PEDGE | DDRA3 | I4/O5 | RTR1 | RTR0 |
| Write: | | | | | | | | |
| Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 9-21. Pulse Accumulator Control Register (PACTL)

DDRA7 — Data Direction for Port A Bit 7

Refer to [Section 6. Parallel Input/Output \(I/O\) Ports](#).

PAEN — Pulse Accumulator System Enable Bit

0 = Pulse accumulator disabled

1 = Pulse accumulator enabled

PAMOD — Pulse Accumulator Mode Bit

0 = Event counter

1 = Gated time accumulation

PEDGE — Pulse Accumulator Edge Control Bit

0 = In event mode, falling edges increment counter. In gated accumulation mode, high level enables accumulator and falling edge sets PAIF.

1 = In event mode, rising edges increment counter. In gated accumulation mode, low level enables accumulator and rising edge sets PAIF.

DDRA3 — Data Direction for Port A Bit 3

Refer to [Section 6. Parallel Input/Output \(I/O\) Ports](#).

I4/O5 — Input Capture 4/Output Compare Bit

Configure TI4/O5 for input capture or output compare

0 = OC5 enabled

1 = IC4 enabled

RTR[1:0] — RTI Interrupt Rate Select Bits

These two bits select the rate for periodic interrupts. Refer to [Table 9-6](#) and [Table 9-7](#).

Table 9-6. RTI Rates (Period Length)

| RTR[1:0] | Period Length Selected | Period Length | | |
|----------|------------------------|---------------|-------------|-------------|
| | | E = 1.0 MHz | E = 2.0 MHz | E = 3.0 MHz |
| 0 0 | $2^{13} \div E$ | 8.19 ms | 4.096 ms | 2.731 ms |
| 0 1 | $2^{14} \div E$ | 16.38 ms | 8.192 ms | 5.461 ms |
| 1 0 | $2^{15} \div E$ | 32.77 ms | 16.384 ms | 10.923 ms |
| 1 1 | $2^{16} \div E$ | 65.54 ms | 32.768 ms | 21.845 ms |

Table 9-7. RTI Rates (Frequency)

| RTR[1:0] | Rate Selected | Frequency | | |
|----------|-----------------|-------------|-------------|-------------|
| | | E = 1.0 MHz | E = 2.0 MHz | E = 3.0 MHz |
| 0 0 | $E \div 2^{13}$ | 122.070 Hz | 244.141 Hz | 366.211 Hz |
| 0 1 | $E \div 2^{14}$ | 61.035 Hz | 122.070 Hz | 183.105 Hz |
| 1 0 | $E \div 2^{15}$ | 30.518 Hz | 61.035 Hz | 91.553 Hz |
| 1 1 | $E \div 2^{16}$ | 15.259 Hz | 30.518 Hz | 45.776 Hz |

9.4.2 Pulse Accumulator Counter Register

Refer to [Figure 9-22](#) for a description of the pulse accumulator counter register (PACNT).

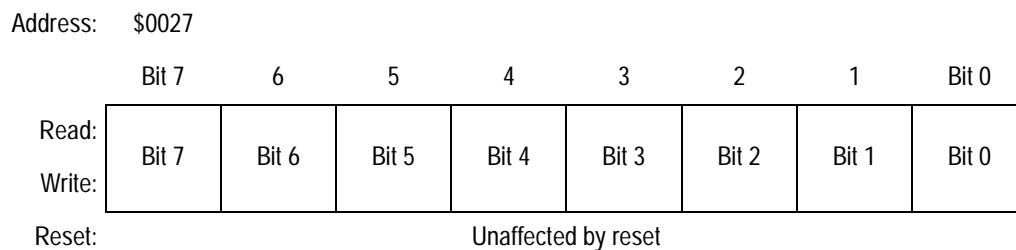


Figure 9-22. Pulse Accumulator Counter Register (PACNT)

How to Reach Us:

USA/EUROPE/LOCATIONS NOT LISTED:

Motorola Literature Distribution
P.O. Box 5405
Denver, Colorado 80217
1-303-675-2140
1-800-441-2447

TECHNICAL INFORMATION CENTER:

1-800-521-6274

JAPAN:

Motorola Japan Ltd.
SPS, Technical Information Center
3-20-1, Minami-Azabu, Minato-ku
Tokyo 106-8573 Japan
81-3-3440-3569

ASIA/PACIFIC:

Motorola Semiconductors H.K. Ltd.
Silicon Harbour Centre
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
852-26668334

HOME PAGE:

<http://www.motorola.com/semiconductors/>



MOTOROLA

**MC68HC11ED0TS/D
REV 1**