HC11

MC68HC11F1

Technical Data

TABLE OF CONTENTS

Paragraph **Title** **Page**

SECTION 1INTRODUCTION

SECTION 2 PIN DESCRIPTIONS

SECTION 3 CENTRAL PROCESSING UNIT

TABLE OF CONTENTS
(Continued)

Paragraph **Title** **Page**

TABLE OF CONTENTS
(Continued)

Paragraph **Title** **Page**

SECTION 5 RESETS AND INTERRUPTS

SECTION 6 PARALLEL INPUT/OUTPUT

TABLE OF CONTENTS
(Continued)

Paragraph **Title** **Page**

TABLE OF CONTENTS
(Continued)

Paragraph	Title	Page

SECTION 9 TIMING SYSTEM

SECTION 10 ANALOG-TO-DIGITAL CONVERTER

TABLE OF CONTENTS
(Continued)

Paragraph **Title** **Page**

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS
(Continued)

Figure	Title	Page

LIST OF TABLES

LIST OF TABLES
(Continued)

Table	Title	Page

SECTION 1INTRODUCTION

The MC68HC11F1 high-performance microcontroller unit (MCU) is an enhanced derivative of the M68HC11 family of microcontrollers and includes many advanced features. This MCU, with a nonmultiplexed expanded bus, is characterized by high speed and low power consumption. The fully static design allows operation at frequencies from 4 MHz to dc.

1.1 Features

- M68HC11 Central Processing Unit (CPU)
- Power Saving STOP and WAIT Modes
- 512 Bytes Electrically Erasable Programmable Read-Only Memory (EEPROM)
- 1024 Bytes RAM, Data Retained During Standby
- Nonmultiplexed Address and Data Buses
- Enhanced 16-Bit Timer
- 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
- Enhanced Asynchronous Nonreturn to Zero (NRZ) Serial Communications Interface (SCI)
- Enhanced Synchronous Serial Peripheral Interface (SPI)
- Eight-Channel 8-Bit Analog-to-Digital (A/D) Converter
- Four Chip-Select Signal Outputs with Programmable Clock Stretching
  — Two I/O Chip Selects
  — One Program Chip Select
  — One General-Purpose Chip Select
- Available in 68-Pin Plastic Leaded Chip Carrier (PLCC) and 80-Pin Plastic Quad Flat Pack (QFP)

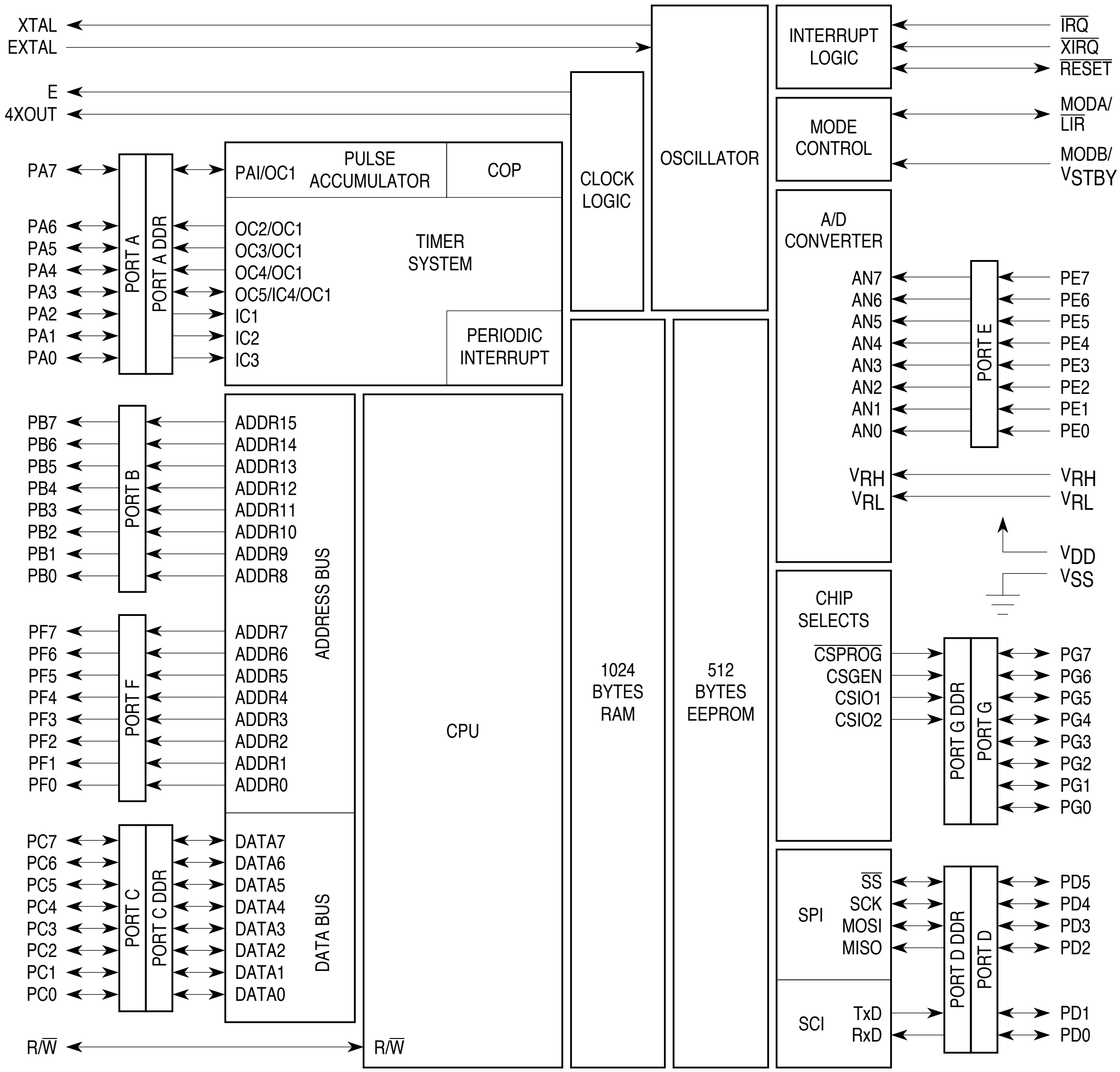


Figure 1-1 MC68HC11F1 Block Diagram

SECTION 2 PIN DESCRIPTIONS

The MC68HC11F1 MCU is available in a 68-pin plastic leaded chip carrier (PLCC) and an 80-pin plastic quad flat pack (QFP). Most pins on this MCU serve two or more functions, as described in the following paragraphs. **Figure 2-1** shows the pin assignments for the PLCC. **Figure 2-2** shows the pin assignments for the QFP.

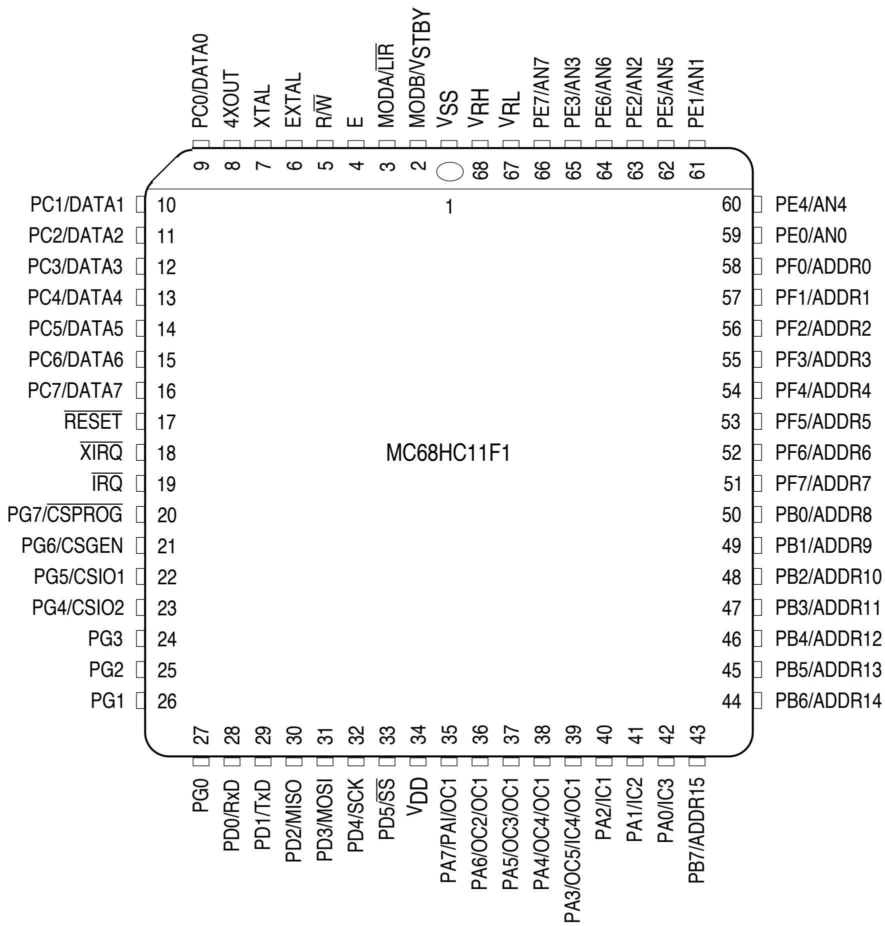


Figure 2-1 Pin Assignments for MC68HC11F1 68-Pin PLCC

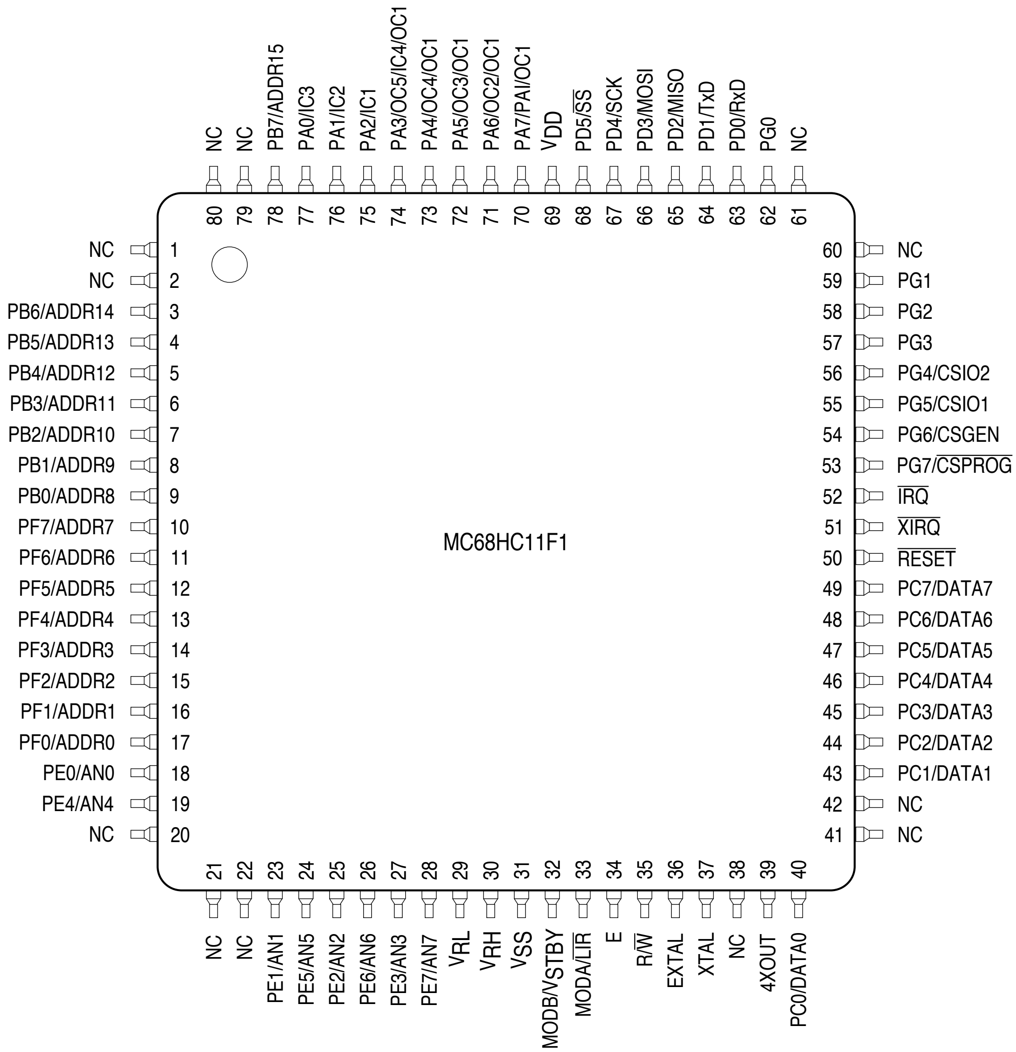


Figure 2-2 Pin Assignments for MC68HC11F1 80-Pin QFP

2.1 V_{DD} and V_{SS}

Power is supplied to the MCU through V_{DD} and V_{SS}. V_{DD} is the power supply, and V_{SS} is ground. The MCU operates from a single 5-volt (nominal) power supply. Very fast signal transitions occur on the MCU pins. The short rise and fall times place high, short duration current demands on the power supply. To prevent noise problems, provide good power-supply bypassing at the MCU. Also, use bypass capacitors that have good high-frequency characteristics and situate them as close to the MCU as possible. Bypass requirements vary, depending on how heavily the MCU pins are loaded.

2.2 Reset ($\overline{\text{RESET}}$)

An active low bidirectional control signal, $\overline{\text{RESET}}$, acts as an input to initialize the MCU to a known start-up state. It also acts as an open-drain output to indicate that an internal failure has been detected in either the clock monitor or COP watchdog circuit. The CPU distinguishes between internal and external reset conditions by sensing whether the reset pin rises to a logic one in less than two E-clock cycles after a reset has occurred. It is not advisable to connect an external resistor-capacitor (RC) power-up delay circuit to the reset pin of M68HC11 devices because the circuit charge time constant can cause the device to misinterpret the type of reset that occurred. Refer to **SECTION 5 RESETS AND INTERRUPTS** for further information.

Figure 2-3 illustrates a reset circuit that uses an external switch. Other circuits can be used, however, it is important to incorporate a low voltage interrupt (LVI) circuit to prevent operation at insufficient voltage levels which could result in erratic behavior or corruption of RAM.

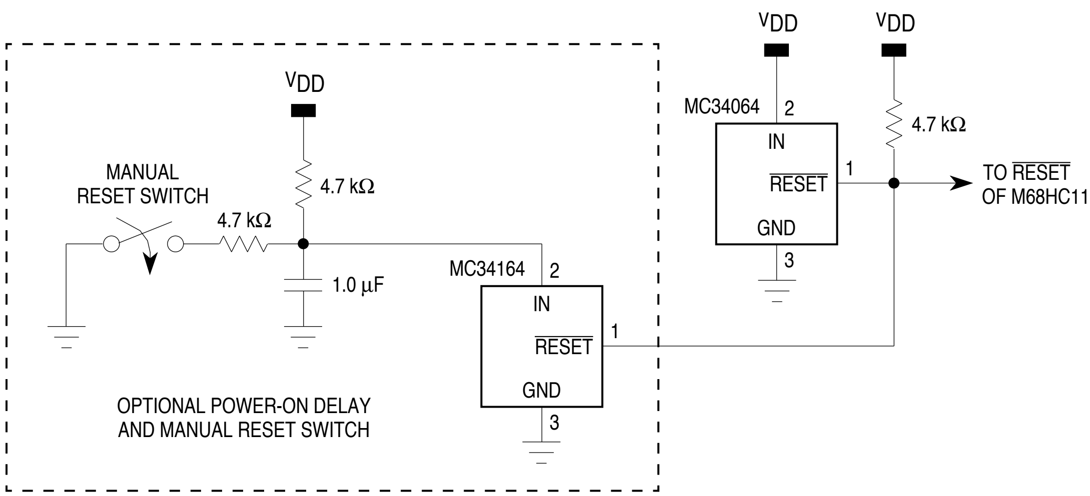


Figure 2-3 External Reset Circuit

2.3 E-Clock Output (E)

E is the output connection for the internally generated E clock. The signal from E is used as a timing reference. The frequency of the E-clock output is one fourth that of the input frequency at the EXTAL pin. When E-clock output is low, an internal process is taking place. When it is high, data is being accessed. All clocks, including the E clock, are halted when the MCU is in STOP mode. The E clock can be turned off in single-chip modes to reduce the effects of radio frequency interference (RFI). Refer to **SECTION 9 TIMING SYSTEM**.

2.4 Crystal Driver and External Clock Input (XTAL, EXTAL)

These two pins provide the interface for either a crystal or a CMOS-compatible clock to control the internal clock generator circuitry. Either a crystal oscillator or a CMOS compatible clock can be used. The resulting E-clock rate is the input frequency divided by four.

The XTAL pin is normally left unterminated when an external CMOS compatible clock is connected to the EXTAL pin. However, a 10 kΩ to 100 kΩ load resistor connected from the XTAL output to ground can be used to reduce RFI noise emission.

The XTAL output is normally used to drive a crystal. The XTAL output can be buffered with a high-impedance buffer, or it can be used to drive the EXTAL input of another M68HC11 device. Refer to **Figure 2-6**.

In all cases, use caution when designing circuitry associated with the oscillator pins. Load capacitances shown in the oscillator circuits include all stray layout capacitances. Refer to **Figure 2-4**, **Figure 2-5**, and **Figure 2-6**.

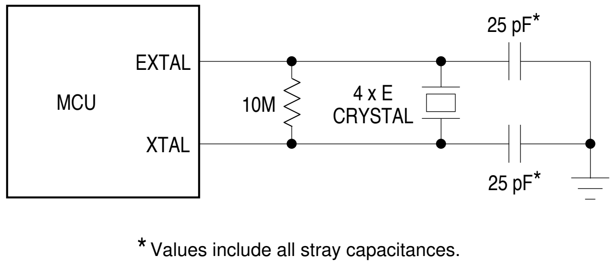


Figure 2-4 Common Crystal Connections

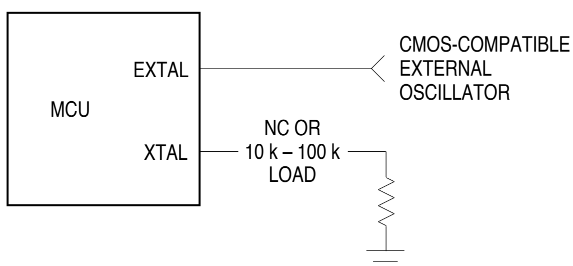


Figure 2-5 External Oscillator Connections

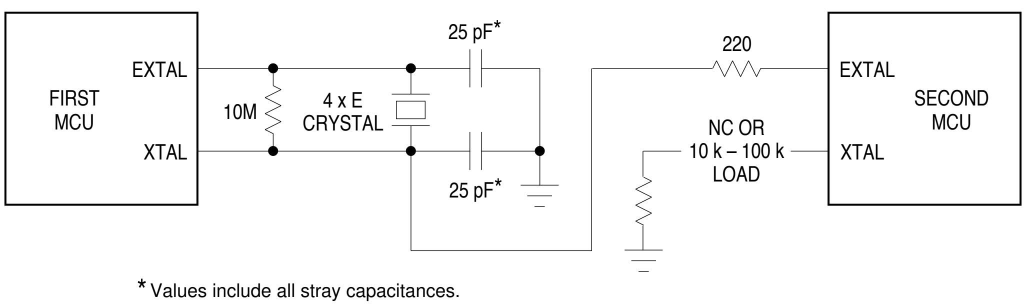


Figure 2-6 One Crystal Driving Two MCUs

2.5 Four Times E-Clock Frequency Output (4XOUT)

Although the circuit shown in **Figure 2-6** will work for any M68HC11 MCU, the MC68HC11F1 has an additional clock output that is four times the E-clock frequency. This output (4XOUT) can be used to directly drive the EXTAL input of another M68HC11 MCU. Refer to **Figure 2-7**. The 4XOUT output is enabled after reset and can be disabled by clearing the CLK4X bit in the OPT2 register.

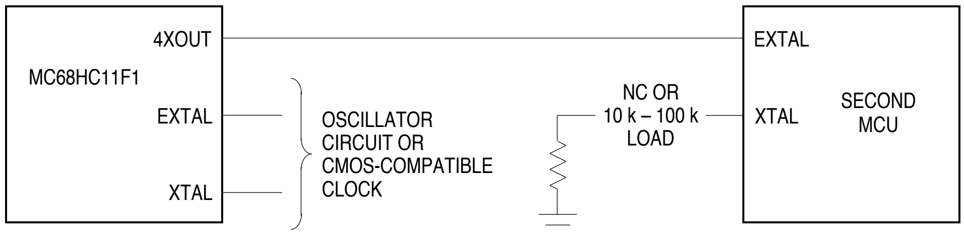


Figure 2-7 4XOUT Signal Driving a Second MCU

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

The $\overline{\text{IRQ}}$ input provides a means of generating asynchronous interrupt requests for the CPU. Either falling-edge triggering or low-level triggering is selected by the IRQE bit in the OPTION register. $\overline{\text{IRQ}}$ is always configured for level-sensitive triggering at reset. Connect an external pull-up resistor, typically 4.7 kΩ, to V_{DD} when $\overline{\text{IRQ}}$ is used in a level-sensitive wired-OR configuration. Refer to **SECTION 5 RESETS AND INTERRUPTS**.

2.7 Non-Maskable Interrupt ($\overline{\text{XIRQ}}$)

The $\overline{\text{XIRQ}}$ input provides a means of requesting a non-maskable interrupt after reset initialization. During reset, the X bit in the condition code register (CCR) is set and any interrupt is masked until MCU software enables it. Because the $\overline{\text{XIRQ}}$ input is level sensitive, it can be connected to a multiple-source wired-OR network with an external pull-up resistor to V_{DD}. $\overline{\text{XIRQ}}$ is often used as a power loss detect interrupt.

Whenever $\overline{\text{XIRQ}}$ or $\overline{\text{IRQ}}$ are used with multiple interrupt sources ($\overline{\text{IRQ}}$ must be configured for level-sensitive operation if there is more than one source of $\overline{\text{IRQ}}$ interrupt), each source must drive the interrupt input with an open-drain type of driver to avoid contention between outputs. There should be a single pull-up resistor near the MCU interrupt input pin (typically 4.7 kΩ). There must also be an interlock mechanism at each interrupt source so that the source holds the interrupt line low until the MCU recognizes and acknowledges the interrupt request. If one or more interrupt sources are still pending after the MCU services a request, the interrupt line will still be held low and the MCU will be interrupted again as soon as the interrupt mask bit in the condition code register (CCR) is cleared (normally upon return from an interrupt). Refer to **SECTION 5 RESETS AND INTERRUPTS**.

2.8 MODA and MODB (MODA/$\overline{\text{LIR}}$ and MODB/V_{STBY})

During reset, MODA and MODB select one of the four operating modes. Refer to **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY**.

After the operating mode has been selected, the $\overline{\text{LIR}}$ pin provides an open-drain output to indicate that execution of an instruction has begun. The LIR pin is configured for wired-OR operation (only pulls low). A series of E-clock cycles occurs during execution of each instruction. The $\overline{\text{LIR}}$ signal is asserted (drives low) during the first E-clock cycle of each instruction (opcode fetch). This output is provided for assistance in program debugging.

The V_{STBY} pin is used to input RAM standby power. The MCU is powered from the V_{DD} signal unless the difference between the level of V_{STBY} and V_{dd} is greater than one MOS threshold (about 0.7 volts). When these voltages differ by more than 0.7 volts, the internal 768-byte RAM and part of the reset logic are powered from V_{STBY} rather than V_{DD}. This allows RAM contents to be retained without V_{DD} power applied to the MCU. Reset must be driven low before V_{DD} is removed and must remain low until V_{DD} has been restored to a valid level.

2.9 V_{RH} and V_{RL}

These pins provide the reference voltage for the analog-to-digital converter. Bypass capacitors should be used to minimize noise on these signals. Any noise on V_{RH} and V_{RL} will directly affect A/D accuracy.

2.10 R/$\overline{\text{W}}$

In expanded and test modes, R/$\overline{\text{W}}$ indicates the direction of transfers on the external data bus. A logic level one on this pin indicates that a read cycle is in progress. A logic zero on this pin indicates that a write cycle is in progress and that no external device should drive the data bus.

The E-clock can be used to enable external devices to drive data onto the data bus during the second half of a read bus cycle (E clock high). R/$\overline{\text{W}}$ can then be used to control the direction of data transfers. R/$\overline{\text{W}}$ drives low when data is being written to the external data bus. R/$\overline{\text{W}}$ will remain low during consecutive data bus write cycles, such as when a double-byte store occurs.

2.11 Port Signals

For the MC68HC11F1, 54 pins are arranged into six 8-bit ports: A, B, C, E, F, and G, and one 6-bit port (D). Each of these seven ports serves a purpose other than I/O, depending on the operating mode or peripheral functions selected. Note that ports B, C, and F are available for I/O functions only in single-chip and bootstrap modes. The pins of ports A, C, D, and G are fully bidirectional. Ports B and F are output-only ports. Port E is an input-only port. Refer to **Table 2-1** for details about the 54 port signals' functions within different operating modes.

Table 2-1 Port Signal Functions

Port/Bit	Single-Chip and Bootstrap Mode	Expanded and Special Test Mode
PA0	PA0/IC3	
PA1	PA1/IC2	
PA2	PA2/IC1	
PA3	PA3/OC5/IC4/OC1	
PA4	PA4/OC4/OC1	
PA5	PA5/OC3/OC1	
PA6	PA6/OC2/OC1	
PA7	PA7/PAI/OC1	
PB[7:0]	PB[7:0]	ADDR[15:8]
PC[7:0]	PC[7:0]	DATA[7:0]
PD0	PD0/RxD	
PD1	PD1/TxD	
PD2	PD2/MISO	
PD3	PD3/MOSI	
PD4	PD4/SCK	
PD5	PD5/$\overline{\text{SS}}$	
PE[7:0]	PE[7:0]/AN[7:0]	
PF[7:0]	PF[7:0]	ADDR[7:0]
PG0	PG0	
PG1	PG1	
PG2	PG2	
PG3	PG3	
PG4	PG4	PG4/CSIO2
PG5	PG5	PG5/CSIO1
PG6	PG6	PG6/CSGEN
PG7	PG7	PG7/$\overline{\text{CSPROG}}$

2.11.1 Port A

Port A is an 8-bit general-purpose I/O port with a data register (PORTA) and a data direction register (DDRA). Port A pins share functions with the 16-bit timer system. PORTA can be read at any time. Inputs return the pin level; outputs return the pin driver input level. If written, PORTA stores the data in internal latches. It drives the pins only if they are configured as outputs. Writes to PORTA do not change the pin state when the pins are configured for timer output compares.

Out of reset, port A pins [7:0] are general-purpose high-impedance inputs. When the timer functions associated with these pins are disabled, the bits in DDRA govern the I/O state of the associated pin. For further information, refer to **SECTION 6 PARALLEL INPUT/OUTPUT**.

NOTE

When using the information about port functions, 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. Port data registers reflect the logic state of the port at reset. The pin function is mode dependent.

2.11.2 Port B

Port B is an 8-bit output-only port. In single-chip modes, port B pins are general-purpose output pins (PB[7:0]). In expanded modes, port B pins act as the high-order address lines (ADDR[15:8]) of the address bus.

PORTB can be read at any time. Reads of PORTB return the pin driver input level. If PORTB is written, the data is stored in internal latches. It drives the pins only in single-chip or bootstrap mode. In expanded operating modes, port B pins are the high-order address outputs (ADDR[15:8]).

Refer to **SECTION 6 PARALLEL INPUT/OUTPUT**.

2.11.3 Port C

Port C is an 8-bit general-purpose I/O port with a data register (PORTC) and a data direction register (DDRC). In single-chip modes, port C pins are general-purpose I/O pins (PC[7:0]). In expanded modes, port C pins are configured as data bus pins (DATA[7:0]).

PORTC can be read at any time. Inputs return the pin level; outputs return the pin driver input level. If PORTC is written, the data is stored in internal latches. It drives the pins only if they are configured as outputs in single-chip or bootstrap mode. Port C pins are general-purpose inputs out of reset in single-chip and bootstrap modes. In expanded and test modes, these pins are data bus lines out of reset.

The CWOM control bit in the OPT2 register disables port C's P-channel output drivers. Because the N-channel driver is not affected by CWOM, setting CWOM causes port C to become an open-drain-type output port suitable for wired-OR operation. In wired-OR mode, (PORTC bits are at logic level zero), pins are actively driven low by the N-channel driver. When a port C bit is at logic level one, the associated pin is in a high-impedance state, as neither the N-channel nor the P-channel devices are active. It is customary to have an external pull-up resistor on lines that are driven by open-drain devices. Port C can only be configured for wired-OR operation when the MCU is in single-chip or bootstrap modes.

Refer to **SECTION 6 PARALLEL INPUT/OUTPUT**.

2.11.4 Port D

Port D, a 6-bit general-purpose I/O port, has a data register (PORTD) and a data direction register (DDRD). The six port D lines (D[5:0]) can be used for general-purpose I/O, for the serial communications interface (SCI) and serial peripheral interface (SPI) subsystems.

PORTD can be read at any time. Inputs return the pin level; outputs return the pin driver input level. If PORTD is written, the data is stored in internal latches and can be driven only if port D is configured for general-purpose output.

The DWOM control bit in the SPCR register disables port D's P-channel output drivers. Because the N-channel driver is not affected by DWOM, setting DWOM causes port D to become an open-drain-type output port suitable for wired-OR operation. In wired-

OR mode, (PORTD bits are at logic level zero), pins are actively driven low by the N-channel driver. When a port D bit is at logic level one, the associated pin is in a high-impedance state, as neither the N-channel nor the P-channel devices are active. It is customary to have an external pull-up resistor on lines that are driven by open-drain devices. Port D can be configured for wired-OR operation in any operating mode.

Refer to **SECTION 6 PARALLEL INPUT/OUTPUT**, **SECTION 7 SERIAL COMMUNICATIONS INTERFACE**, and **SECTION 8 SERIAL PERIPHERAL INTERFACE**.

2.11.5 Port E

Port E is an 8-bit input-only port that is also used as the analog input port for the analog-to-digital converter. Port E pins that are not used for the A/D system can be used as general-purpose inputs. However, PORTE should not be read during the sample portion of an A/D conversion sequence.

Refer to **SECTION 10 ANALOG-TO-DIGITAL CONVERTER**.

2.11.6 Port F

Port F is an 8-bit output-only port. In single-chip mode, port F pins are general-purpose output pins (PF[7:0]). In expanded mode, port F pins act as the low-order address outputs (ADDR[7:0]).

PORTF can be read at any time. Reads of PORTF return the pin driver input level. If PORTF is written, the data is stored in internal latches. It drives the pins only in single-chip or bootstrap mode. In expanded operating modes, port F pins are the low-order address outputs (ADDR[7:0]).

Refer to **SECTION 6 PARALLEL INPUT/OUTPUT**.

2.11.7 Port G

Port G is an 8-bit general-purpose I/O port. When enabled, four chip select signals are alternate functions of port G bits [7:4].

PORTG can be read at any time. Inputs return the pin level; outputs return the pin driver input level. If PORTG is written, the data is stored in internal latches. It drives the pins only if they are configured as outputs.

The GWOM control bit in the OPT2 register disables port G's P-channel output drivers. Because the N-channel driver is not affected by GWOM, setting GWOM causes port G to become an open-drain-type output port suitable for wired-OR operation. In wired-OR mode, (PORTG bits are at logic level zero), pins are actively driven low by the N-channel driver. When a port G bit is at logic level one, the associated pin is in a high-impedance state, as neither the N-channel nor the P-channel devices are active. It is customary to have an external pull-up resistor on lines that are driven by open-drain devices. Port G can be configured for wired-OR operation in any operating mode.

Refer to **SECTION 6 PARALLEL INPUT/OUTPUT** and **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY**.

SECTION 3 CENTRAL PROCESSING UNIT

This section presents information on M68HC11 central processing unit (CPU) architecture. Data types, addressing modes, the instruction set, and the extended addressing range required to support this MCU's memory expansion feature are also included, as are special operations such as subroutine calls and interrupts.

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

3.1 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**.

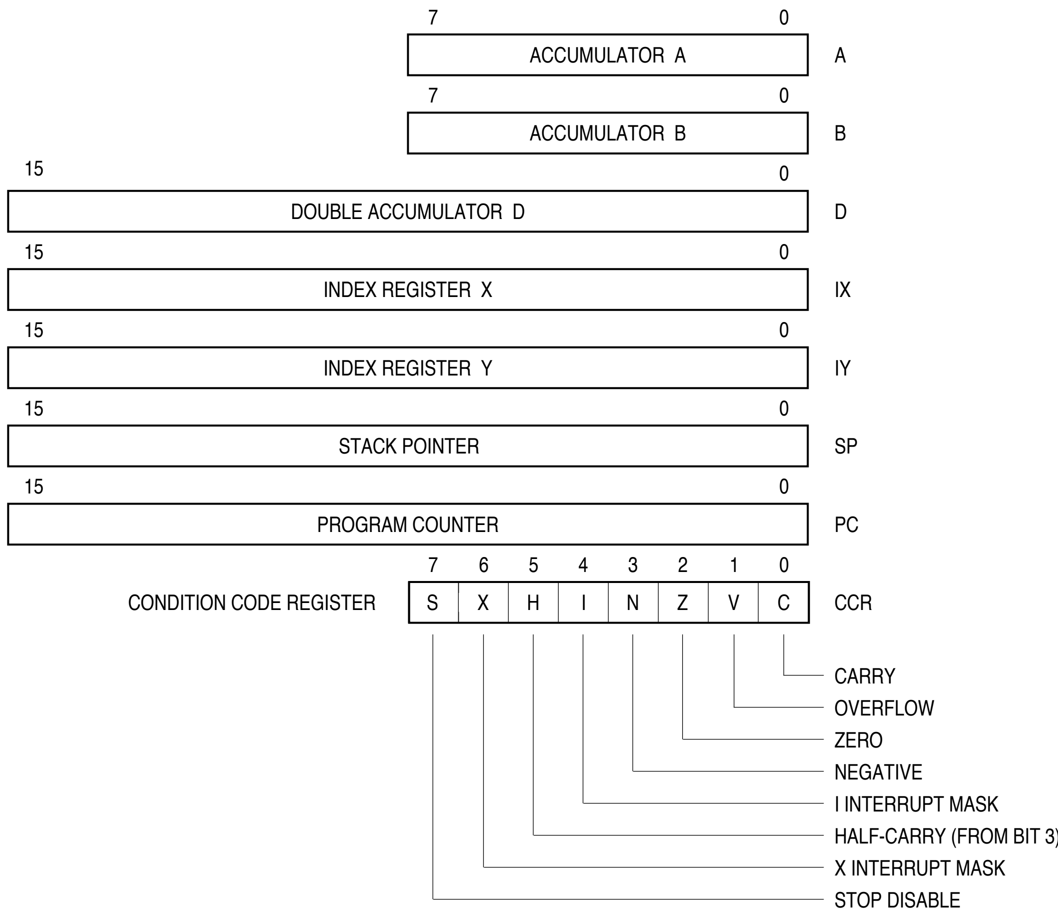


Figure 3-1 Programming Model

3.1.1 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, the following 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.

3.1.2 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.

3.1.3 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 **3.3 Opcodes and Operands** for further information.

3.1.4 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.

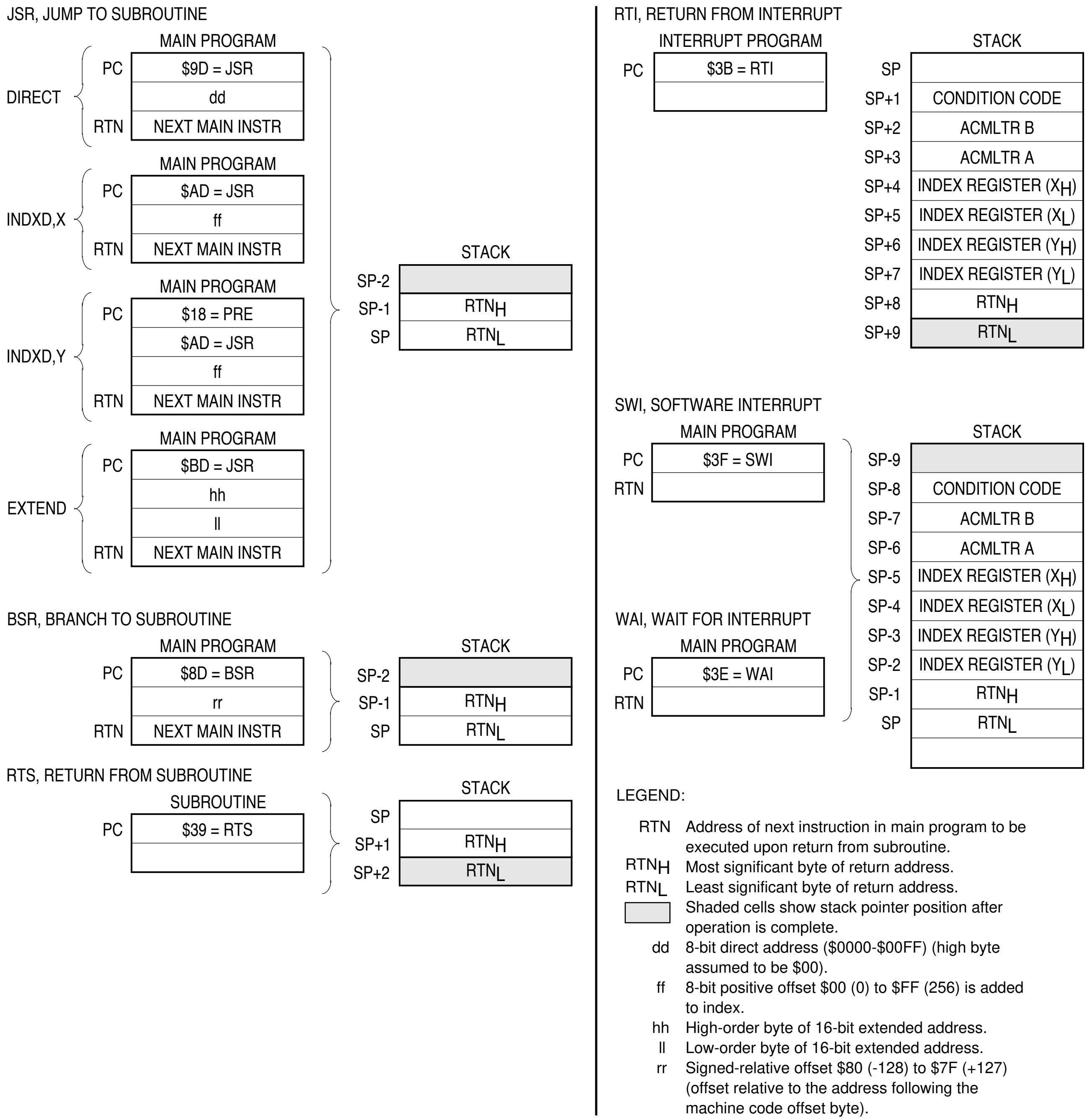


Figure 3-2 Stacking Operations

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, an 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.

There are instructions that push and pull the A and B accumulators and the X and Y index registers. These instructions 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.

3.1.5 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.

Table 3-1 Reset Vector Comparison

	POR or $\overline{\text{RESET}}$ Pin	**Clock Monitor**	**COP Watchdog**
Normal	$FFFE, F	$FFFC, D	$FFFA, B
Test or Boot	$BFFE, F	$BFFC, D	$BFFA, B

3.1.6 Condition Code Register (CCR)

This 8-bit register contains five condition code indicators (C, V, Z, N, and H), two interrupt masking bits, (I and X) and a stop disable bit (S). In the M68HC11 CPU, condition codes are automatically updated 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.

3.1.6.1 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.

3.1.6.2 Overflow (V)

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

3.1.6.3 Zero (Z)

The Z bit is set if the result of an arithmetic, logic, or data manipulation operation is zero. 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 - conditions can be determined.

3.1.6.4 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 one. 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.

3.1.6.5 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 only be cleared 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 zero 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 only be done when there is a clear understanding of latency and of the arbitration mechanism. Refer to **SECTION 5 RESETS AND INTERRUPTS**.

3.1.6.6 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.

3.1.6.7 X Interrupt Mask (X)

The $\overline{\text{XIRQ}}$ 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 zero; 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.

3.1.6.8 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 CPU encounters a STOP instruction 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 disabled by default.

3.2 Data Types

The M68HC11 CPU supports the following data types:

- Bit data
- 8-bit and 16-bit signed and unsigned integers
- 16-bit unsigned fractions
- 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.

3.3 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 four-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.

3.4 Addressing Modes

Six addressing modes can be used to access memory: immediate, direct, extended, indexed, inherent, and 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.

3.4.1 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 two-, three-, and four-

(if prebyte is required) byte immediate instructions. The effective address is the address of the byte following the instruction.

3.4.2 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 two-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.

3.4.3 Extended

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

3.4.4 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 two- to five-byte instructions, depending on whether or not a prebyte is required.

3.4.5 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 one- or two-byte instructions.

3.4.6 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 two-byte instructions.

3.5 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 6)

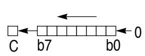
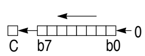
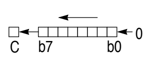
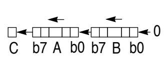
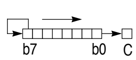
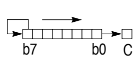
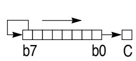
Mnemonic	Operation	Description	Addressing Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	S	X	H	I	N	Z	V	C
ABA	Add Accumulators	A + B ⇒ A	INH	1B	—	2	—	—	Δ	—	Δ	Δ	Δ	Δ
ABX	Add B to X	IX + (00 : B) ⇒ IX	INH	3A	—	3	—	—	—	—	—	—	—	—
ABY	Add B to Y	IY + (00 : B) ⇒ IY	INH	18 3A	—	4	—	—	—	—	—	—	—	—
ADCA (opr)	Add with Carry to A	A + M + C ⇒ A	A IMM A DIR A EXT A IND,X A IND,Y	89 99 B9 A9 18 A9	ii dd hh ll ff ff	2 3 4 4 5	—	—	Δ	—	Δ	Δ	Δ	Δ
ADCB (opr)	Add with Carry to B	B + M + C ⇒ B	B IMM B DIR B EXT B IND,X B IND,Y	C9 D9 F9 E9 18 E9	ii dd hh ll ff ff	2 3 4 4 5	—	—	Δ	—	Δ	Δ	Δ	Δ
ADDA (opr)	Add Memory to A	A + M ⇒ A	A IMM A DIR A EXT A IND,X A IND,Y	8B 9B BB AB 18 AB	ii dd hh ll ff ff	2 3 4 4 5	—	—	Δ	—	Δ	Δ	Δ	Δ
ADDB (opr)	Add Memory to B	B + M ⇒ B	B IMM B DIR B EXT B IND,X B IND,Y	CB DB FB EB 18 EB	ii dd hh ll ff ff	2 3 4 4 5	—	—	Δ	—	Δ	Δ	Δ	Δ
ADDD (opr)	Add 16-Bit to D	D + (M : M + 1) ⇒ D	IMM DIR EXT IND,X IND,Y	C3 D3 F3 E3 18 E3	jj kk dd hh ll ff ff	4 5 6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
ANDA (opr)	AND A with Memory	A • M ⇒ A	A IMM A DIR A EXT A IND,X A IND,Y	84 94 B4 A4 18 A4	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
ANDB (opr)	AND B with Memory	B • M ⇒ B	B IMM B DIR B EXT B IND,X B IND,Y	C4 D4 F4 E4 18 E4	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
ASL (opr)	Arithmetic Shift Left		EXT IND,X IND,Y	78 68 18 68	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
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	77 67 18 67	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
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 • (mm) ⇒ M	DIR IND,X IND,Y	15 1D 18 1D	dd mm ff mm ff mm	6 7 8	—	—	—	—	Δ	Δ	0	—
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 ⊕ V = 0	REL	2C	rr	3	—	—	—	—	—	—	—	—

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

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
BGT (rel)	Branch if > Zero	? Z + (N ⊕ 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 A DIR A EXT A IND,X A IND,Y	85 95 B5 A5 18 A5	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
BITB (opr)	Bit(s) Test B with Memory	B • M	B IMM B DIR B EXT B IND,X B IND,Y	C5 D5 F5 E5 18 E5	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
BLE (rel)	Branch if Δ Zero	? Z + (N ⊕ 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 ⊕ 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 IND,X IND,Y	13 1F 18 1F	dd mm rr ff mm rr ff mm rr	6 7 8	—	—	—	—	—	—	—	—
BRN (rel)	Branch Never	? 1 = 0	REL	21	rr	3	—	—	—	—	—	—	—	—
BRSET(opr) (msk) (rel)	Branch if Bit(s) Set	? ($\overline{M}$) • mm = 0	DIR IND,X IND,Y	12 1E 18 1E	dd mm rr ff mm rr ff mm rr	6 7 8	—	—	—	—	—	—	—	—
BSET (opr) (msk)	Set Bit(s)	M + mm ⇒ M	DIR IND,X IND,Y	14 1C 18 1C	dd mm ff mm ff mm	6 7 8	—	—	—	—	Δ	Δ	0	—
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 ⇒ C	INH	0C	—	2	—	—	—	—	—	—	—	0
CLI	Clear Interrupt Mask	0 ⇒ I	INH	0E	—	2	—	—	—	0	—	—	—	—
CLR (opr)	Clear Memory Byte	0 ⇒ M	EXT IND,X IND,Y	7F 6F 18 6F	hh ll ff ff	6 6 7	—	—	—	—	0	1	0	0
CLRA	Clear Accumulator A	0 ⇒ A	A INH	4F	—	2	—	—	—	—	0	1	0	0
CLRB	Clear Accumulator B	0 ⇒ B	B INH	5F	—	2	—	—	—	—	0	1	0	0
CLV	Clear Overflow Flag	0 ⇒ V	INH	0A	—	2	—	—	—	—	—	—	0	—
CMPA (opr)	Compare A to Memory	A – M	A IMM A DIR A EXT A IND,X A IND,Y	81 91 B1 A1 18 A1	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
CMPB (opr)	Compare B to Memory	B – M	B IMM B DIR B EXT B IND,X B IND,Y	C1 D1 F1 E1 18 E1	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ

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

Mnemonic	Operation	Description	Addressing Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	S	X	H	I	N	Z	V	C
COM (opr)	Ones Complement Memory Byte	$FF – M ⇒ M	EXT IND,X IND,Y	73 63 18 63	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	0	1
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 DIR EXT IND,X IND,Y	1A 83 1A 93 1A B3 1A A3 CD A3	jj kk dd hh ll ff ff	5 6 7 7 7	—	—	—	—	Δ	Δ	Δ	Δ
CPX (opr)	Compare X to Memory 16-Bit	IX – M : M + 1	IMM DIR EXT IND,X IND,Y	8C 9C BC AC CD AC	jj kk dd hh ll ff ff	4 5 6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
CPY (opr)	Compare Y to Memory 16-Bit	IY – M : M + 1	IMM DIR EXT IND,X IND,Y	18 8C 18 9C 18 BC 1A AC 18 AC	jj kk dd hh ll ff ff	5 6 7 7 7	—	—	—	—	Δ	Δ	Δ	Δ
DAA	Decimal Adjust A	Adjust Sum to BCD	INH	19	—	2	—	—	—	—	Δ	Δ	Δ	Δ
DEC (opr)	Decrement Memory Byte	M – 1 ⇒ M	EXT IND,X IND,Y	7A 6A 18 6A	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	—
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 A DIR A EXT A IND,X A IND,Y	88 98 B8 A8 18 A8	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
EORB (opr)	Exclusive OR B with Memory	B ⊕ M ⇒ B	B IMM B DIR B EXT B IND,X B IND,Y	C8 D8 F8 E8 18 E8	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
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 IND,X IND,Y	7C 6C 18 6C	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	—
INCA	Increment Accumulator A	A + 1 ⇒ A	A INH	4C	—	2	—	—	—	—	Δ	Δ	Δ	—
INCB	Increment Accumulator B	B + 1 ⇒ B	B INH	5C	—	2	—	—	—	—	Δ	Δ	Δ	—
INS	Increment Stack Pointer	SP + 1 ⇒ SP	INH	31	—	3	—	—	—	—	—	—	—	—
INX	Increment Index Register X	IX + 1 ⇒ IX	INH	08	—	3	—	—	—	—	—	Δ	—	—

Table 3-2 Instruction Set (Sheet 4 of 6)

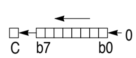
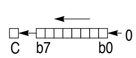
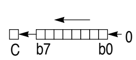
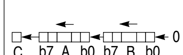
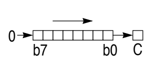
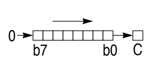
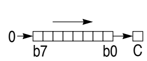
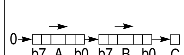
Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
INY	Increment Index Register Y	IY + 1 ⇒ IY	INH	18 08	—	4	—	—	—	—	—	Δ	—	—
JMP (opr)	Jump	See Figure 3–2	EXT IND,X IND,Y	7E 6E 18 6E	hh ll ff ff	3 3 4	—	—	—	—	—	—	—	—
JSR (opr)	Jump to Subroutine	See Figure 3–2	DIR EXT IND,X IND,Y	9D BD AD 18 AD	dd hh ll ff ff	5 6 6 7	—	—	—	—	—	—	—	—
LDAA (opr)	Load Accumulator A	M ⇒ A	A IMM A DIR A EXT A IND,X A IND,Y	86 96 B6 A6 18 A6	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
LDAB (opr)	Load Accumulator B	M ⇒ B	B IMM B DIR B EXT B IND,X B IND,Y	C6 D6 F6 E6 18 E6	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
LDD (opr)	Load Double Accumulator D	M ⇒ A,M + 1 ⇒ B	IMM DIR EXT IND,X IND,Y	CC DC FC EC 18 EC	jj kk dd hh ll ff ff	3 4 5 5 6	—	—	—	—	Δ	Δ	0	—
LDS (opr)	Load Stack Pointer	M : M + 1 ⇒ SP	IMM DIR EXT IND,X IND,Y	8E 9E BE AE 18 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 ⇒ IX	IMM DIR EXT IND,X IND,Y	CE DE FE EE CD 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 ⇒ 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 18 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 18 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
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 ⇒ D	INH	3D	—	10	—	—	—	—	—	—	—	Δ
NEG (opr)	Two's Complement Memory Byte	0 – M ⇒ M	EXT IND,X IND,Y	70 60 18 60	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
NEGA	Two's Complement A	0 – A ⇒ A	A INH	40	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NEGB	Two's Complement B	0 – B ⇒ B	B INH	50	—	2	—	—	—	—	Δ	Δ	Δ	Δ

Table 3-2 Instruction Set (Sheet 5 of 6)

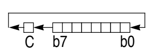
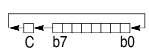
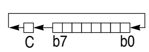
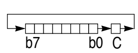
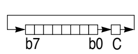
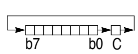
Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
NOP	No operation	No Operation	INH	01	—	2	—	—	—	—	—	—	—	—
ORAA (opr)	OR Accumulator A (Inclusive)	A + M ⇒ A	A IMM A DIR A EXT A IND,X A IND,Y	8A 9A BA AA 18 AA	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
ORAB (opr)	OR Accumulator B (Inclusive)	B + M ⇒ B	B IMM B DIR B EXT B IND,X B IND,Y	CA DA FA EA 18 EA	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
PSHA	Push A onto Stack	A ⇒ Stk,SP = SP – 1	A INH	36	—	3	—	—	—	—	—	—	—	—
PSHB	Push B onto Stack	B ⇒ Stk,SP = SP – 1	B INH	37	—	3	—	—	—	—	—	—	—	—
PSHX	Push X onto Stack (Lo First)	IX ⇒ Stk,SP = SP – 2	INH	3C	—	4	—	—	—	—	—	—	—	—
PSHY	Push Y onto Stack (Lo First)	IY ⇒ Stk,SP = SP – 2	INH	18 3C	—	5	—	—	—	—	—	—	—	—
PULA	Pull A from Stack	SP = SP + 1, A ⇐ Stk	A INH	32	—	4	—	—	—	—	—	—	—	—
PULB	Pull B from Stack	SP = SP + 1, B ⇐ Stk	B INH	33	—	4	—	—	—	—	—	—	—	—
PULX	Pull X From Stack (Hi First)	SP = SP + 2, IX ⇐ Stk	INH	38	—	5	—	—	—	—	—	—	—	—
PULY	Pull Y from Stack (Hi First)	SP = SP + 2, IY ⇐ Stk	INH	18 38	—	6	—	—	—	—	—	—	—	—
ROL (opr)	Rotate Left		EXT IND,X IND,Y	79 69 18 69	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
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 66 18 66	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
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
RTS	Return from Subroutine	See Figure 3–2	INH	39	—	5	—	—	—	—	—	—	—	—
SBA	Subtract B from A	A – B ⇒ A	INH	10	—	2	—	—	—	—	Δ	Δ	Δ	Δ
SBCA (opr)	Subtract with Carry from A	A – M – C ⇒ A	A IMM A DIR A EXT A IND,X A IND,Y	82 92 B2 A2 18 A2	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
SBCB (opr)	Subtract with Carry from B	B – M – C ⇒ B	B IMM B DIR B EXT B IND,X B IND,Y	C2 D2 F2 E2 18 E2	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
SEC	Set Carry	1 ⇒ C	INH	0D	—	2	—	—	—	—	—	—	—	1
SEI	Set Interrupt Mask	1 ⇒ I	INH	0F	—	2	—	—	—	1	—	—	—	—
SEV	Set Overflow Flag	1 ⇒ V	INH	0B	—	2	—	—	—	—	—	—	1	—
STAA (opr)	Store Accumulator A	A ⇒ M	A DIR A EXT A IND,X A IND,Y	97 B7 A7 18 A7	dd hh ll ff ff	3 4 4 5	—	—	—	—	Δ	Δ	0	—

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

Mnemonic	Operation	Description	Addressing Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	S	X	H	I	N	Z	V	C
STAB (opr)	Store Accumulator B	B ⇒ M	B DIR	D7	dd	3	—	—	—	—	Δ	Δ	0	—
			B EXT	F7	hh ll	4								
			B IND,X	E7	ff	4								
			B IND,Y	18 E7	ff	5								
STD (opr)	Store Accumulator D	A ⇒ M, B ⇒ M + 1	DIR	DD	dd	4	—	—	—	—	Δ	Δ	0	—
			EXT	FD	hh ll	5								
			IND,X	ED	ff	5								
			IND,Y	18 ED	ff	6								
STOP	Stop Internal Clocks	—	INH	CF	—	2	—	—	—	—	—	—	—	—
STS (opr)	Store Stack Pointer	SP ⇒ M : M + 1	DIR	9F	dd	4	—	—	—	—	Δ	Δ	0	—
			EXT	BF	hh ll	5								
			IND,X	AF	ff	5								
			IND,Y	18 AF	ff	6								
STX (opr)	Store Index Register X	IX ⇒ M : M + 1	DIR	DF	dd	4	—	—	—	—	Δ	Δ	0	—
			EXT	FF	hh ll	5								
			IND,X	EF	ff	5								
			IND,Y	CD EF	ff	6								
STY (opr)	Store Index Register Y	IY ⇒ M : M + 1	DIR	18 DF	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 ⇒ A	A IMM	80	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
			A DIR	90	dd	3								
			A EXT	B0	hh ll	4								
			A IND,X	A0	ff	4								
			A IND,Y	18 A0	ff	5								
SUBB (opr)	Subtract Memory from B	B – M ⇒ B	A IMM	C0	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
			A DIR	D0	dd	3								
			A EXT	F0	hh ll	4								
			A IND,X	E0	ff	4								
			A IND,Y	18 E0	ff	5								
SUBD (opr)	Subtract Memory from D	D – M : M + 1 ⇒ 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	18 A3	ff	7								
SWI	Software Interrupt	See Figure 3–2	INH	3F	—	14	—	—	—	1	—	—	—	—
TAB	Transfer A to B	A ⇒ B	INH	16	—	2	—	—	—	—	Δ	Δ	0	—
TAP	Transfer A to CC Register	A ⇒ CCR	INH	06	—	2	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ
TBA	Transfer B to A	B ⇒ A	INH	17	—	2	—	—	—	—	Δ	Δ	0	—
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	7D	hh ll	6	—	—	—	—	Δ	Δ	0	0
			IND,X	6D	ff	6								
			IND,Y	18 6D	ff	7								
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	—	**	—	—	—	—	—	—	—	—
XGDX	Exchange D with X	IX ⇒ D, D ⇒ IX	INH	8F	—	3	—	—	—	—	—	—	—	—
XGDY	Exchange D with Y	IY ⇒ D, D ⇒ IY	INH	18 8F	—	4	—	—	—	—	—	—	—	—

SECTION 4OPERATING MODES AND ON-CHIP MEMORY

This section contains information about the modes that define MC68HC11F1 operating conditions, and about the on-chip memory that allows the MCU to be configured for various applications.

4.1 Operating Modes

The values of the mode select inputs MODB and MODA during reset determine the operating mode. Single chip and expanded modes are the normal modes. In single-chip mode only on-board resources are available. Expanded mode, however, allows access to external memory or peripheral devices. Each of these two normal modes is paired with a special mode. Bootstrap mode, a variation of the single-chip mode, executes a bootloader program from an internal bootstrap ROM. Test mode allows privileged access to internal resources.

4.1.1 Single-Chip Operating Mode

In single-chip operating mode, the MC68HC11F1 has no external address or data bus. Ports B, C, and F are available for general-purpose I/O.

4.1.2 Expanded Operating Mode

In expanded operating mode, the MCU can access a 64-Kbyte physical address space. The address space includes the same on-chip memory addresses used for single-chip mode, in addition to external memory and peripheral devices.

The expansion bus is made up of ports B, C, F and the $R/\overline{W}$ signal. In expanded mode, high order address bits are output on the port B pins, low order address bits on the port F pins, and the data bus on port C. The $R/\overline{W}$ pin indicates the direction of data transfer on the port C bus.

4.1.3 Special Test Mode

Special test mode, a variation of the expanded mode, is primarily used during Motorola's internal production testing; however, it is accessible for programming the CONFIG register, programming calibration data into EEPROM, and supporting emulation and debugging during development.

4.1.4 Special Bootstrap Mode

Bootstrap mode is a special variation of the single-chip mode. Bootstrap mode allows special-purpose programs to be entered into internal RAM. When boot mode is selected at reset, a small bootstrap ROM becomes present in the memory map. Reset and interrupt vectors are located in bootstrap ROM at $BFC0–$BFFF. The MCU fetches the reset vector, then executes the bootloader.

The bootstrap ROM contains a small program which initializes the SCI and allows the user to download a program of up to 1024 bytes into on-chip RAM. After a four-character delay, or after receiving the character for address $03FF, control passes to the loaded program at $0000. An external pull-up resistor is required when using the SCI transmitter pin (TxD) because port D pins are configured for wired-OR operation by the bootloader. In bootstrap mode, the interrupt vectors point to RAM. This allows the use of interrupts through a jump table. Refer to Motorola application note AN1060, M68HC11 Bootstrap Mode.

4.2 On-Chip Memory

The MC68HC11F1 contains 1024 bytes of on-chip RAM and 512 bytes of EEPROM. The bootloader ROM occupies 256 bytes. The CONFIG register is implemented as a separate EEPROM byte.

4.2.1 Mapping Allocations

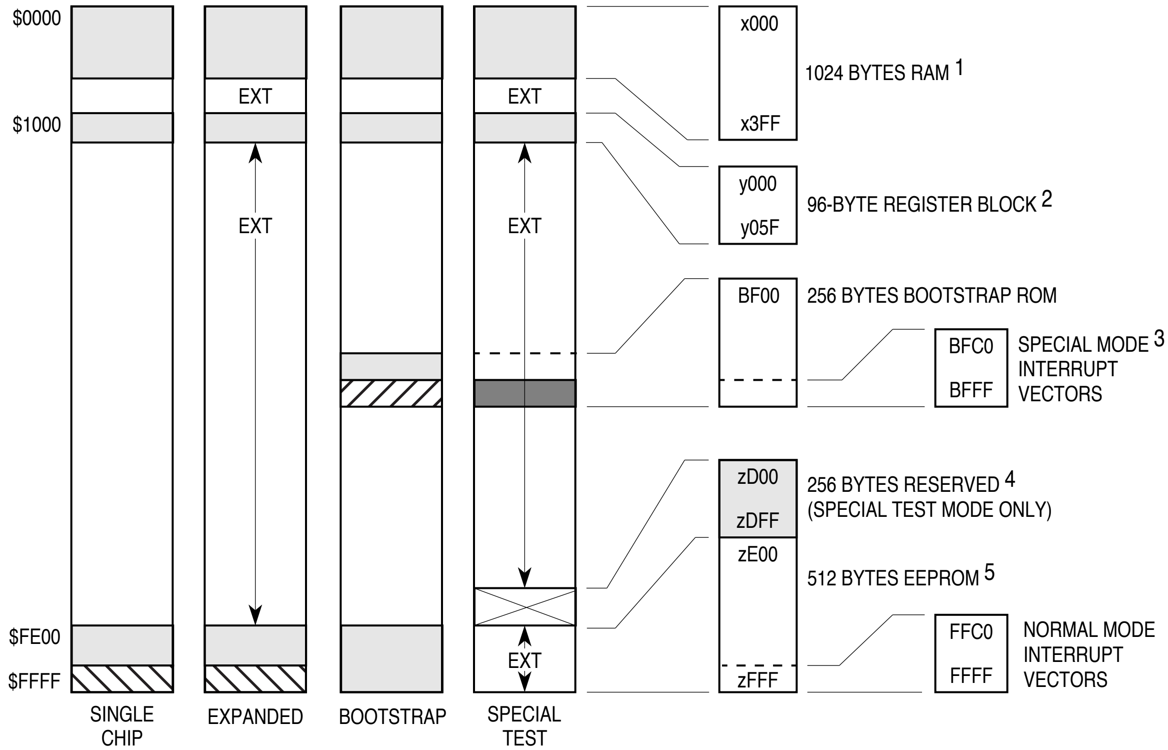
Memory locations for on-chip resources are the same for both expanded and single-chip modes. The 96-byte register block originates at $1000 after reset and can be placed at any other 4-Kbyte boundary ($x000) after reset by writing an appropriate value to the INIT register. Refer to **Figure 4-1**, which illustrates the memory map.

The on-board 1024-byte RAM is initially located at $0000 after reset. If RAM and registers are both mapped to the same 4-Kbyte boundary, the first 96 bytes of RAM are inaccessible (registers have higher priority). Remapping is accomplished by writing appropriate values to the INIT register.

The 512-byte EEPROM array is initially located at $FE00 after reset when EEPROM is enabled in the memory map by the CONFIG register. In expanded and special test modes EEPROM can be placed at any other 4-Kbyte boundary ($xE00) by programming bits EE[3:0] in the CONFIG register to an appropriate value. In single-chip and bootstrap modes the EEPROM is forced on and cannot be remapped.

In special bootstrap mode, a bootloader ROM is enabled at locations $BF00–$BFFF. The vectors for special bootstrap mode are contained in the bootloader program. The boot ROM fills 256 bytes of the memory map even though not all locations are used.

4.2.2 Memory Map



NOTES:
1. RAM can be remapped to any 4-Kbyte boundary ($x000). "x" represents the value contained in RAM[3:0] in the init register.
2. The register block can be remapped to any 4-Kbyte boundary ($y000). "y" represents the value contained in reg[3:0] in the init register.
3. Special test mode vectors are externally addressed.
4. In special test mode the address locations $zD00–$zDFF are not externally addressable. "z" represents the value of bits EE[3:0] in the config register.
5. EEPROM can be remapped to any 4-Kbyte boundary ($z000). "z" represents the value contained in EE[3:0] in the config register.

Figure 4-1 MC68HC11F1 Memory Map

4.2.2.1 RAM

The MC68HC11F1 microcontroller has 1024 bytes of fully static RAM that can be used for storing instructions, variables, and temporary data during program execution. RAM can be placed at any 4-Kbyte boundary in the 64 Kbyte address space by writing an appropriate value to the INIT register.

RAM is initially located at $0000 in the memory map upon reset. Direct addressing mode can access the first 256 locations of RAM using a one-byte address operand. Direct mode accesses save program memory space and execution time.

The on-chip RAM is a fully static memory. RAM contents can be preserved during periods of processor inactivity by either of two methods, both of which reduce power consumption.

During the software-based STOP mode, MCU clocks are stopped, but the MCU continues to draw power from V_{DD}. Power supply current is directly proportional to operating frequency in CMOS integrated circuits and there is very little leakage when the clocks are stopped. These two factors reduce power consumption while the MCU is in STOP mode.

To reduce power consumption to a minimum, V_{DD} can be turned off, and the MODB/V_{STBY} pin can be used to supply RAM power from either a battery back-up or a second power supply. Although this method requires external hardware, it is very effective. Refer to **SECTION 2 PIN DESCRIPTIONS** for information about how to connect the standby RAM power supply. Refer to **SECTION 5 RESETS AND INTERRUPTS** for a description of low power operation.

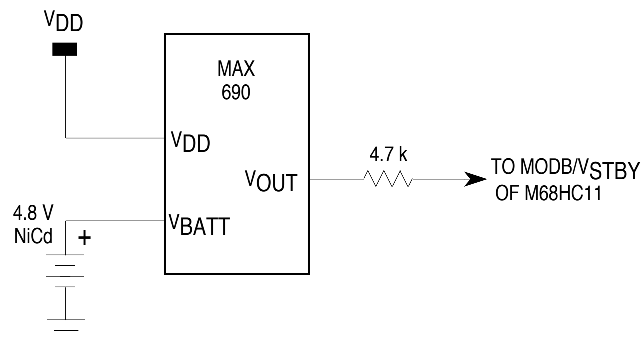


Figure 4-2 RAM Standby MODB/V_{STBY} Connections

4.2.2.2 Bootloader ROM

The bootloader ROM is enabled at address $BF00–$BFFF during special bootstrap mode. The reset vector is fetched from this ROM and the MCU executes the bootloader firmware. In normal modes, the bootloader ROM is disabled.

4.2.2.3 EEPROM

The MC68HC11F1 contains 512 bytes of electrically erasable programmable read-only memory (EEPROM). The default location for EEPROM is $FE00–$FFFF. Other locations can be chosen according to the values written to EE[3:0] in the CONFIG register. In single-chip and bootstrap modes, the EEPROM is forced on and located at the default position. In these modes, the EEPROM cannot be remapped. In special test mode, the EEPROM is disabled initially.

4.2.3 Registers

Table 4-1, a summary of registers and control bits, the registers are shown in ascending order within the 96-byte register block. The addresses shown are for default block mapping ($1000–$105F), however, the register block can be remapped to any 4-Kbyte page ($x000–$x05F) by the INIT register.

Table 4-1 Register and Control Bit Assignments

The register block can be remapped to any 4-Kbyte boundary.

	Bit 7	6	5	4	3	2	1	Bit 0	
$1000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA
$1001	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
$1002	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	PORTG
$1003	DDG7	DDG6	DDG5	DDG4	DDG3	DDG2	DDG1	DDG0	DDRG
$1004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
$1005	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	PORTF
$1006	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC
$1007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
$1008	0	0	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
$1009	0	0	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
$100A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	PORTE
$100B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	CFORC
$100C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	OC1M
$100D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	OC1D
$100E	Bit 15	14	13	12	11	10	9	Bit 8	TCNT (High)
$100F	Bit 7	6	5	4	3	2	1	Bit 0	TCNT (Low)
$1010	Bit 15	14	13	12	11	10	9	Bit 8	TIC1 (High)
$1011	Bit 7	6	5	4	3	2	1	Bit 0	TIC1 (Low)
$1012	Bit 15	14	13	12	11	10	9	Bit 8	TIC2 (High)
$1013	Bit 7	6	5	4	3	2	1	Bit 0	TIC2 (Low)
$1014	Bit 15	14	13	12	11	10	9	Bit 8	TIC3 (High)
$1015	Bit 7	6	5	4	3	2	1	Bit 0	TIC3 (Low)
$1016	Bit 15	14	13	12	11	10	9	Bit 8	TOC1 (High)
$1017	Bit 7	6	5	4	3	2	1	Bit 0	TOC1 (Low)
$1018	Bit 15	14	13	12	11	10	9	Bit 8	TOC2 (High)
$1019	Bit 7	6	5	4	3	2	1	Bit 0	TOC2 (Low)
$101A	Bit 15	14	13	12	11	10	9	Bit 8	TOC3 (High)
$101B	Bit 7	6	5	4	3	2	1	Bit 0	TOC3 (Low)
$101C	Bit 15	14	13	12	11	10	9	Bit 8	TOC4 (High)
$101D	Bit 7	6	5	4	3	2	1	Bit 0	TOC4 (Low)
$101E	Bit 15	14	13	12	11	10	9	Bit 8	TI4/O5 (High)
$101F	Bit 7	6	5	4	3	2	1	Bit 0	TI4/O5 (Low)
$1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1
$1021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2
$1022	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I	TMSK1
$1023	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F	TFLG1
$1024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	TMSK2
$1025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	TFLG2
$1026	0	PAEN	PAMOD	PEDGE	0	I4/O5	RTR1	RTR0	PACTL
$1027	Bit 7	6	5	4	3	2	1	Bit 0	PACNT

Table 4-1 Register and Control Bit Assignments (Continued)

The register block can be remapped to any 4-Kbyte boundary.

	Bit 7	6	5	4	3	2	1	Bit 0	
$1028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
$1029	SPIF	WCOL	0	MODF	0	0	0	Bit 0	SPSR
$102A	Bit 7	6	5	4	3	2	1	Bit 0	SPDR
$102B	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD
$102C	R8	T8	0	M	WAKE	0	0	0	SCCR1
$102D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2
$102E	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	SCSR
$102F	Bit 7	6	5	4	3	2	1	Bit 0	SCDR
$1030	CCF	0	SCAN	MULT	CD	CC	CB	CA	ADCTL
$1031	Bit 7	6	5	4	3	2	1	Bit 0	ADR1
$1032	Bit 7	6	5	4	3	2	1	Bit 0	ADR2
$1033	Bit 7	6	5	4	3	2	1	Bit 0	ADR3
$1034	Bit 7	6	5	4	3	2	1	Bit 0	ADR4
$1035	0	0	0	PTCON	BPRT3	BPRT2	BPRT1	BPRT0	BPROT
$1036									Reserved
$1037									Reserved
$1038	GWOM	CWOM	CLK4X	0	0	0	0	0	OPT2
$1039	ADPU	CSEL	IRQE	DLY	CME	FCME	CR1	CR0	OPTION
$103A	Bit 7	6	5	4	3	2	1	Bit 0	COPRST
$103B	ODD	EVEN	0	BYTE	ROW	ERASE	EELAT	EEPGM	PPROG
$103C	RBOOT	SMOD	MDA	IRV	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO
$103D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT
$103E	TILOP	0	OCCR	CBYP	DISR	FCM	FCOP	0	TEST1
$103F	EE3`	EE2	EE1	EE0	1	NOCOP	1	EEON	CONFIG
$1040									Reserved
to									
$105B									Reserved
$105C	IO1SA	IO1SB	IO2SA	IO2SB	GSTHA	GSTHB	PSTHA	PSTHB	CSSTRH
$105D	IO1EN	IO1PL	IO2EN	IO2PL	GCSPR	PCSEN	PSIZA	PSIZB	CSCTL
$105E	GA15	GA14	GA13	GA12	GA11	GA10	0	0	CSGADR
$105F	IO1AV	IO2AV	0	GNPOL	GAVLD	GSIZA	GSIZB	GSIZC	CSGSIZ

4.3 System Initialization

Registers and bits that control initialization and the basic operation of the MCU are protected against writes except under special circumstances. The following table lists registers that can be written only once after reset or that must be written within the first 64 cycles after reset.

Table 4-2 Write Access Limited Registers

Register Address	Register Name	Must be Written in First 64 Cycles	Write One Time Only
$x024	Timer Interrupt Mask 2 (TMSK2)	Note 1	—
$x035	Block Protect Register (BPROT)	Note 2	—
$x038	System Configuration Options 2 (OPT2)	No	Note 4
$x039	System Configuration Options (OPTION)	Note 3	—
$x03C	Highest Priority I-bit and Miscellaneous (HPRIO)	No	Note 5
$x03D	RAM and I/O Map Register (INIT)	Yes	Note 6

Notes:
1. Bits 1 and 0 can be written once only in first 64 cycles. When SMOD = 1, these bits can be written any time. All other bits can be written at any time.
2. Bits can be written to zero (protection disabled) once only in first 64 cycles or at any time in special modes. Bits can be set to one at any time.
3. Bits 5, 4, 2, 1, and 0 can be written once only in first 64 cycles. When SMOD = 1, bits 5, 4, 2, 1, and 0 can be written at any time. All other bits can be written at any time
4. Bit 5 (CLK4X) can be written only one time.
5. Bit 4 (IRV) can be written only one time.
6. Can be written once in first 64 cycles after reset in normal modes or at any time in special modes.

4.3.1 Mode Selection

The four mode variations are selected by the logic levels present on the MODA and MODB pins at the rising edge of $\overline{\text{RESET}}$. The MODA and MODB logic levels determine the logic state of SMOD and MDA control bits in the HPRIO register.

After reset is released, the mode select pins no longer influence the MCU operating mode. In single-chip operating mode, the MODA pin is connected to a logic level zero. In expanded mode, MODA should be connected to V_{DD} through a pull-up resistor of 4.7 kΩ. The MODA pin also functions as the load instruction register ($\overline{\text{LIR}}$) pin when the MCU is not in reset. The open-drain active low $\overline{\text{LIR}}$ output pin drives low during the first E cycle of each instruction (opcode fetch). The MODB pin also functions as standby power input (V_{STBY}), which allows RAM contents to be maintained in absence of V_{DD}. Refer to **APPENDIX A ELECTRICAL CHARACTERISTICS** for V_{STBY} voltage requirements.

Refer to **Table 4-3**, which is a summary of mode pin operation, the mode control bits, and the four operating modes.

Table 4-3 Hardware Mode Select Summary

Input Levels at Reset		Mode	Control Bits in HPRIO (Latched at Reset)		
MODB	MODA		RBOOT	SMOD	MDA
1	0	Single Chip	0	0	0
1	1	Expanded	0	0	1
0	0	Special Bootstrap	1	1	0
0	1	Special Test	0	1	1

A normal mode is selected when MODB is logic one during reset. One of three reset vectors is fetched from address $FFFA–$FFFF, and program execution begins from the address indicated by this vector. If MODB is logic zero during reset, the special mode reset vector is fetched from addresses $BFFA–$BFFF and software has access to special test features. Refer to **SECTION 5 RESETS AND INTERRUPTS** for information regarding reset vectors.

4.3.1.1 HPRIO Register

Bits in the HPRIO register select the highest priority interrupt level, select whether bootstrap ROM is present, and control visibility of internal reads by the CPU. After reset, MDA and SMOD select the operating mode.

HPRIO — Highest Priority I-Bit Interrupt and Miscellaneous **$103C**

	Bit 7	6	5	4	3	2	1	Bit 0	
	RBOOT*	SMOD*	MDA*	IRV	PSEL3	PSEL2	PSEL1	PSEL0	
RESET:	0	0	0	0	0	1	1	0	Single Chip
	0	0	1	0	0	1	1	0	Expanded
	1	1	0	1	0	1	1	0	Bootstrap
	0	1	1	1	0	1	1	0	Special Test

*Reset states of RBOOT, SMOD, and MDA bits depend on hardware mode selection. Refer to **Table 4-3**.

RBOOT — Read Bootstrap ROM

Set to one out of reset in bootstrap mode. Valid while in special modes only. Can be read anytime. Can only be written in special modes.

0 = Bootloader ROM disabled and not in map
1 = Bootloader ROM enabled and in map at $BF00–$BFFF

SMOD and MDA — Special Mode Select and Mode Select A

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 at any time in special modes. Neither bit can be written is normal modes. SMOD cannot be set once it has been cleared. Refer to **Table 4-3**.

IRV — Internal Read Visibility

IRV can be written at any time in special modes (SMOD = 1). In normal modes (SMOD = 0) IRV can be written only once. In expanded and test modes, IRV determines whether internal read visibility is on or off. In single-chip and bootstrap modes, IRV has no meaning or effect.

0 = No internal read visibility on external bus
1 = Data from internal reads is driven out the external data bus.

PSEL[3:0] — Priority Select Bits [3:0]

Refer to **5.3.1 Highest Priority Interrupt and Miscellaneous Register**.

4.3.2 Initialization

Because bits in the following registers control the basic configuration of the MCU, an accidental change of their values could cause serious system problems. The protection mechanism, overridden in special operating modes, requires a write to the protected bits only within the first 64 bus cycles after any reset, or only once after each reset. **Table 4-2** summarizes the write access limited registers.

4.3.2.1 CONFIG Register

CONFIG controls the presence and position of the EEPROM in the memory map. CONFIG also enables the COP watchdog timer.

CONFIG — System Configuration Register **$103F**

	Bit 7	6	5	4	3	2	1	Bit 0	
	EE3	EE2	EE1	EE0	—	NOCOP	—	EEON	
RESET:	1	1	1	1	1	P	1	1	Single Chip
	1	1	1	1	1	P(L)	1	1	Bootstrap
	P	P	P	P	1	P	1	P	Expanded
	P	P	P	P	1	P(L)	1	0	Special Test

P indicates a previously programmed bit. P(L) indicates that the bit resets to the logic level held in the latch prior to reset, but the function of COP is controlled by DISR bit in TEST1 register.

The CONFIG register consists of an EEPROM byte and static latches that control the start-up configuration of the MCU. The contents of the EEPROM byte are transferred into static working latches during reset sequences. The operation of the MCU is controlled directly by these latches and not by CONFIG itself. In normal modes, changes to CONFIG do not affect operation of the MCU until after the next reset sequence. When programming, the CONFIG register itself is accessed. When the CONFIG register is read, the static latches are accessed.

These bits can be read at any time. The value read is the one latched into the register from the EEPROM cells during the last reset sequence. A new value programmed into this register cannot be read until after a subsequent reset sequence. Unused bits always read as ones.

In special test mode, the static latches can be written directly at any time. In all modes, CONFIG bits can only be programmed using the EEPROM programming sequence, and are neither readable nor active until latched via the next reset. Refer to **4.4.3 CONFIG Register Programming**.

EE[3:0] — EEPROM Mapping Control

EE[3:0] select the upper four bits of the EEPROM base address. In single-chip and bootstrap modes, EEPROM is forced to \$FE00–\$FFFF regardless of the value of EE[3:0].

Table 4-4 EEPROM Mapping

EE[3:0]	EEPROM Position
0 0 0 0	$0E00 – $0FFF
0 0 0 1	$1E00 – $1FFF
0 0 1 0	$2E00 – $2FFF
0 0 1 1	$3E00 – $3FFF
0 1 0 0	$4E00 – $4FFF
0 1 0 1	$5E00 – $5FFF
0 1 1 0	$6E00 – $6FFF
0 1 1 1	$7E00 – $7FFF
1 0 0 0	$8E00 – $8FFF
1 0 0 1	$9E00 – $9FFF
1 0 1 0	$AE00 – $AFFF
1 0 1 1	$BE00 – $BFFF
1 1 0 0	$CE00 – $CFFF
1 1 0 1	$DE00 – $DFFF
1 1 1 0	$EE00 – $EFFF
1 1 1 1	$FE00 – $FFFF

Bit 3 — Not implemented
Always reads one

NOCOP — COP System Disable
0 = COP system enabled (forces reset on time-out)
1 = COP system disabled

Bit 1 — Not implemented
Always reads one

EEON — EEPROM Enable
In single-chip modes EEON is forced to one (EEPROM enabled).
0 = 512 bytes of EEPROM is disabled from the memory map
1 = 512 bytes of EEPROM is present in the memory map

4.3.2.2 INIT Register

The internal registers used to control the operation of the MCU can be relocated on 4-Kbyte boundaries within the memory space with the use of INIT. This 8-bit special-purpose register can change the default locations of the RAM and control registers within the MCU memory map. It can be written only once within the first 64 E-clock cycles after a reset. It then becomes a read-only register.

INIT — RAM and I/O Mapping Register **$103D**

	Bit 7	6	5	4	3	2	1	Bit 0
	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0
RESET:	0	0	0	0	0	0	0	1

RAM[3:0] — RAM Map Position

These four bits, which specify the upper hexadecimal digit of the RAM address, control position of RAM in the memory map. RAM can be positioned at the beginning of any 4-Kbyte page in the memory map. Refer to **Table 4-5**.

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

These four bits specify the upper hexadecimal digit of the address for the 128-byte block of internal registers. The register block is positioned at the beginning of any 4-Kbyte page in the memory map. Refer to **Table 4-5**.

Table 4-5 RAM and Register Mapping

RAM[3:0]	Location
0000	$0000–$03FF
0001	$1000–$13FF
0010	$2000–$23FF
0011	$3000–$33FF
0100	$4000–$43FF
0101	$5000–$53FF
0110	$6000–$63FF
0111	$7000–$73FF
1000	$8000–$83FF
1001	$9000–$93FF
1010	$A000–$A3FF
1011	$B000–$B3FF
1100	$C000–$C3FF
1101	$D000–$D3FF
1110	$E000–$E3FF
1111	$F000–$F3FF

REG[3:0]	Location
0000	$0000–$005F
0001	$1000–$105F
0010	$2000–$205F
0011	$3000–$305F
0100	$4000–$405F
0101	$5000–$505F
0110	$6000–$605F
0111	$7000–$705F
1000	$8000–$805F
1001	$9000–$905F
1010	$A000–$A05F
1011	$B000–$B05F
1100	$C000–$C05F
1101	$D000–$D05F
1110	$E000–$E05F
1111	$F000–$F05F

When the memory map has the 96-byte register block mapped at the same location as RAM, the registers have priority and the lower 96 bytes of RAM are inaccessible. No harmful conflicts occur due to a hardware resource priority scheme. On-chip registers have the highest priority of all on-chip resources, followed by on-chip RAM, bootstrap ROM, and on-chip EEPROM.

4.3.2.3 OPTION Register

The 8-bit special-purpose OPTION register sets internal system configuration options during initialization. In single-chip and expanded modes (SMOD = 0), IRQE, DLY, FCME, and CR[1:0] can be written only once and only in the first 64 cycles after a reset. This minimizes the possibility of any accidental changes to the system configuration. In special test and bootstrap modes (SMOD = 1), these bits can be written at any time.

OPTION — System Configuration Options **$1039**

	Bit 7	6	5	4	3	2	1	Bit 0
	ADPU	CSEL	IRQE*	DLY*	CME	FCME*	CR1*	CR0*
RESET:	0	0	0	0	0	0	0	0

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

ADPU — A/D Power-Up

Refer to **SECTION 10 ANALOG-TO-DIGITAL CONVERTER**.

0 = A/D system disabled

1 = A/D system power enabled

CSEL — Clock Select

Selects alternate clock source for on-chip EEPROM and A/D charge pumps. On-chip RC clock should be used when E clock falls below 1 MHz. Refer to **SECTION 10 ANALOG-TO-DIGITAL CONVERTER**.

0 = A/D and EEPROM use system E clock

1 = A/D and EEPROM use internal RC clock

IRQE — Configure $\overline{\text{IRQ}}$ for Falling Edge-Sensitive Operation

Refer to **SECTION 5 RESETS AND INTERRUPTS**.

0 = Low level-sensitive operation.

1 = Falling edge-sensitive only operation.

DLY — Enable Oscillator Start-up Delay

Refer to **SECTION 5 RESETS AND INTERRUPTS**.

0 = The oscillator start-up delay coming out of STOP is bypassed and the MCU resumes processing within about four bus cycles.

1 = A delay of approximately 4000 E-clock cycles is imposed as the MCU is started up from the STOP power-saving mode.

CME — Clock Monitor Enable

In order to use both STOP and clock monitor, the CME bit must be written to zero before executing STOP, then written to one after recovering from STOP. Refer to **SECTION 5 RESETS AND INTERRUPTS**.

0 = Clock monitor disabled

1 = Clock monitor enabled

FCME — Force Clock Monitor Enable

When FCME equals one, slow or stopped clocks will cause a clock failure reset. To use STOP mode, FCME must always equal zero. Refer to **SECTION 5 RESETS AND INTERRUPTS**.

0 = Clock monitor follows state of CME bit

1 = Clock monitor enabled and cannot be disabled until next reset

CR[1:0] — COP Timer Rate Select Bits

These control bits determine a scaling factor for the watchdog timer. Refer to **SECTION 5 RESETS AND INTERRUPTS**.

4.3.2.4 OPT2 Register

The system configuration options 2 register (OPT2) controls three additional system options.

OPT2 — System Configuration Options 2 **$1038**

	Bit 7	6	5	4	3	2	1	Bit 0
	GWOM	CWOM	CLK4X	—	—	—	—	—
RESET:	0	0	0	0	0	0	0	0

GWOM — Port G Wired-OR Mode

Refer to **SECTION 6 PARALLEL INPUT/OUTPUT**.

0 = Port G operates normally.
1 = Port G outputs are open-drain type.

CWOM — Port C Wired-OR Mode

Refer to **SECTION 6 PARALLEL INPUT/OUTPUT**.

0 = Port C operates normally.
1 = Port C outputs are open-drain type.

CLK4X — 4XOUT Clock Enable

The 4XOUT signal, when enabled, is a buffered XTAL signal and is four times the frequency of the E-clock. This buffered clock is intended to synchronize external devices with the MCU. Refer to **SECTION 2 PIN DESCRIPTIONS**.

0 = The 4XOUT pin is driven low.
1 = The 4XOUT signal is driven on the 4XOUT pin.

Bits [4:0] — Not implemented

Always read zero

4.3.2.5 Block Protect Register (BPROT)

BPROT prevents accidental writes to EEPROM and the CONFIG register. The bits in this register can be written to zero during the first 64 E-clock cycles after reset in the normal modes. Once the bits are cleared to zero, the EEPROM array and the CONFIG register can be programmed or erased. Setting the bits in the BPROT register to logic one protects the EEPROM and CONFIG register until the next reset. Refer to **Table 4-6**.

BPROT — Block Protect **$1035**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	—	PTCON	BPRT3	BPRT2	BPRT1	BPRT0
RESET:	0	0	0	1	1	1	1	1

Bits [7:5] — Not implemented

Always read zero

PTCON — Protect for CONFIG

0 = CONFIG register can be programmed or erased normally
1 = CONFIG register cannot be programmed or erased

BPRT[3:0] — Block Protect Bits for EEPROM

0 = Protection disabled for associated block
1 = Protection enabled for associated block

Table 4-6 EEPROM Block Protection

Bit Name	Block Protected	Block Size
BPRT0	$xE00–$xE1F	32 Bytes
BPRT1	$xE20–$xE5F	64 Bytes
BPRT2	$xE60–$xEDF	128 Bytes
BPRT3	$xEE0–$xFFF	288 Bytes

4.4 EEPROM and CONFIG Register

The 512-byte EEPROM array and the single-byte CONFIG register are implemented with the same type of memory cells. The CONFIG register is a separate address located within the register block rather than in the EEPROM array. Unlike other registers within the register block, the CONFIG register can only be altered using the EEPROM programming procedure.

4.4.1 EEPROM

The 512-byte on-board EEPROM is initially located from $FE00 to $FFFF after reset in single-chip modes. It can be mapped to any other 4-Kbyte boundary by programming bits EE[3:0] in the CONFIG register. The EEPROM is enabled by the EEON bit in the CONFIG register. Programming and erasing is controlled by the PPROG register.

Unlike information stored in ROM, data in the 512 bytes of EEPROM can be erased and reprogrammed under software control. Because programming and erasing operations use an on-chip charge pump, a separate external power supply is not required.

Use of the block protect register (BPROT) prevents inadvertent writes to (or erases of) blocks of EEPROM. The CSEL bit in the OPTION register selects an on-chip oscillator clock for programming and erasing while operating at frequencies below 1 MHz.

4.4.1.1 EEPROM Programming

An exact register access sequence must be followed to allow successful programming and erasure of the EEPROM. The following procedures for modifying the EEPROM and CONFIG register detail the sequence. If an attempt is made to set both EELAT and EEPGM bits in the same write cycle and this attempt occurs before the required write cycle with the EELAT bit set, then neither bit is set. If a write to an EEPROM address is performed while the EEPGM bit is set, the write is ignored, and the programming operation in progress is not disturbed. If no EEPROM address is written between the point at which EELAT is set and EEPGM is set, then no program or erase operation occurs. These safeguards are included to prevent accidental EEPROM changes in cases of program runaway. If the frequency of the E clock is 1 MHz or less, the CSEL bit in the OPTION register must be set to select the internal RC clock.

When the EELAT bit in the PPROG register is cleared, the EEPROM can be read as if it were a ROM. The block protect register has no effect during reads. During EEPROM programming, the ROW and BYTE bits of PPROG are not used.

Recall that zeros must be erased by a separate erase operation before programming. The following example of how to program an EEPROM byte assumes that the appropriate bits in BPROT have been cleared and the data to be programmed is present in accumulator A.

```
PROG    LDAB    #$02     EELAT=1, EEPGM=0
        STAB    $103B    Set EELAT bit
        STAA    $FE00    Store data to EEPROM address
        LDAB    #$03     EELAT=1, EEPGM=1
        STAB    $103B    Turn on programming voltage
        JSR     DLY10    Delay 10 ms
        CLR     $103B    Turn off high voltage and set to READ mode
```

4.4.1.2 EEPROM Bulk Erase

To erase the EEPROM, ensure that the proper bits of the BPROT register are cleared, then complete the following steps using the PPROG register:

1. Write to PPROG with the ERASE, EELAT, and appropriate BYTE and ROW bits set.
2. Write to the appropriate EEPROM address with any data. Row erase only requires a write to any location in the row. Bulk erase is accomplished by writing to any location in the array.
3. Write to PPROG with ERASE, EELAT, EEPGM, and the appropriate BYTE and ROW bits set.
4. Delay for 10 ms or more, as appropriate.
5. Clear the EEPGM bit in PPROG to turn off the high voltage.
6. Return to step 1 for next byte or row or proceed to step 7.
7. Clear the PPROG register to reconfigure the EEPROM address and data buses for normal operation.

The following is an example of how to bulk erase the 512-byte EEPROM. The CONFIG register is not affected in this example. When bulk erasing the CONFIG register, CONFIG and the 512-byte array are all erased.

```
BULKE   LDAB    #$06     ERASE=1, EELAT=1, EEPGM=0
        STAB    $103B    Set EELAT bit
        STAB    $FE00    Store any data to any EEPROM address
        LDAB    #$07     EELAT=1, EEPGM=1
        STAB    $103B    Turn on programming voltage
        JSR     DLY10    Delay 10 ms
        CLR     $103B    Turn off high voltage and set to READ mode
```

4.4.1.3 EEPROM Row Erase

The following example shows how to perform a fast erase of large sections of EEPROM and assumes that index register X contains the address of a location in the desired row.

```
ROWE    LDAB    #$0E     ROW=1, ERASE=1, EELAT=1, EEPGM=0
        STAB    $103B    Set to ROW erase mode
        STAB    0,X      Store any data to any address in ROW
        LDAB    #$0F     ROW=1, ERASE=1, EELAT=1, EEPGM=1
        STAB    $103B    Turn on high voltage
        JSR     DLY10    Delay 10 ms
        CLR     $103B    Turn off high voltage and set to READ mode
```

4.4.1.4 EEPROM Byte Erase

The following is an example of how to erase a single byte of EEPROM and assumes that index register X contains the address of the byte to be erased.

```
BYTEE   LDAB    #$16     BYTE=1, ROW=0, ERASE=1, EELAT=1, EEPGM=0
        STAB    $103B    Set to BYTE erase mode
        STAB    0,X      Store any data to address to be erased
        LDAB    #$17     BYTE=1, ROW=0, ERASE=1, EELAT=1, EEPGM=1
        STAB    $103B    Turn on high voltage
        JSR     DLY10    Delay 10 ms
        CLR     $103B    Turn off high voltage and set to READ mode
```

4.4.2 PPROG EEPROM Programming Control Register

Bits in PPROG register control parameters associated with EEPROM programming.

PPROG — EEPROM Programming Control **$103B**

	Bit 7	6	5	4	3	2	1	Bit 0
	ODD	EVEN	—	BYTE	ROW	ERASE	EELAT	EEPGM
RESET:	0	0	0	0	0	0	0	0

ODD — Program Odd Rows in Half of EEPROM (TEST)

EVEN — Program Even Rows in Half of EEPROM (TEST)

Bit 5 — Not implemented
Always reads zero

BYTE — Byte/Other EEPROM Erase Mode
0 = Row or bulk erase mode used
1 = Erase only one byte of EEPROM

ROW — Row/All EEPROM Erase Mode (only valid when BYTE = 0)
0 = All 512 bytes of EEPROM erased
1 = Erase only one 16-byte row of EEPROM

Table 4-7 EEPROM Erase Mode Control

BYTE	ROW	Action
0	0	Bulk Erase (All 512 Bytes)
0	1	Row Erase (16 Bytes)
1	0	Byte Erase
1	1	Byte Erase

ERASE — Erase/Normal Control for EEPROM

Can be read or written any time.

0 = Normal read or program mode
1 = Erase mode

EELAT — EEPROM Latch Control

Can be read or written any time. When EELAT equals one, writes to EEPROM cause address and data to be latched.

0 = EEPROM address and data bus configured for normal reads
1 = EEPROM address and data bus configured for programming or erasing

EEPGM — EEPROM Program Command

Can be read any time. Can only be written while EELAT = 1.

0 = Program or erase voltage switched off to EEPROM array
1 = Program or erase voltage switched on to EEPROM array

4.4.3 CONFIG Register Programming

Because the CONFIG register is implemented with EEPROM cells, use EEPROM procedures to erase and program this register. The procedure for programming is the same as for programming a byte in the EEPROM array, except that the CONFIG register address is used. CONFIG can be programmed or erased (including byte erase) while the MCU is operating in any mode, provided that PTCON in BPROT is clear. To change the value in the CONFIG register, complete the following procedure. Do not initiate a reset until the procedure is complete. The new value will not take effect until after the next reset sequence.

1. Erase the CONFIG register.
2. Program the new value to the CONFIG address.
3. Initiate reset.

CONFIG — System Configuration Register **$103F**

	Bit 7	6	5	4	3	2	1	Bit 0	
	EE3	EE2	EE1	EE0	—	NOCOP	—	EEON	
RESET:	1	1	1	1	1	P	1	1	Single Chip
	1	1	1	1	1	P(L)	1	1	Bootstrap
	P	P	P	P	1	P	1	P	Expanded
	P	P	P	P	1	P(L)	1	0	Special Test

P indicates a previously programmed bit. P(L) indicates that the bit resets to the logic level held in the EEPROM bit prior to reset, but the function of COP is controlled by DISR bit in TEST1 register.

For a description of the bits contained in the CONFIG register refer to **4.3.2.1 CONFIG Register**.

4.5 Chip Selects

The function of the chip selects is to minimize the amount of external glue logic needed to interface the MCU to external devices. The MC68HC11F1 has four software configured chip selects that can be enabled in expanded modes. The chip selects for I/O (CSIO1 and CSIO2) are used for I/O expansion. The program chip select ($\overline{\text{CSPROG}}$) is used with an external memory that contains the program code and reset vectors. The general-purpose chip select (CSGEN) is the most flexible and is used to enable external devices.

Such factors as polarity, block size, base address and clock stretching can be controlled using the four chip-select control registers. When a port G pin is not used for chip select functions it can be used for general-purpose I/O.

When enabled, a chip select signal is asserted whenever the CPU makes an access to a designated range of addresses. Bus control signals and chip select signals are synchronous with the external E clock signal. For more information refer to Table A–7. Expansion Bus Timing in **APPENDIX A ELECTRICAL CHARACTERISTICS**. The length of the external E clock cycle to which the external device is synchronized can be stretched to accommodate devices that are slower than the MCU.

4.5.1 Program Chip Select

The program chip select ($\overline{\text{CSPROG}}$) is active in the range of memory where the main program exists. Refer to **Figure 4-3**.

When enabled, the $\overline{\text{CSPROG}}$ is active during address valid time and is an active-low signal. Although the general-purpose chip select has priority over the program chip select, $\overline{\text{CSPROG}}$ can be raised to a higher priority level by setting the GCSPR bit in CSCTL register. Bits in CSCTL enable the program chip select and determine its address range and priority level. Bits in CSSTRH select from zero to three clock cycles of delay.

4.5.2 I/O Chip Selects

The I/O chip selects (CSIO1 and CSIO2) are fixed in size and fill the remainder of the 4-Kbyte block occupied by the register block. CSIO1 is mapped at $x060–$x7FF and CSIO2 is mapped at $x800–$xFFF, where “x” corresponds to the high-order nibble of the register block base address, represented by the value contained in REG[3:0] in the INIT register.

Bits in the CSCTL register determine the polarity of the active state and enable both I/O chip selects. Bits in CSGSIZ select whether each chip select is active for address-valid or E-valid time. Bits in CSSTRH select from zero to three clock cycles of delay. Refer to **Figure 4-3**.

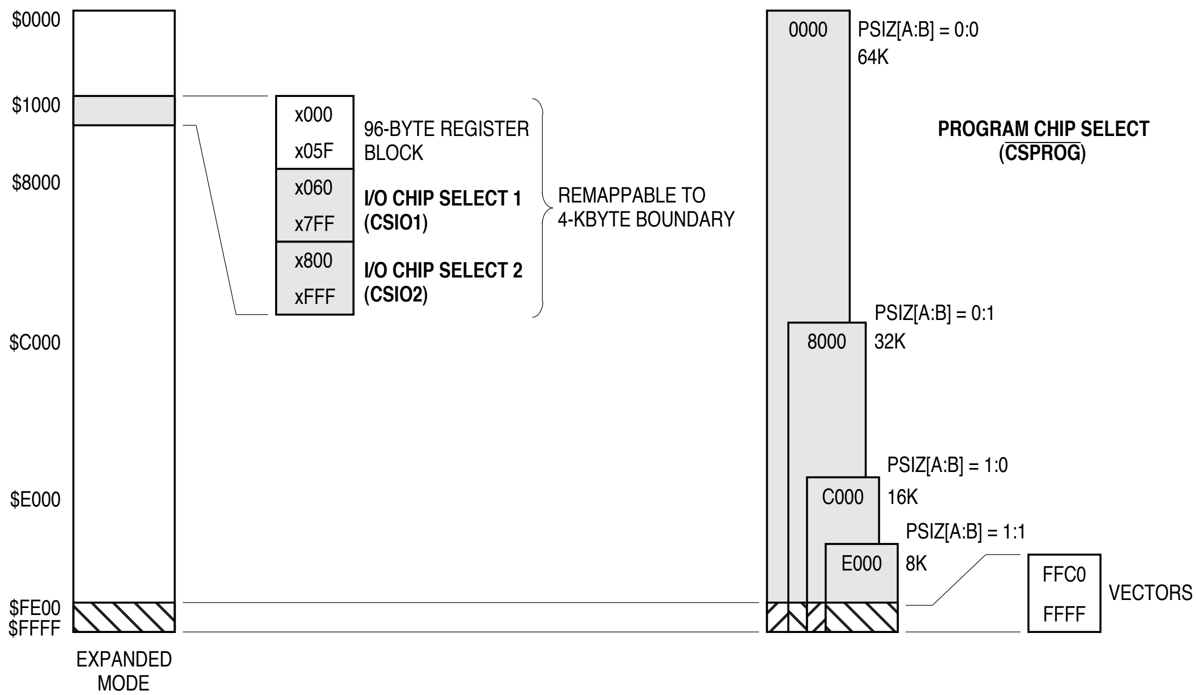


Figure 4-3 Address Map for I/O and Program Chip Selects

4.5.3 General-Purpose Chip Select

The general-purpose chip select (CSGEN) is the most flexible and has the most control bits. Polarity of the active state, E-valid or address-valid timing, size, starting address, and clock delay are all programmable.

A single bit in CSCTL selects a priority between CSGEN and $\overline{\text{CSPROG}}$. Bits in CSGSIZ select between address valid or E-clock valid timing, determine the polarity of the active state and the address range of CSGEN. The value contained in the CSGADR register determines the starting address for CSGEN. Depending on the size selected for CSGEN, some bits in CSGADR will be invalid (don't cares). Note that CSGEN is disabled when a size of zero is selected. Refer to **Figure 4-4**.

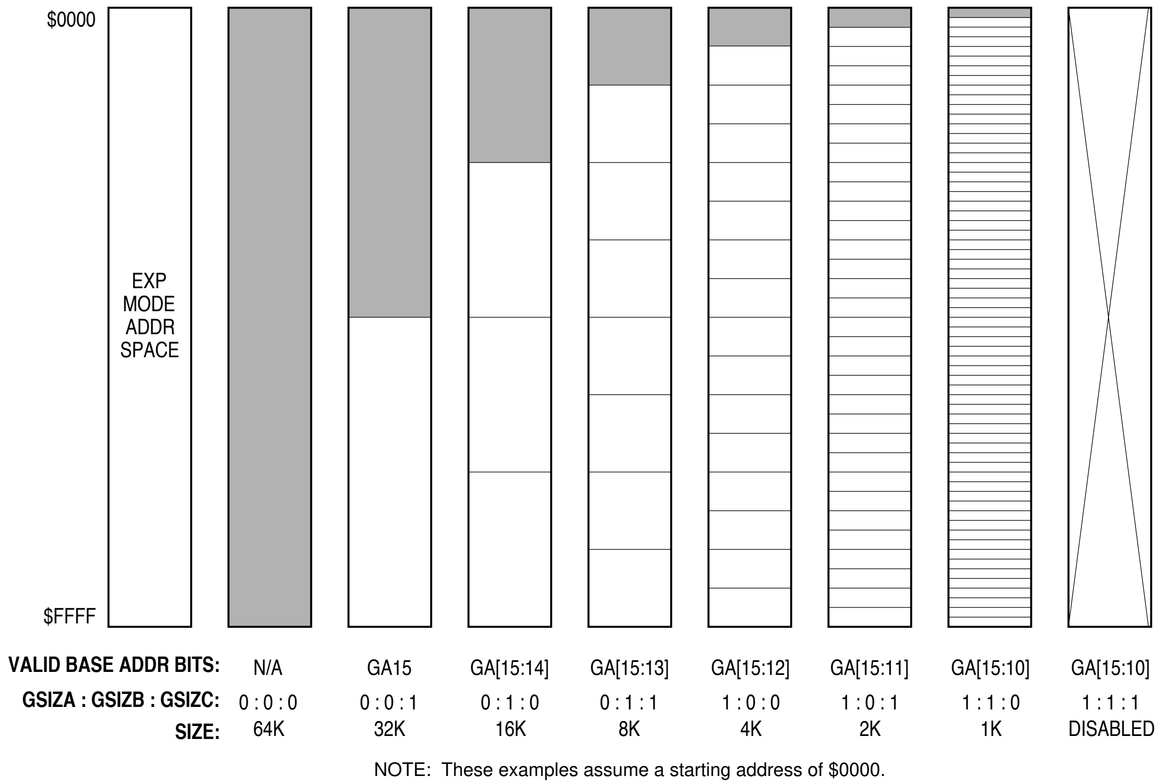


Figure 4-4 Address Map for General-Purpose Chip Select

CSSTRH — Chip Select Clock Stretch Select **$105C**

	Bit 7	6	5	4	3	2	1	Bit 0
	IO1SA	IO1SB	IO2SA	IO2SB	GSTHA	GSTHB	PSTHA	PSTHB
RESET:	0	0	0	0	0	0	0	0

Table 4-8 Chip Select Clock Stretch Control

Bit A	Bit B	Clock Stretch Selected
0	0	None
0	1	1 cycle
1	0	2 cycles
1	1	3 cycles

IO1SA–IO1SB — I/O Chip Select 1 Clock Stretch Select
Refer to **Table 4-8**.

IO2SA–IO2SB — I/O Chip Select 2 Clock Stretch Select
Refer to **Table 4-8**.

GSTHA–GSTHB — General-Purpose Chip Select Clock Stretch Select
Refer to **Table 4-8**.

PSTHA–PSTHB — Program Chip Select Clock Stretch Select
Refer to **Table 4-8**.

CSCTL — Chip Select Control **$105D**

	Bit 7	6	5	4	3	2	1	Bit 0
	IO1EN	IO1PL	IO2EN	IO2PL	GCSPR	PCSEN*	PSIZA	PSIZB
RESET:	0	0	0	0	0	—	0	0

*PCSEN is set out of reset in expanded modes and cleared in single-chip modes.

IO1EN — I/O Chip Select 1 Enable
- 0 = CSIO1 is disabled and port G bit 5 is general-purpose I/O.
- 1 = CSIO1 is enabled and uses port G bit 5.

IO1PL — I/O Chip Select 1 Polarity Select
- 0 = CSIO1 active low
- 1 = CSIO1 active high

IO2EN — I/O Chip Select 2 Enable
- 0 = CSIO2 is disabled and port G bit 4 is general-purpose I/O.
- 1 = CSIO2 is enabled and uses port G bit 4.

IO2PL — I/O Chip Select 2 Polarity Select
- 0 = CSIO2 active low
- 1 = CSIO2 active high

GCSPR — General-Purpose Chip Select Priority
- 0 = Program chip select has priority over general-purpose chip select
- 1 = General-purpose chip select has priority over program chip select

PCSEN — Program Chip Select Enable

This bit is set out of reset in expanded modes and cleared in single-chip modes.
- 0 = CSPROG disabled and port G bit 7 available as general-purpose I/O
- 1 = CSPROG enabled out of reset and uses port G bit 7 pin

PSIZA, PSIZB — Program Chip Select Size (A or B)

Table 4-9 Program Chip Select Size Control

PSIZA	PSIZB	Size (Bytes)	Address Range
0	0	64 K	$0000–$FFFF
0	1	32 K	$8000–$FFFF
1	0	16 K	$C000–$FFFF
1	1	8 K	$E000–$FFFF

CSGADR — General-Purpose Chip Select Address Register **$105E**

	Bit 7	6	5	4	3	2	1	Bit 0
	GA15	GA14	GA13	GA12	GA11	GA10	—	—
RESET:	0	0	0	0	0	0	0	0

GA[15:10] — General-Purpose Chip Select Base Address

GA[15:10] correspond to MCU address bits ADDR[15:10] and select the starting address of the general-purpose chip select's address range. Which bits are valid depends upon the size selected by GSIZA–GSIZC in CSGSIZ register. Refer to the following table and to **Figure 4-4**.

Table 4-10 General-Purpose Chip Select Starting Address

Size (Bytes)	Valid Starting Address Bits
0 K (Disabled)	None
1 K	GA[15:10]
2 K	GA[15:11]
4 K	GA[15:12]
8 K	GA[15:13]
16 K	GA[15:14]
32 K	GA15
64 K	None

CSGSIZ — General-Purpose Chip Select Size Control **$105F**

	Bit 7	6	5	4	3	2	1	Bit 0
	IO1AV	IO2AV	—	GNPOL	GAVLD	GSIZA	GSIZB	GSIZC
RESET:	0	0	0	0	0	1	1	1

IO1AV — I/O Chip Select 1 Address Valid

0 = I/O chip select 1 is active during E-clock valid time (E-clock high)

1 = I/O chip select 1 is active during address valid time

IO2AV — I/O Chip Select 2 Address Valid

0 = I/O chip select 1 is active during E-clock valid time (E-clock high)

1 = I/O chip select 1 is active during address valid time

GNPOL — General-Purpose Chip Select Polarity Select

0 = CSGEN is active low

1 = CSGEN is active high

GAVLD — General-Purpose Chip Select Address Valid Select

0 = CSGEN is valid during E-clock valid time (E-clock high)

1 = CSGEN is valid during address valid time

G1SZA–G1SZC — General-Purpose Chip Select Size

Refer to **Table 4-11**.

Table 4-11 General-Purpose Chip Select Size Control

GSIZA	GSIZB	GSIZC	Size (Bytes)
0	0	0	64 K
0	0	1	32 K
0	1	0	16 K
0	1	1	8 K
1	0	0	4 K
1	0	1	2 K
1	1	0	1 K
1	1	1	0 K (Disabled)

Table 4-12 Chip Select Control Parameter Summary

CSIO1	Enable	IO1EN in CSCTL —	1 = On, off at reset (0)
	Valid	IO1AV in CSGSIZ —	1 = Address valid, 0 = E valid
	Polarity	IO1PL in CSCTL —	1 = Active high, 0 = Active low
	Size	Fixed —	($x060–$x7FF)
	Start Address	$x060 —	"x" is determined by REG[3:0] in INIT
	Stretch	IO1SA–IO1SB in CSSTRH —	0, 1, 2, or 3 E clocks

CSIO2	Enable	IO2EN in CSCTL —	1 = On, off at reset (0)
	Valid	IO2AV in CSGSIZ —	1 = Address valid, 0 = E valid
	Polarity	IO2PL in CSCTL —	1 = Active high, 0 = Active low
	Size	Fixed —	($x800–$xFFF)
	Start Address	$x800 —	"x" is determined by REG[3:0] in INIT
	Stretch	IO2SA–IO2SB in CSSTRH —	0, 1, 2, or 3 E clocks

$\overline{\text{CSPROG}}$	Enable	PCSEN in CSCTL —	1 = On, on after reset in expanded modes off after reset in single-chip modes
	Valid	Fixed (Address valid)	
	Polarity	Fixed (Active low)	
	Size	PSIZA–PSIZB — in CSCTL	0:0 = 64K ($0000–$FFFF) 0:1 = 32K ($8000–$FFFF) 1:0 = 16K ($C000–$FFFF) 1:1 = 8K ($E000–$FFFF)
	Start Address	Fixed (determined by size)	
	Stretch	PSTHA–PSTHB in CSSTRH —	0, 1, 2, or 3 E clocks 1 cycle after reset in expanded mode no delay after reset in all other modes
	Priority	GCSPR in CSCTL —	1 = CSGEN above $\overline{\text{CSPROG}}$ 0 = $\overline{\text{CSPROG}}$ above CSGEN

CSGEN	Enable	Set size to 0K to disable —	1 = CSGEN above $\overline{\text{CSPROG}}$ 0 = $\overline{\text{CSPROG}}$ above CSGEN
	Valid	GAVLD in CSGSIZ —	Address valid or E valid
	Polarity	GNPOL in CSGSIZ —	Active high or low
	Size	GSIZA–GSIZC in CSGSIZ —	Refer to Table 4–12
	Start Address	GA[15:10] in CSGADR	
	Stretch	GSTHA–GSTHB in CSSTRH —	0, 1, 2, or 3 E clocks

SECTION 5 RESETS AND INTERRUPTS

Resets and interrupt operations load the program counter with a vector that points to a new location from which instructions are to be fetched. A reset causes the internal control registers to be initialized to a known state. The program counter is loaded with a known starting address and execution of instructions begins. An interrupt temporarily suspends normal program execution while an interrupt service routine is being executed. After an interrupt has been serviced, the main program resumes as if there had been no interruption.

5.1 Resets

There are four possible sources of reset. Power-on reset (POR) and external reset share the normal reset vector. The computer operating properly (COP) reset and the clock monitor reset each has its own vector.

5.1.1 Power-On Reset

A positive transition on V_{DD} generates a power-on reset (POR), which is used only for power-up conditions. POR cannot be used to detect drops in power supply voltages. A 4064 t_{cyc} (internal clock cycle) delay after the oscillator becomes active allows the clock generator to stabilize. If $\overline{\text{RESET}}$ is at logical zero at the end of 4064 t_{cyc}, the CPU remains in the reset condition until $\overline{\text{RESET}}$ goes to logical one.

It is important to protect the MCU during power transitions. To protect data in EEPROM, M68HC11 systems need an external circuit that holds the $\overline{\text{RESET}}$ pin low whenever V_{DD} is below the minimum operating level. This external voltage level detector, or other external reset circuits, are the usual source of reset in a system. The POR circuit only initializes internal circuitry during cold starts. Refer to Figure 2–3.

5.1.2 External Reset ($\overline{\text{RESET}}$)

The CPU distinguishes between internal and external reset conditions by sensing whether the reset pin rises to a logic one in less than two E-clock cycles after an internal device releases reset. When a reset condition is sensed, the $\overline{\text{RESET}}$ pin is driven low by an internal device for four E-clock cycles, then released. Two E-clock cycles later it is sampled. If the pin is still held low, the CPU assumes that an external reset has occurred. If the pin is high, it indicates that the reset was initiated internally by either the COP system or the clock monitor. It is not advisable to connect an external resistor capacitor (RC) power-up delay circuit to the reset pin of M68HC11 devices because the circuit charge time constant can cause the device to misinterpret the type of reset that occurred.

5.1.3 Computer Operating Properly (COP) Reset

The MCU includes a COP system to help protect against software failures. When the COP is enabled, the software is responsible for keeping a free-running watchdog timer from timing out. When the software is no longer being executed in the intended sequence, a system reset is initiated.

The state of the NOCOP bit in the CONFIG register determines whether the COP system is enabled or disabled. To change the enable status of the COP system, change the contents of the CONFIG register and then perform a system reset. In the special test and bootstrap operating modes, the COP system is initially inhibited by the disable resets (DISR) control bit in the TEST1 register. The DISR bit can subsequently be written to zero to enable COP resets.

The COP timer rate control bits CR[1:0] in the OPTION register determine the COP time-out period. The system E clock is divided by the values shown in **Table 5-1**. After reset, these bits are zero, which selects the fastest time-out period. In normal operating modes, these bits can only be written once within 64 bus cycles after reset.

Table 5-1 COP Timer Rate Selection

CR[1:0]	Divide E By	XTAL = 8.0 MHz Time-out –0 ms, +16.4 ms	XTAL = 12.0 MHz Time-out –0 ms, +10.9 ms	XTAL = 16.0 MHz Time-out –0 ms, +8.2 ms
0 0	2^{15}	16.384 ms	10.923 ms	8.192 ms
0 1	2^{17}	65.536 ms	43.691 ms	32.768 ms
1 0	2^{19}	262.14 ms	174.76 ms	131.07 ms
1 1	2^{21}	1.049 s	699.05 ms	524.29 ms
	E =	2.0 MHz	3.0 MHz	4.0 MHz

COPRST — Arm/Reset COP Timer Circuitry **$103A**

	Bit 7	6	5	4	3	2	1	Bit 0
	7	6	5	4	3	2	1	0
RESET:	0	0	0	0	0	0	0	0

Complete the following reset sequence to service the COP timer. Write $55 to COPRST to arm the COP timer clearing mechanism. Then write $AA to COPRST to clear the COP timer. Performing instructions between these two steps is possible as long as both steps are completed in the correct sequence before the timer times out.

5.1.4 Clock Monitor Reset

The clock monitor circuit is based on an internal RC time delay. If no MCU clock edges are detected within this RC time delay, the clock monitor can optionally generate a system reset. The clock monitor function is enabled or disabled by the CME and FCME control bits in the OPTION register. The presence of a time-out is determined by the RC delay, which allows the clock monitor to operate without any MCU clocks.

Clock monitor is used as a backup for the COP system. Because the COP needs a clock to function, it is disabled when the clocks stop. Therefore, the clock monitor system can detect clock failures not detected by the COP system.

Semiconductor wafer processing causes variations of the RC time-out values between individual devices. An E-clock frequency below 10 kHz is detected as a clock monitor error. An E-clock frequency of 200 kHz or more prevents clock monitor errors. Using the clock monitor function when the E-clock is below 200 kHz is not recommended.

Special considerations are needed when a STOP instruction is executed and the clock monitor is enabled. Because the STOP function causes the clocks to be halted, the clock monitor function generates a reset sequence if it is enabled at the time the STOP mode was initiated. Before executing a STOP instruction, clear to zero the CME bit in the OPTION register to disable the clock monitor. After recovery from STOP, set the CME bit to logic one to enable the clock monitor.

5.1.5 OPTION Register

OPTION — System Configuration Options **$1039**

	Bit 7	6	5	4	3	2	1	Bit 0
	ADPU	CSEL	IRQE*	DLY*	CME	FCME*	CR1*	CR0*
RESET:	0	0	0	1	0	0	0	0

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

ADPU — Analog-to-Digital Converter Power-Up
Refer to **SECTION 10 ANALOG-TO-DIGITAL CONVERTER**.

CSEL — Clock Select
Refer to **SECTION 10 ANALOG-TO-DIGITAL CONVERTER**.

IRQE — Configure $\overline{\text{IRQ}}$ for Edge-Sensitive Only Operation
0 = Low level sensitive operation.
1 = Falling edge sensitive only operation.

DLY — Enable Oscillator Start-up Delay
0 = The oscillator start-up delay coming out of STOP is bypassed and the MCU resumes processing within about four bus cycles.
1 = A delay of approximately 4000 E-clock cycles is imposed as the MCU is started up from the STOP power-saving mode.

CME — Clock Monitor Enable
This control bit can be read or written at any time and controls whether or not the internal clock monitor circuit triggers a reset sequence when the system clock is slow or absent. When it is clear, the clock monitor circuit is disabled, and when it is set, the clock monitor circuit is enabled. Reset clears the CME bit.

FCME — Force Clock Monitor Enable
To use STOP mode, the FCME bit must equal zero.
0 = Clock monitor follows the state of the CME bit.
1 = Clock monitor circuit is enabled until next reset

CR[1:0] — COP Timer Rate Select

The internal E clock is first divided by 2^{15} before it enters the COP watchdog system. These control bits determine a scaling factor for the watchdog timer. Refer to **Table 5-1**.

5.1.6 CONFIG Register

CONFIG — System Configuration Register **$103F**

	Bit 7	6	5	4	3	2	1	Bit 0	
	EE3	EE2	EE1	EE0	—	NOCOP	—	EEON	
RESET:	1	1	1	1	1	P	1	1	Single Chip
	1	1	1	1	1	P(L)	1	1	Bootstrap
	P	P	P	P	1	P	1	P	Expanded
	P	P	P	P	1	P(L)	1	0	Special Test

P indicates a previously programmed bit. P(L) indicates that the bit resets to the logic level held in the latch prior to reset, but the function of COP is controlled by DISR in TEST1 register.

EE[3:0] — EEPROM Mapping Control

Refer to **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY**.

Bit 3 — Not implemented

Always reads one

NOCOP — COP System Disable

0 = COP system enabled (forces reset on time-out)

1 = COP system disabled

Bit 1 — Not implemented

Always reads one

EEON — EEPROM Enable

Refer to **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY**.

5.2 Effects of Reset

When a reset condition is recognized, the internal registers and control bits are forced to an initial state. Depending on the cause of the reset and the operating mode, the reset vector can be fetched from any of six possible locations. Refer to **Table 5-2**.

Table 5-2 Reset Cause, Operating Mode, and Reset Vector

Cause of Reset	Normal Mode Vector	Special Test or Bootstrap
POR or RESET Pin	$FFFE, FFFF	$BFFE, $BFFF
Clock Monitor Failure	$FFFC, FFFD	$BFFC, $BFFD
COP Watchdog Time-out	$FFFA, FFFB	$BFFA, $BFFB

These initial states then control on-chip peripheral systems to force them to known start-up states, as follows:

5.2.1 Central Processing Unit

After reset, the CPU fetches the reset vector from the appropriate address during the first three cycles, and begins executing instructions. The stack pointer and other CPU registers are indeterminate immediately after reset; however, the X and I interrupt mask bits in the condition code register (CCR) are set to mask any interrupt requests. Also, the S bit in the CCR is set to inhibit the STOP mode.

5.2.2 Memory Map

After reset, the INIT register is initialized to $01, putting the 1024 bytes of RAM at locations $0000 through $03FF, and the control registers at locations $1000 through $105F. The EE[3:0] bits in the CONFIG register control the location of the 512-byte EEPROM array.

5.2.3 Parallel I/O

When a reset occurs in expanded operating modes, port B, C, and F pins used for parallel I/O are dedicated to the expansion bus. If a reset occurs during a single-chip operating mode, all ports are configured as general-purpose high-impedance inputs.

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. Port data registers reflect the port's functional state at reset. The pin function is mode dependent.

5.2.4 Timer

During reset, the timer system is initialized to a count of $0000. The prescaler bits are cleared, and all output compare registers are initialized to $FFFF. All input capture registers are indeterminate after reset. The output compare 1 mask (OC1M) register is cleared so that successful OC1 compares do not affect any I/O pins. The other four output compares are configured so that they do not affect any I/O pins on successful compares. All input capture edge-detector circuits are configured for capture disabled operation. The timer overflow interrupt flag and all eight timer function interrupt flags are cleared. All nine timer interrupts are disabled because their mask bits have been cleared.

The I4/O5 bit in the PACTL register is cleared to configure the I4/O5 function as OC5; however, the OM5–OL5 control bits in the TCTL1 register are clear so OC5 does not control the PA3 pin.

5.2.5 Real-Time Interrupt (RTI)

The real-time interrupt flag (RTIF) is cleared and automatic hardware interrupts are masked. The rate control bits are cleared after reset and can be initialized by software before the real-time interrupt (RTI) system is used.

5.2.6 Pulse Accumulator

The pulse accumulator system is disabled at reset so that the pulse accumulator input (PAI) pin defaults to being a general-purpose input pin.

5.2.7 Computer Operating Properly (COP)

The COP watchdog system is enabled if the NOCOP control bit in the CONFIG register is cleared, and disabled if NOCOP is set. The COP rate is set for the shortest duration time-out.

5.2.8 Serial Communications Interface (SCI)

The reset condition of the SCI system is independent of the operating mode. All transmit and receive interrupts are masked and both the transmitter and receiver are disabled so the port pins default to being general-purpose I/O lines. The SCI frame format is initialized to an 8-bit character size. The send break and receiver wakeup functions are disabled. The TDRE and TC status bits in the SCI status register are both set, indicating that there is no transmit data in either the transmit data register or the transmit serial shift register. The RDRF, IDLE, OR, NF, FE, PF, and RAF receive-related status bits are cleared.

5.2.9 Serial Peripheral Interface (SPI)

The SPI system is disabled by reset. The port pins associated with this function default to being general-purpose I/O lines.

5.2.10 Analog-to-Digital Converter

The A/D converter configuration is indeterminate after reset. The ADPU bit is cleared by reset, which disables the A/D system. The conversion complete flag is cleared by reset.

5.2.11 System

The EEPROM programming controls are disabled, so the memory system is configured for normal read operation. PSEL[3:0] are initialized with the binary value %0101, causing the external $\overline{\text{IRQ}}$ pin to have the highest I-bit interrupt priority. The $\overline{\text{IRQ}}$ pin is configured for level-sensitive operation (for wired-OR systems). The RBOOT, SMOD, and MDA bits in the HPRIO register reflect the status of the MODB and MODA inputs at the rising edge of reset. The DLY control bit is set to specify that an oscillator start-up delay is imposed upon recovery from STOP mode. The clock monitor system is disabled because CME and FCME are cleared.

5.3 Reset and Interrupt Priority

Resets and interrupts have a hardware priority that determines which reset or interrupt is serviced first when simultaneous requests occur. Any maskable interrupt can be given priority over other maskable interrupts.

The first six interrupt sources are not maskable. The priority arrangement for these sources is as follows:

1. POR or $\overline{\text{RESET}}$ pin
2. Clock monitor reset
3. COP watchdog reset
4. $\overline{\text{XIRQ}}$ interrupt
5. Illegal opcode interrupt
6. Software interrupt (SWI)

The maskable interrupt sources have the following priority arrangement:

1. $\overline{\text{IRQ}}$
2. Real-time interrupt
3. Timer input capture 1
4. Timer input capture 2
5. Timer input capture 3
6. Timer output compare 1
7. Timer output compare 2
8. Timer output compare 3
9. Timer output compare 4
10. Timer input capture 4/output compare 5
11. Timer overflow
12. Pulse accumulator overflow
13. Pulse accumulator input edge
14. SPI transfer complete
15. SCI system (refer to **Figure 5-5**)

Any one of these interrupts can be assigned the highest maskable interrupt priority by writing the appropriate value to the PSEL bits in the HPRIO register. Otherwise, the priority arrangement remains the same. An interrupt that is assigned highest priority is still subject to global masking by the I bit in the CCR, or by any associated local bits. Interrupt vectors are not affected by priority assignment. To avoid race conditions, HP-RIO can only be written while I-bit interrupts are inhibited.

5.3.1 Highest Priority Interrupt and Miscellaneous Register

HPRIO — Highest Priority I-Bit Interrupt and Miscellaneous **$103C**

	Bit 7	6	5	4	3	2	1	Bit 0	
	RBOOT*	SMOD*	MDA*	IRV	PSEL3	PSEL2	PSEL1	PSEL0	
RESET:	0	0	0	0	0	1	0	1	Single Chip
	0	0	1	1	0	1	0	1	Expanded
	1	1	0	0	0	1	0	1	Bootstrap
	0	1	1	1	0	1	0	1	Special Test

*The values of the RBOOT, SMOD, MDA, and IRV reset bits depend on the operating mode selected during power-up. Refer to Table 4–3.

RBOOT — Read Bootstrap ROM

Set to one out of reset in bootstrap mode. Valid while in special modes only. Can be read any time. Can only be written in special modes. Refer to **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY** for more information.

SMOD — Special Mode Select

Can be read any time. Can only be written in special modes (SMOD = 1). Can only be written to zero. Refer to **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY** for more information.

MDA — Mode Select A

Can be read any time. Can only be written in special modes (SMOD = 1). Refer to **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY** for more information.

IRV — Internal Read Visibility

The IRV control bit allows internal read accesses to be available on the external data bus during operation in expanded modes. In special modes (SMOD = 1), IRV resets to one (enabled) and can be written any time. In normal modes (SMOD = 0), IRV resets to zero (disabled) and only **one** write is allowed.

PSEL[3:0] — Priority Select Bits

These bits select one interrupt source to be elevated above all other I-bit-related sources and can only be written while the I bit in the CCR is set (interrupts disabled).

Table 5-3 Highest Priority Interrupt Selection

PSEL3	PSEL2	PSEL1	PSEL0	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	$\overline{\text{IRQ}}$
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 Output Compare 5/Input Capture 4

5.4 Interrupts

The MCU has 18 interrupt vectors that support 22 interrupt sources. The 15 maskable interrupts are generated by on-chip peripheral systems. These interrupts are recognized when the global interrupt mask bit (I) in the condition code register (CCR) is clear. The three non-maskable interrupt sources are illegal opcode trap, software interrupt, and $\overline{\text{XIRQ}}$ pin. Refer to **Table 5-4**, which shows the interrupt sources and vector assignments for each source.

Table 5-4 Interrupt and Reset Vector Assignments

Vector Address	Interrupt Source	CCR Mask Bit	Local Mask
FFC0, C1 – FFD4, D5	Reserved	—	—
FFD6, D7	SCI Serial System	I	
	• SCI Receive Data Register Full		RIE
	• SCI Receiver Overrun		RIE
	• SCI Transmit Data Register Empty		TIE
	• SCI Transmit Complete		TCIE
	• SCI Idle Line Detect		ILIE
FFD8, D9	SPI Serial Transfer Complete	I	SPIE
FFDA, DB	Pulse Accumulator Input Edge	I	PAII
FFDC, DD	Pulse Accumulator Overflow	I	PAOVI
FFDE, DF	Timer Overflow	I	TOI
FFE0, E1	Timer Input Capture 4/Output Compare 5	I	I4/O5I
FFE2, E3	Timer Output Compare 4	I	OC4I
FFE4, E5	Timer Output Compare 3	I	OC3I
FFE6, E7	Timer Output Compare 2	I	OC2I
FFE8, E9	Timer Output Compare 1	I	OC1I
FFEA, EB	Timer Input Capture 3	I	IC3I
FFEC, ED	Timer Input Capture 2	I	IC2I
FFEE, EF	Timer Input Capture 1	I	IC1I
FFF0, F1	Real-Time Interrupt	I	RTII
FFF2, F3	$\overline{\text{IRQ}}$	I	None
FFF4, F5	$\overline{\text{XIRQ}}$ Pin	X	None
FFF6, F7	Software Interrupt	None	None
FFF8, F9	Illegal Opcode Trap	None	None
FFFA, FB	COP Failure	None	NOCOP
FFFC, FD	Clock Monitor Fail	None	CME
FFFE, FF	$\overline{\text{RESET}}$	None	None

For some interrupt sources, such as the 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 consisting of 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.

5.4.1 Interrupt Recognition and Register Stacking

An interrupt can be recognized at any time after it is enabled by its local mask, if any, and by the global mask bit in the CCR. Once an interrupt source is recognized, the CPU responds at the completion of the instruction being executed. Interrupt latency varies according to the number of cycles required to complete the current instruction. When the CPU begins to service an interrupt, the contents of the CPU registers are pushed onto the stack in the order shown in **Table 5-5**. After the CCR value is stacked, the I bit and the X bit (if $\overline{\text{XIRQ}}$ is pending) are set to inhibit further interrupts. The interrupt vector for the highest priority pending source is fetched, and execution continues at the address specified by the vector. At the end of the interrupt service routine, the return from interrupt instruction is executed and the saved registers are pulled from the stack in reverse order so that normal program execution can resume. Refer to **SECTION 3 CENTRAL PROCESSING UNIT** for further information.

Table 5-5 Stacking Order on Entry to Interrupts

Memory Location	CPU Registers
SP	PCL
SP – 1	PCH
SP – 2	IYL
SP – 3	IYH
SP – 4	IXL
SP – 5	IXH
SP – 6	ACCA
SP – 7	ACCB
SP – 8	CCR

5.4.2 Non-Maskable Interrupt Request ($\overline{\textbf{XIRQ}}$)

Non-maskable interrupts are useful because they can always interrupt CPU operations. The most common use for such an interrupt is for serious system problems, such as program runaway or power failure. The $\overline{\text{XIRQ}}$ input is an updated version of the $\overline{\text{NMI}}$ input of earlier MCUs.

Upon reset, both the X bit and I bit of the CCR are set to inhibit all maskable interrupts and $\overline{\text{XIRQ}}$. After minimum system initialization, software can clear the X bit by a TAP instruction, enabling $\overline{\text{XIRQ}}$ interrupts. Thereafter, software cannot set the X bit. Thus, an $\overline{\text{XIRQ}}$ interrupt is a nonmaskable interrupt. Because the operation of the I-bit-related interrupt structure has no effect on the X bit, the internal $\overline{\text{XIRQ}}$ pin remains nonmasked. In the interrupt priority logic, the $\overline{\text{XIRQ}}$ interrupt has a higher priority than any source that is maskable by the I bit. All I-bit-related interrupts operate normally with their own priority relationship.

When an I-bit-related interrupt occurs, the I bit is automatically set by hardware after stacking the CCR byte. The X bit is not affected. When an X-bit-related interrupt occurs, both the X and I bits are automatically set by hardware after stacking the CCR. A return from interrupt instruction restores the X and I bits to their pre-interrupt request state.

5.4.3 Illegal Opcode Trap

Because not all possible opcodes or opcode sequences are defined, the MCU includes an illegal opcode detection circuit, which generates an interrupt request. When an illegal opcode is detected and the interrupt is recognized, the current value of the program counter is stacked. After interrupt service is complete, reinitialize the stack pointer so repeated execution of illegal opcodes does not cause stack underflow. Left uninitialized, the illegal opcode vector can point to a memory location that contains an illegal opcode. This condition causes an infinite loop that causes stack underflow. The stack grows until the system crashes.

The illegal opcode trap mechanism works for all unimplemented opcodes on all four opcode map pages. The address stacked as the return address for the illegal opcode interrupt is the address of the first byte of the illegal opcode. Otherwise, it would be almost impossible to determine whether the illegal opcode had been one or two bytes. The stacked return address can be used as a pointer to the illegal opcode so the illegal opcode service routine can evaluate the offending opcode.

5.4.4 Software Interrupt

SWI is an instruction, and thus cannot be interrupted until complete. SWI is not inhibited by the global mask bits in the CCR. Because execution of SWI sets the I mask bit, once an SWI interrupt begins, other interrupts are inhibited until SWI is complete, or until user software clears the I bit in the CCR.

5.4.5 Maskable Interrupts

The maskable interrupt structure of the MCU can be extended to include additional external interrupt sources through the $\overline{\text{IRQ}}$ pin. The default configuration of this pin is a low-level sensitive wired-OR network. When an event triggers an interrupt, a software accessible interrupt flag is set. When enabled, this flag causes a constant request for interrupt service. After the flag is cleared, the service request is released.

5.4.6 Reset and Interrupt Processing

Figure 5-1 and **Figure 5-3** illustrate the reset and interrupt process. **Figure 5-1** illustrates how the CPU begins from a reset and how interrupt detection relates to normal opcode fetches. **Figure 5-3** is an expansion of a block in **Figure 5-1** and illustrates interrupt priorities. **Figure 5-5** shows the resolution of interrupt sources within the SCI subsystem.

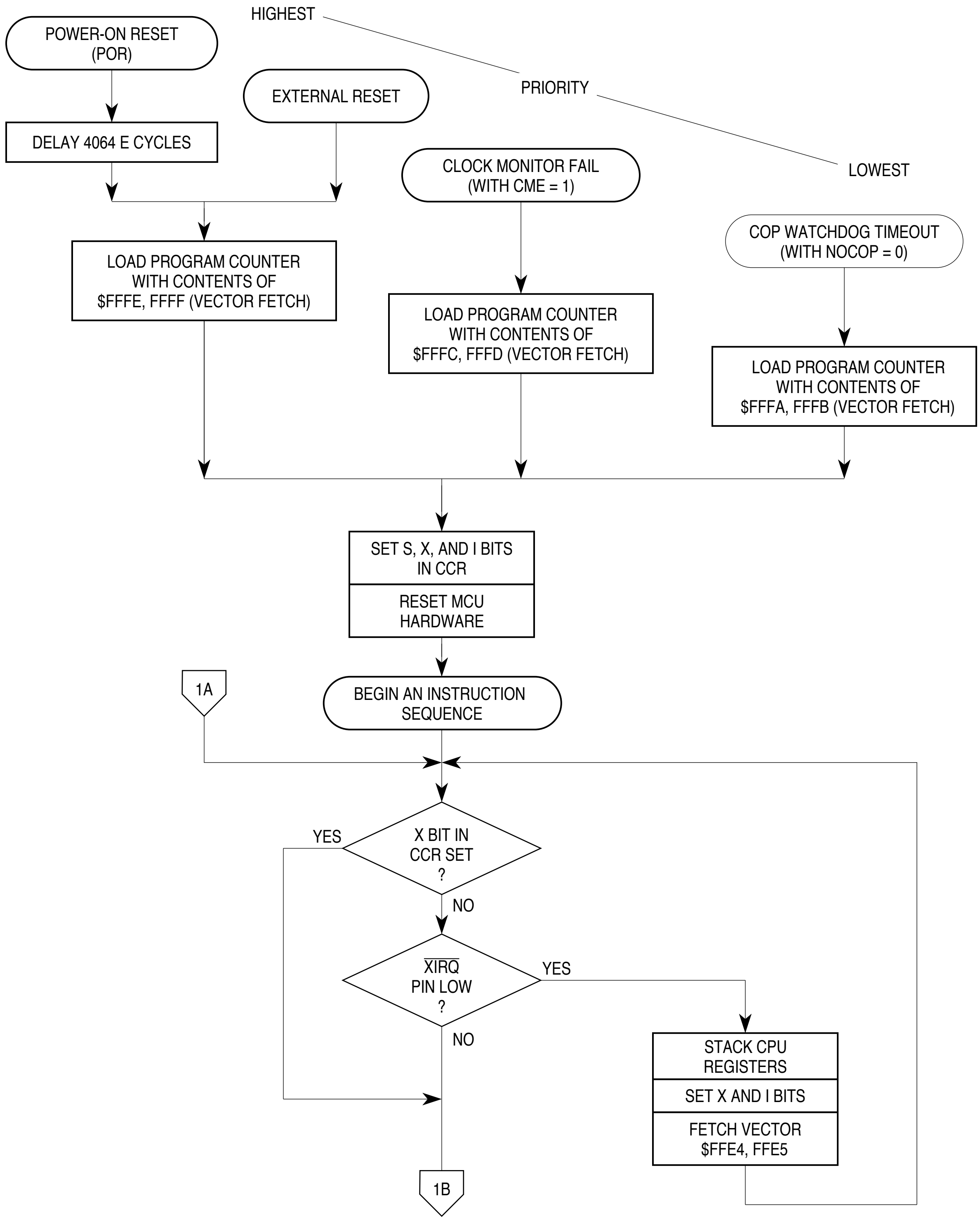


Figure 5-1 Processing Flow Out of Reset (1 of 2)

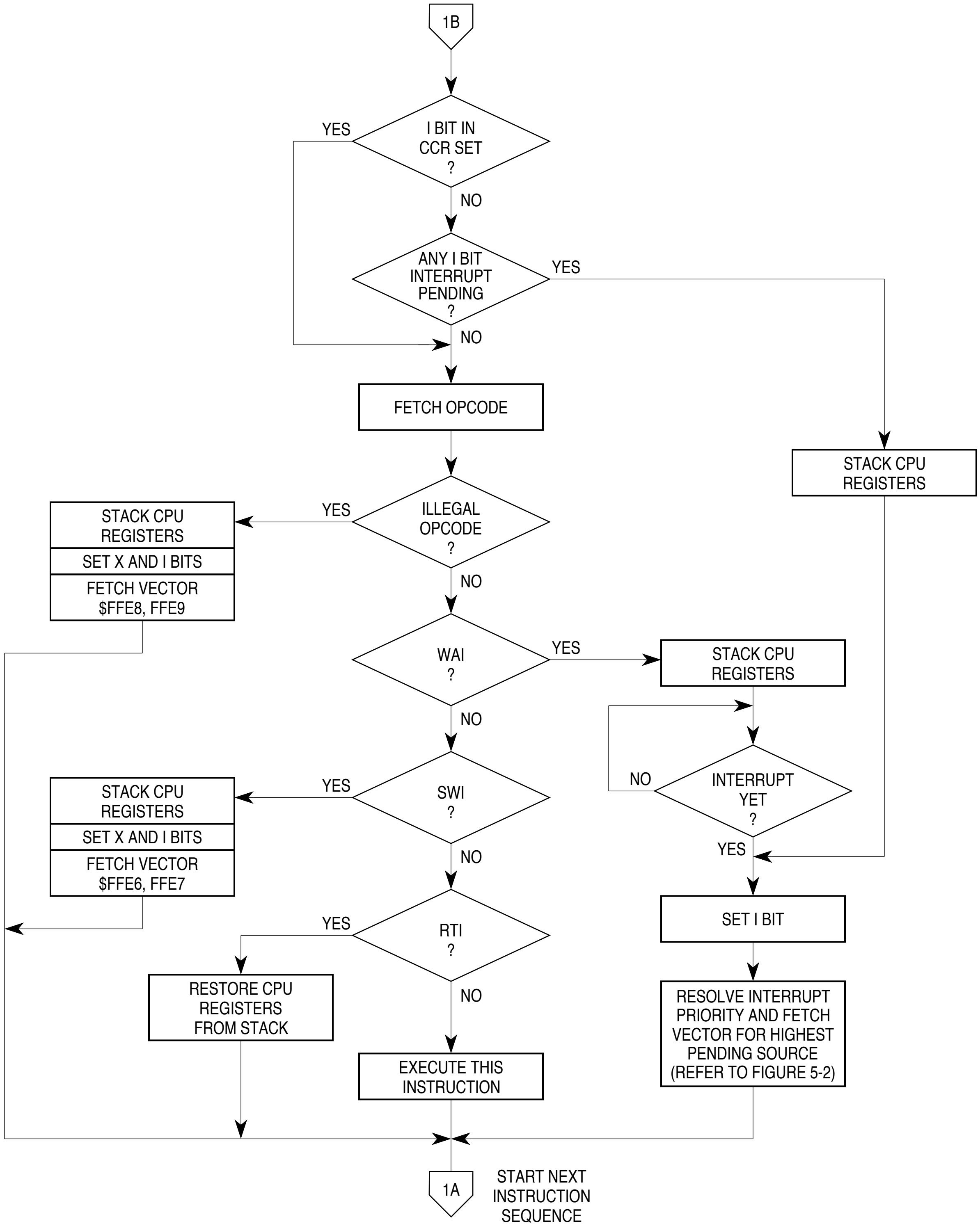


Figure 5-2 Processing Flow Out of Reset (2 of 2)

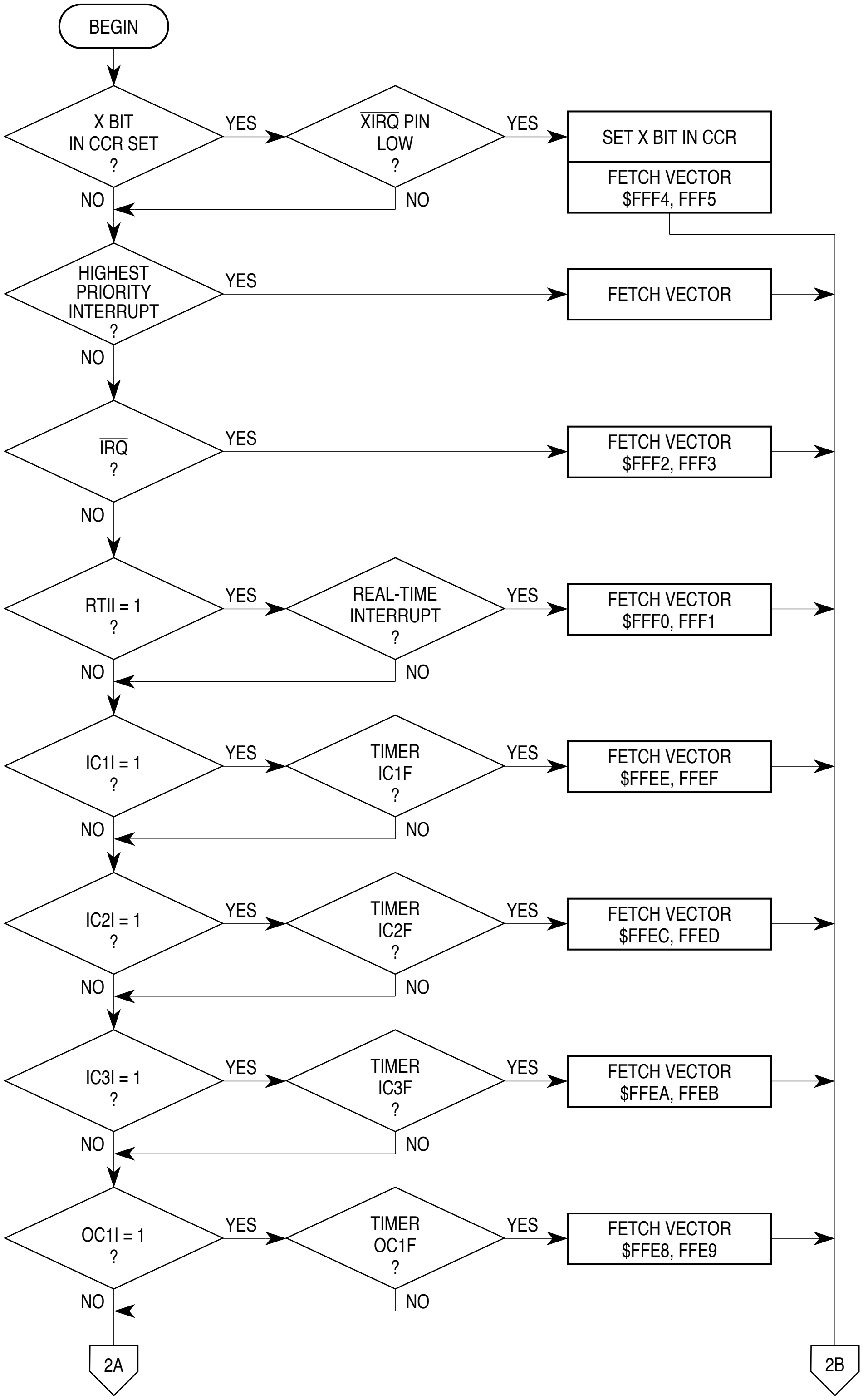


Figure 5-3 Interrupt Priority Resolution (1 of 2)

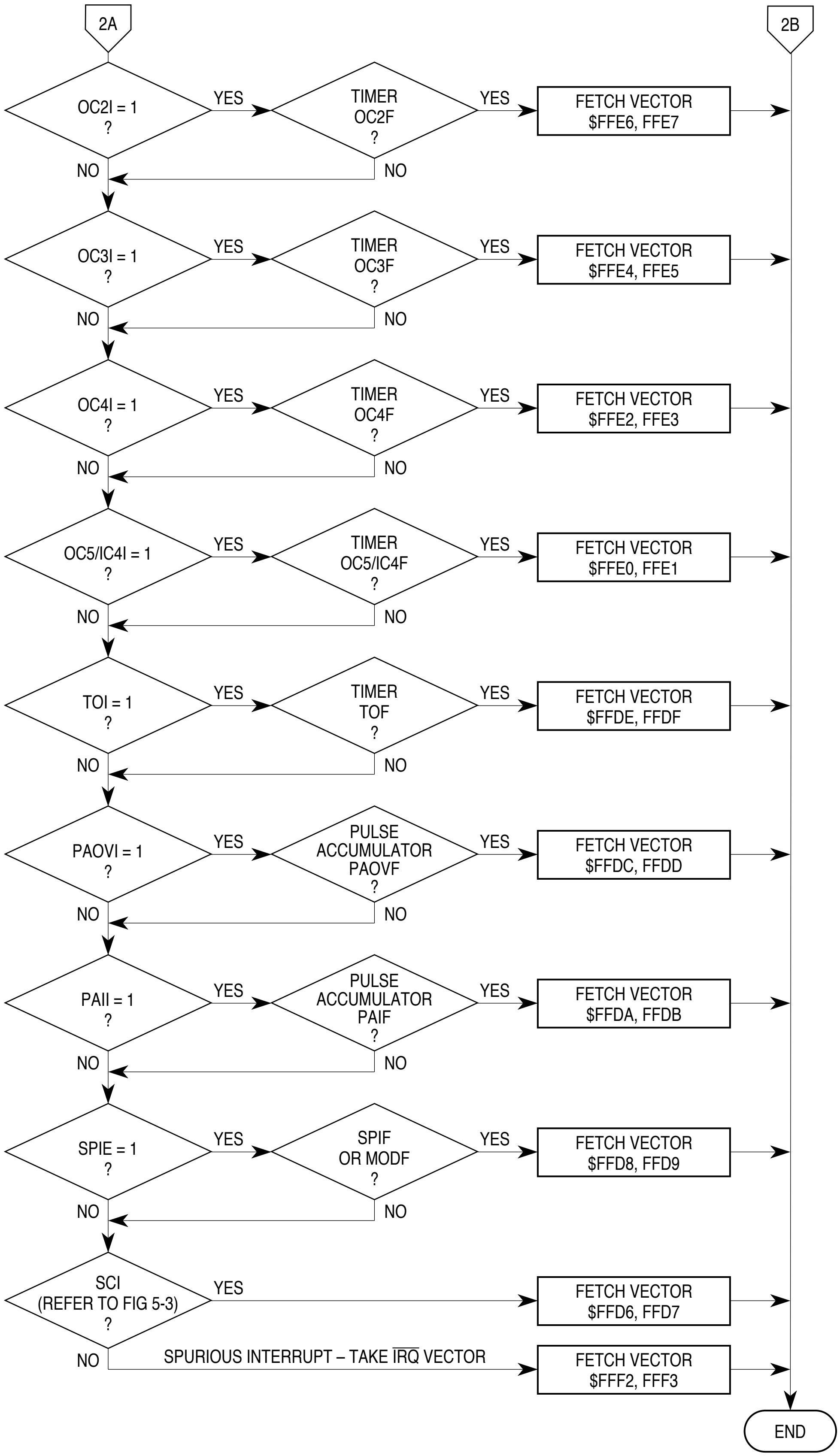


Figure 5-4 Interrupt Priority Resolution (2 of 2)

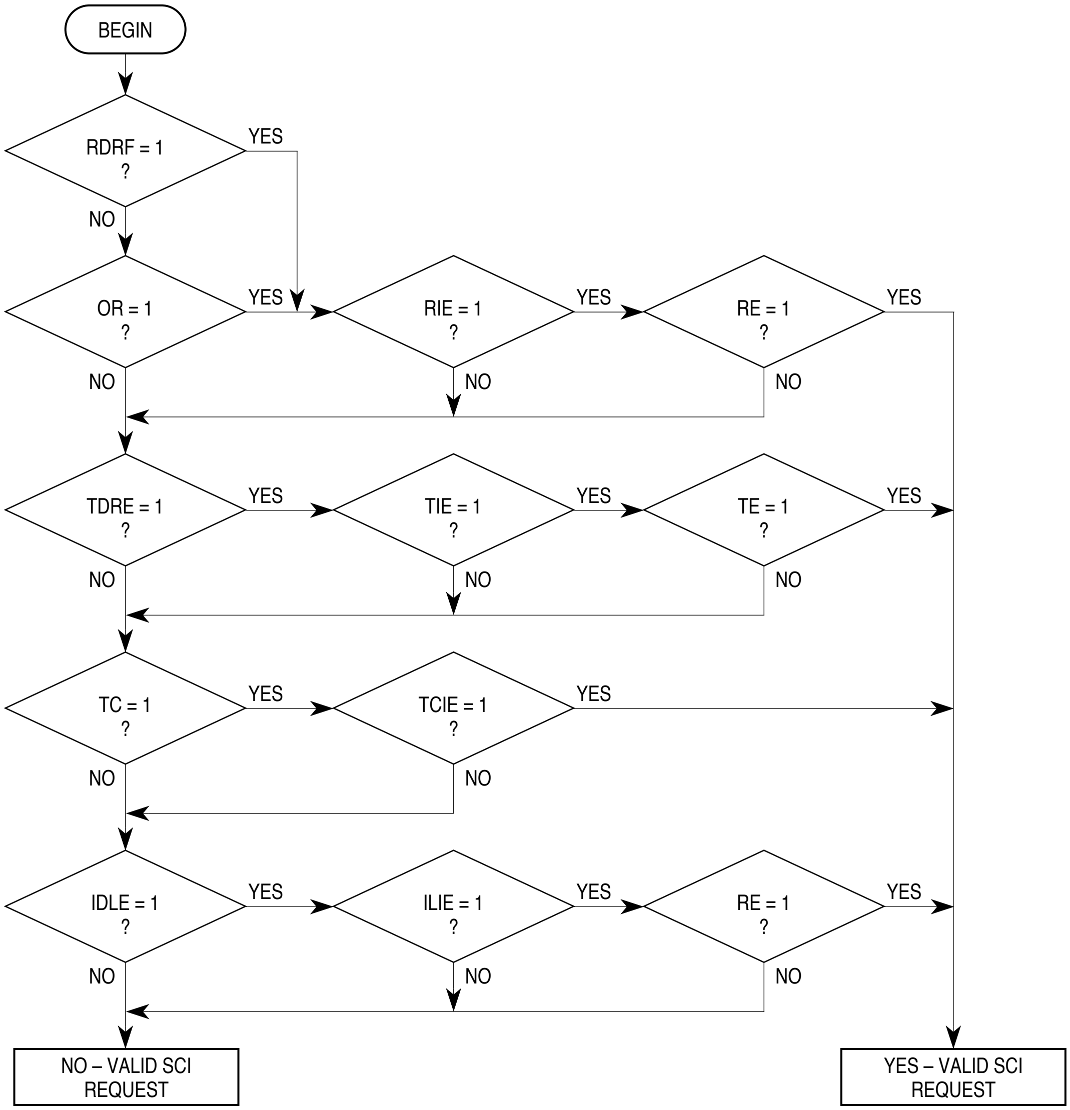


Figure 5-5 Interrupt Source Resolution Within SCI

5.5 Low Power Operation

Both STOP and WAIT suspend CPU operation until a reset or interrupt occurs. The WAIT condition suspends processing and reduces power consumption to an intermediate level. The STOP condition turns off all on-chip clocks and reduces power consumption to an absolute minimum while retaining the contents of all 1024 bytes of RAM.

5.5.1 WAIT

The WAI opcode places the MCU in the WAIT condition, during which the CPU registers are stacked and CPU processing is suspended until a qualified interrupt is detected. The interrupt can be an external $\overline{IRQ}$, an $\overline{XIRQ}$, or any of the internally generated interrupts, such as the timer or serial interrupts. The on-chip crystal oscillator remains active throughout the WAIT standby period.

The reduction of power in the WAIT condition depends on how many internal clock signals driving on-chip peripheral functions can be shut down. The CPU is always shut down during WAIT. While in the wait state, the address/data bus repeatedly runs read cycles to the address where the CCR contents were stacked. Ensuring that the stack contents are placed in internal RAM will further reduce power consumption. The MCU leaves the wait state when it senses any interrupt that has not been masked.

The free-running timer system is shut down only if the I bit is set to one and the COP system is disabled by NOCOP being set to one. Several other systems can also be in a reduced power consumption state depending on the state of software-controlled configuration control bits. Power consumption by the analog-to-digital (A/D) converter is not affected significantly by the WAIT condition. However, the A/D converter current can be eliminated by writing the ADPU bit to zero. The SPI system is enabled or disabled by the SPE control bit. The SCI transmitter is enabled or disabled by the TE bit, and the SCI receiver is enabled or disabled by the RE bit. Therefore the power consumption in WAIT is dependent on the particular application.

5.5.2 STOP

Executing the STOP instruction while the S bit in the CCR is equal to zero places the MCU in the STOP condition. If the S bit is not zero, the STOP opcode is treated as a no-op (NOP). The STOP condition offers minimum power consumption because all clocks, including the crystal oscillator, are stopped while in this mode. To exit STOP and resume normal processing, a logic low level must be applied to one of the external interrupts ($\overline{IRQ}$ or $\overline{XIRQ}$) or to the $\overline{RESET}$ pin. A pending edge-triggered $\overline{IRQ}$ can also bring the CPU out of STOP.

Because all clocks are stopped in this mode, all internal peripheral functions also stop. The data in the internal RAM is retained as long as V_{DD} power is maintained. The CPU state and I/O pin levels are static and are unchanged by STOP. Therefore, when an interrupt restarts the system, the MCU resumes processing as if there were no interruption. If reset is used to restart the system a normal reset sequence results where all I/O pins and functions are also restored to their initial states.

To use the $\overline{IRQ}$ pin as a means of recovering from STOP, the I bit in the CCR must be clear ($\overline{IRQ}$ not masked). The $\overline{XIRQ}$ pin can be used to wake up the MCU from STOP regardless of the state of the X bit in the CCR, although the recovery sequence depends on the state of the X bit. If X is set to zero ($\overline{XIRQ}$ not masked), the MCU starts up, beginning with the stacking sequence leading to normal service of the $\overline{XIRQ}$ request. If X is set to one ($\overline{XIRQ}$ masked or inhibited), then processing continues with the instruction that immediately follows the STOP instruction, and no $\overline{XIRQ}$ interrupt service is requested or pending.

Because the oscillator is stopped in STOP mode, a restart delay may be imposed to allow oscillator stabilization upon leaving STOP. If the internal oscillator is being used, this delay is required; however, if a stable external oscillator is being used, the DLY control bit can be used to bypass this start-up delay. The DLY control bit is set by reset and can be optionally cleared during initialization. If the DLY equal to zero option is used to avoid start-up delay on recovery from STOP, then reset should not be used as the means of recovering from STOP, as this causes DLY to be set again by reset, imposing the restart delay. This same delay also applies to power-on reset, regardless of the state of the DLY control bit, but does not apply to a reset while the clocks are running.

SECTION 6 PARALLEL INPUT/OUTPUT

The MC68HC11F1 MCU has up to 54 input/output lines, depending on the operating mode. The data bus of this microcontroller is nonmultiplexed. I/O lines are organized into seven parallel ports. Ports with bidirectional pins have an associated data direction control register. This register (DDRx) contains a data direction control bit for each bidirectional port line. The following table is a summary of the configuration and features of each port.

Table 6-1 I/O Port Configuration

Port	Input Pins	Output Pins	Bidirectional Pins	Shared Functions
Port A	—	—	8	Timer
Port B	—	8	—	High-Order Address
Port C	—	—	8	Data Bus
Port D	—	—	6	SCI and SPI
Port E	8	—	—	A/D Converter
Port F	—	8	—	Low-Order Address
Port G	—	—	8	Chip Select Outputs

Port pin function is mode dependent. Do not confuse pin function with the electrical state of the pin at reset. Port pins are either driven to a specified logic level or are configured as high impedance inputs. I/O pins configured as high-impedance inputs have port data that is indeterminate. The contents of the corresponding latches are dependent upon the electrical state of the pins during reset. In port descriptions, an “I” indicates this condition. Port pins that are driven to a known logic level during reset are shown with a value of either one or zero. Some control bits are unaffected by reset. Reset states for these bits are indicated with a “U”.

6.1 Port A

Port A has eight bidirectional I/O pins and shares functions with the timer system.

PORTA — Port A Data **$1000**

	Bit 7	6	5	4	3	2	1	Bit 0
	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
RESET:	I	I	I	I	I	I	I	I
Alt. Pin Func.:	PAI	OC2	OC3	OC4	IC4/OC5	IC1	IC2	IC3
And/or:	OC1	OC1	OC1	OC1	OC1	—	—	—

DDRA — Data Direction Register for Port A **$1001**

	Bit 7	6	5	4	3	2	1	Bit 0
	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0
RESET:	0	0	0	0	0	0	0	0

DDA[7:0] — Data Direction for Port A
0 = Input
1 = Output

NOTE

To enable PA3 as fourth input capture, set the I4/O5 bit in the PACTL register. Otherwise, PA3 is configured as a fifth output compare out of reset, with bit I4/O5 being cleared. If the DDA3 bit 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 TI4/O5 has no effect when the TI4/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. Note that even when PA7 is configured as an output, the pin still drives the pulse accumulator input.

6.2 Port B

Reset state is mode dependent. In single-chip or bootstrap modes, port B pins are general-purpose outputs. In expanded and test modes, port B pins are high-order address outputs and PORTB is not in the memory map.

PORTB — Port B Data **$1004**

	Bit 7	6	5	4	3	2	1	Bit 0
	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
S. Chip or Boot:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
RESET:	0	0	0	0	0	0	0	0
Expan. or Test:	ADDR15	ADDR14	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8

6.3 Port C

Reset state is mode dependent. In single-chip and bootstrap modes, port C pins are high-impedance inputs. It is customary to have an external pull-up resistor on lines that are driven by open-drain devices. In expanded or test modes, port C pins are data bus inputs/outputs and PORTC is not in the memory map. The R/$\overline{W}$ signal is used to control the direction of data transfers.

The CWOM control bit in the OPT2 register disables port C's P-channel output drivers. Because the N-channel driver is not affected by CWOM, setting CWOM causes port C to become an open-drain-type output port suitable for wired-OR operation. In wired-OR mode, (PORTC bits are at logic level zero), pins are actively driven low by the N-channel driver. When a port C bit is at logic level one, the associated pin is in a high-impedance state, as neither the N-channel nor the P-channel devices are active. It is customary to have an external pull-up resistor on lines that are driven by open-drain devices. Port C can only be configured for wired-OR operation when the MCU is in single-chip or bootstrap modes.

PORTC — Port C Data **$1006**

	Bit 7	6	5	4	3	2	1	Bit 0
	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
S. Chip or Boot:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
RESET:	I	I	I	I	I	I	I	I
Expan. or Test:	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

DDRC — Data Direction Register for Port C **$1007**

	Bit 7	6	5	4	3	2	1	Bit 0
	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
RESET:	0	0	0	0	0	0	0	0

DDC[7:0] — Data Direction for Port C
0 = Input
1 = Output

6.4 Port D

In all modes, port D bits [5:0] can be used either for general-purpose I/O, or with the SCI and SPI subsystems. During reset, port D pins are configured as high impedance inputs (DDRD bits cleared).

The DWOM control bit in the SPCR register disables port D's P-channel output drivers. Because the N-channel driver is not affected by DWOM, setting DWOM causes port D to become an open-drain-type output port suitable for wired-OR operation. In wired-OR mode, (PORTD bits are at logic level zero), pins are actively driven low by the N-channel driver. When a port D bit is at logic level one, the associated pin is in a high-impedance state, as neither the N-channel nor the P-channel devices are active. It is customary to have an external pull-up resistor on lines that are driven by open-drain devices. Port D can be configured for wired-OR operation in any operating mode.

PORTD — Port D Data **$1008**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	PD5	PD4	PD3	PD2	PD1	PD0
RESET:	0	0	I	I	I	I	I	I
Alt. Pin Func.:	—	—	$\overline{\text{SS}}$	SCK	MOSI	MISO	TxD	RxD

DDRD — Data Direction Register for Port D **$1009**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
RESET:	0	0	0	0	0	0	0	0

Bits [7:6] — Not implemented
Always read zero

DDD[5:0] — Data Direction for Port D
0 = Input
1 = Output

NOTE

When the SPI system is in slave mode, DDD5 has no meaning nor 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.

6.5 Port E

Port E has eight general-purpose input pins and shares functions with the A/D converter system. When some port E pins are being used for general-purpose input and others are being used as A/D inputs, PORTE should not be read during the sample portion of an A/D conversion.

PORTE — Port E Data **$100A**

	Bit 7	6	5	4	3	2	1	Bit 0
	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
RESET:	I	I	I	I	I	I	I	I
Alt. Pin Func.:	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0

6.6 Port F

Reset state is mode dependent. In single-chip or bootstrap modes, port F pins are general-purpose outputs. In expanded and test modes, port F pins are low order address outputs and PORTF is not in the memory map.

PORTF — Port F Data **$1005**

	Bit 7	6	5	4	3	2	1	Bit 0
	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
S. Chip or Boot:	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
RESET:	0	0	0	0	0	0	0	0
Expan. or Test:	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0

6.7 Port G

Port G pins reset to high-impedance inputs except in expanded modes where reset causes PG7 to become the $\overline{\text{CSPROG}}$ output. Alternate functions for port G bits [7:4] are chip select outputs. All port G bits are bidirectional and have corresponding data direction bits.

The GWOM control bit in the OPT2 register disables port G's P-channel output drivers. Because the N-channel driver is not affected by GWOM, setting GWOM causes port G to become an open-drain-type output port suitable for wired-OR operation. In wired-OR mode, (PORTG bits are at logic level zero), pins are actively driven low by the N-channel driver. When a port G bit is at logic level one, the associated pin is in a high-impedance state, as neither the N-channel nor the P-channel devices are active. It is customary to have an external pull-up resistor on lines that are driven by open-drain devices. Port G can be configured for wired-OR operation in any operating mode.

PORTG — Port G Data **$1002**

	Bit 7	6	5	4	3	2	1	Bit 0
	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
RESET:	I	I	I	I	I	I	I	I
Alt. Pin Func.:	$\overline{\text{CSPROG}}$	CSGEN	CSIO1	CSIO2	—	—	—	—

DDRG — Data Direction Register for Port G **$1003**

	Bit 7	6	5	4	3	2	1	Bit 0
	DDG7	DDG6	DDG5	DDG4	DDG3	DDG2	DDG1	DDG0
RESET:	0	0	0	0	0	0	0	0

DDG[7:0] — Data Direction for Port G

0 = Input

1 = Output

6.8 System Configuration Options 2

The system configuration options 2 register controls several configuration parameters. Bit 6, CWOM, is the only bit in this register that directly affects parallel I/O.

OPT2 — System Configuration Options 2 **$1038**

	Bit 7	6	5	4	3	2	1	Bit 0
	GWOM	CWOM	CLK4X	—	—	—	—	—
RESET:	0	0	1	0	0	0	0	0

GWOM — Port G Wired-OR Mode
0 = Port G operates normally
1 = Port G outputs are open drain

CWOM — Port C Wired-OR Mode
0 = Port C operates normally
1 = Port C outputs are open drain

CLK4X — 4XOUT Clock Enable
Refer to **SECTION 2 PIN DESCRIPTIONS**.

Bits [4:0] — Not implemented
Always read zero

SECTION 7 SERIAL COMMUNICATIONS INTERFACE

The serial communications interface (SCI) is a universal asynchronous receiver transmitter (UART), one of two independent serial I/O subsystems in the MC68HC11F1 MCU. It has a standard nonreturn to zero (NRZ) format (one start bit, eight or nine data bits, and one stop bit). Several baud rates are available. The SCI transmitter and receiver are independent, but use the same data format and bit rate.

7.1 Data Format

The serial data format requires the following conditions:

1. An idle-line in the high state before transmission or reception of a message.
2. A start bit, logic zero, transmitted or received, that indicates the start of each character.
3. Data that is transmitted and received least significant bit (LSB) first.
4. A stop bit, logic one, used to indicate the end of a frame. (A frame consists of a start bit, a character of eight or nine data bits, and a stop bit.)
5. A break (defined as the transmission or reception of a logic zero for some multiple number of frames).

Selection of the word length is controlled by the M bit of SCI control register SCCR1.

7.2 Transmit Operation

The SCI transmitter includes a parallel transmit data register (SCDR) and a serial shift register. The contents of the serial shift register can only be written through the SCDR. This double buffered operation allows a character to be shifted out serially while another character is waiting in the SCDR to be transferred into the serial shift register. The output of the serial shift register is applied to TxD as long as transmission is in progress or the transmit enable (TE) bit of serial communication control register 2 (SCCR2) is set. The block diagram, **Figure 7-1**, shows the transmit serial shift register and the buffer logic at the top of the figure.

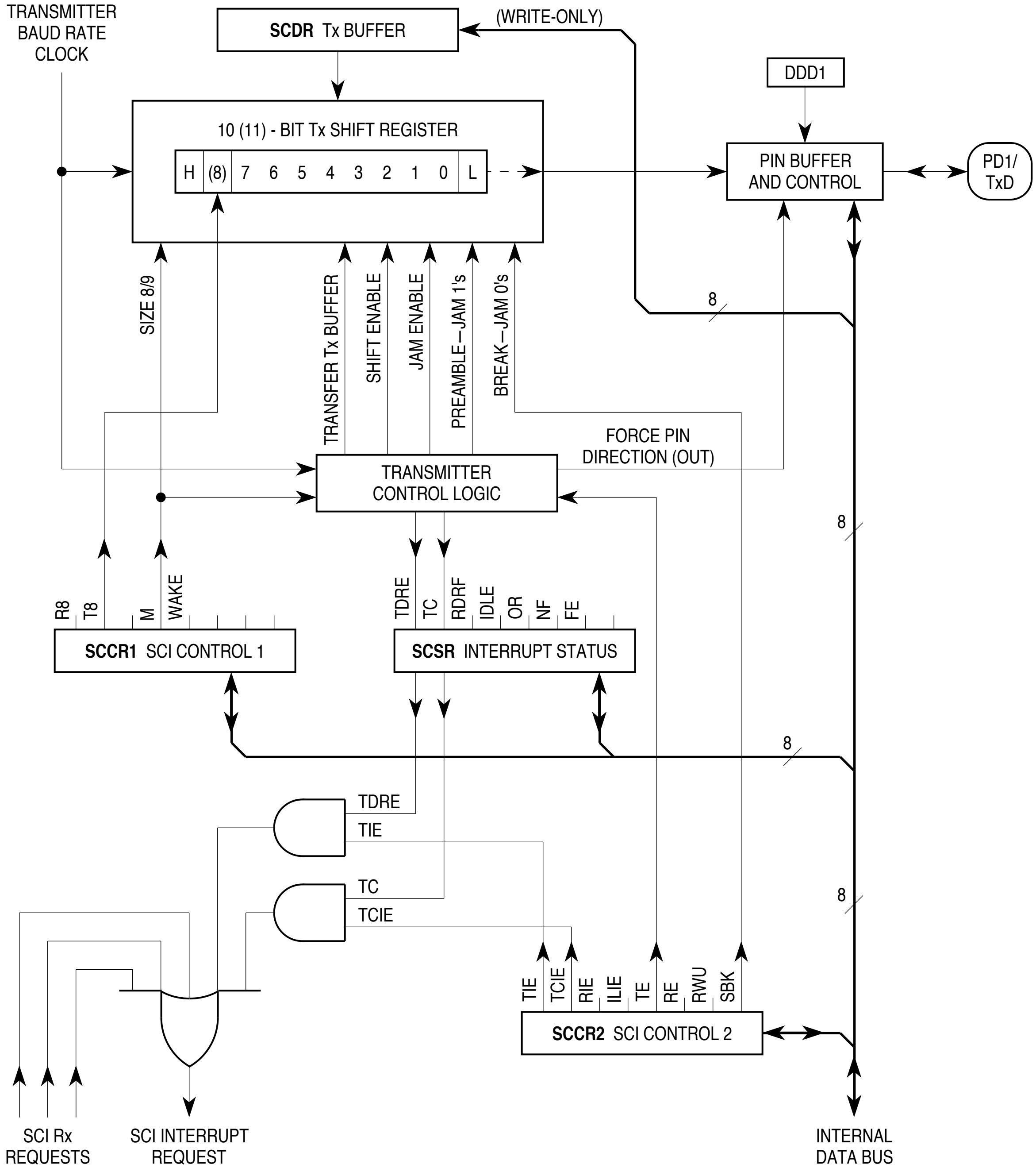


Figure 7-1 SCI Transmitter Block Diagram

7.3 Receive Operation

During receive operations, the transmit sequence is reversed. The serial shift register receives data and transfers it to a parallel receive data register (SCDR) as a complete word. This double buffered operation allows a character to be shifted in serially while another character is already in the SCDR. An advanced data recovery scheme distinguishes valid data from noise in the serial data stream. The data input is selectively sampled to detect receive data, and a majority voting circuit determines the value and integrity of each bit.

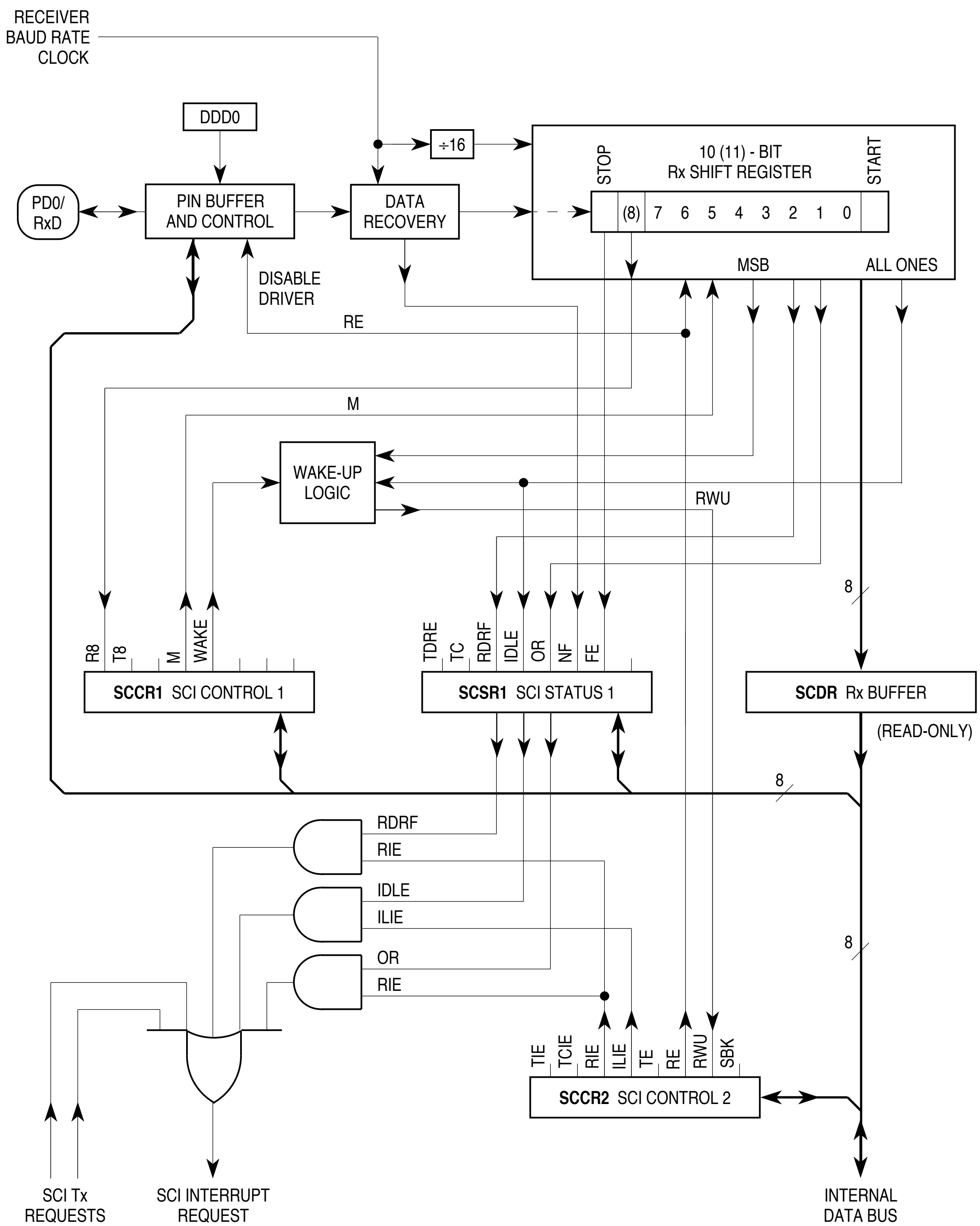


Figure 7-2 SCI Receiver Block Diagram

7.4 Wakeup Feature

The wakeup feature reduces SCI service overhead in multiple receiver systems. Software for each receiver evaluates the first character of each message. The receiver is placed in wakeup mode by writing a one to the RWU bit in the SCCR2 register. While RWU is one, all of the receiver-related status flags (RDRF, IDLE, OR, NF, and FE) are inhibited (cannot become set). Although RWU can be cleared by a software write to SCCR2, to do so would be unusual. Normally RWU is set by software and is cleared automatically with hardware. Whenever a new message begins, logic alerts the sleeping receivers to wake up and evaluate the initial character of the new message.

Two methods of wakeup are available: idle-line wakeup and address-mark wakeup. During idle-line wakeup, a sleeping receiver awakens as soon as the RxD line becomes idle. In the address-mark wakeup, logic one in the most significant bit (MSB) of a character wakes up all sleeping receivers.

7.4.1 Idle-Line Wakeup

To use the receiver wakeup method, establish a software addressing scheme to allow the transmitting devices to direct a message to individual receivers or to groups of receivers. This addressing scheme can take any form as long as all transmitting and receiving devices are programmed to understand the same scheme. Because the addressing information is usually the first frame(s) in a message, receivers that are not part of the current task do not become burdened with the entire set of addressing frames. All receivers are awake (RWU = 0) when each message begins. As soon as a receiver determines that the message is not intended for it, software sets the RWU bit (RWU = 1), which inhibits further flag setting until the RxD line goes idle at the end of the message. As soon as an idle line is detected by receiver logic, hardware automatically clears the RWU bit so that the first frame of the next message can be received. This type of receiver wakeup requires a minimum of one idle-line frame time between messages, and no idle time between frames in a message.

7.4.2 Address-Mark Wakeup

The serial characters in this type of wakeup consist of seven (eight if M = 1) information bits and an MSB, which indicates an address character (when set to one, or mark). The first character of each message is an addressing character (MSB = 1). All receivers in the system evaluate this character to determine if the remainder of the message is directed toward this particular receiver. As soon as a receiver determines that a message is not intended for it, the receiver activates the RWU function by using a software write to set the RWU bit. Because setting RWU inhibits receiver-related flags, there is no further software overhead for the rest of this message.

When the next message begins, its first character has its MSB set, which automatically clears the RWU bit and enables normal character reception. The first character whose MSB is set is also the first character to be received after wakeup because RWU gets cleared before the stop bit for that frame is serially received. This type of wakeup allows messages to include gaps of idle time, unlike the idle-line method, but there is a loss of efficiency because of the extra bit time for each character (address bit) required for all characters.

7.5 SCI Error Detection

Three error conditions, SCDR overrun, received bit noise, and framing can occur during generation of SCI system interrupts. Three bits (OR, NF, and FE) in the serial communications status register (SCSR) indicate if one of these error conditions exists.

The overrun error (OR) bit is set when the next byte is ready to be transferred from the receive shift register to the SCDR and the SCDR is already full (RDRF bit is set). When an overrun error occurs, the data that caused the overrun is lost and the data that was already in SCDR is not disturbed. The OR is cleared when the SCSR is read (with OR set), followed by a read of the SCDR.

The noise flag (NF) bit is set if there is noise on any of the received bits, including the start and stop bits. The NF bit is not set until the RDRF flag is set. The NF bit is cleared when the SCSR is read (with FE equal to one) followed by a read of the SCDR.

When no stop bit is detected in the received data character, the framing error (FE) bit is set. FE is set at the same time as the RDRF. If the byte received causes both framing and overrun errors, the processor only recognizes the overrun error. The framing error flag inhibits further transfer of data into the SCDR until it is cleared. The FE bit is cleared when the SCSR is read (with FE equal to one) followed by a read of the SCDR.

7.6 SCI Registers

There are five addressable registers associated with the SCI. SCCR1, SCCR2, and BAUD are control registers. SCDR is the SCI data register and SCSR is the SCI status register. Refer to the BAUD register description as well as the block diagram for the baud rate generator.

7.6.1 Serial Communications Data Register

SCDR is a parallel register that performs two functions. It is the receive data register when it is read, and the transmit data register when it is written. Reads access the receive data buffer and writes access the transmit data buffer. Receive and transmit are double buffered.

SCDR — SCI Data Register **$102F**

	Bit 7	6	5	4	3	2	1	Bit 0
	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0
RESET:	I	I	I	I	I	I	I	I

7.6.2 Serial Communications Control Register 1

The SCCR1 register provides the control bits that determine word length and select the method used for the wakeup feature.

SCCR1 — SCI Control Register 1 **$102C**

	Bit 7	6	5	4	3	2	1	Bit 0
	R8	T8	—	M	WAKE	—	—	—
RESET:	I	I	0	0	0	0	0	0

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.

M — Mode (Select Character Format)

0 = Start bit, 8 data bits, 1 stop bit
1 = Start bit, 9 data bits, 1 stop bit

WAKE — Wakeup by Address Mark/Idle

0 = Wakeup by IDLE line recognition
1 = Wakeup by address mark (most significant data bit set)

7.6.3 Serial Communications Control Register 2

The SCCR2 register provides the control bits that enable or disable individual SCI functions.

SCCR2 — SCI Control Register 2 **$102D**

	Bit 7	6	5	4	3	2	1	Bit 0
	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET:	0	0	0	0	0	0	0	0

TIE — Transmit Interrupt Enable

0 = TDRE interrupts disabled
1 = SCI interrupt requested when TDRE status flag is set

TCIE — Transmit Complete Interrupt Enable

0 = TC interrupts disabled
1 = SCI interrupt requested when TC status flag is set

RIE — Receiver Interrupt Enable

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

0 = IDLE interrupts disabled
1 = SCI interrupt requested when IDLE status flag is set

TE — Transmitter Enable

When TE goes from zero to one, one unit of idle character time (logic one) is queued as a preamble.

0 = Transmitter disabled
1 = Transmitter enabled

RE — Receiver Enable

0 = Receiver disabled
1 = Receiver enabled

RWU — Receiver Wakeup Control
0 = Normal SCI receiver
1 = Wakeup enabled and receiver interrupts inhibited

SBK — Send Break
At least one character time of break is queued and sent each time SBK is written to one. As long as the SBK bit is set, break characters are queued and sent. More than one break may be sent if the transmitter is idle at the time the SBK bit is toggled on and off, as the baud rate clock edge could occur between writing the one and writing the zero to SBK.
0 = Break generator off
1 = Break codes generated

7.6.4 Serial Communication Status Register

The SCSR provides inputs to the interrupt logic circuits for generation of the SCI system interrupt.

SCSR — SCI Status Register **$102E**

	Bit 7	6	5	4	3	2	1	Bit 0
	TDRE	TC	RDRF	IDLE	OR	NF	FE	—
RESET:	1	1	0	0	0	0	0	0

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 is active
1 = RxD line is idle

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

FE is set when a zero 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 — Not implemented

Always reads zero

7.6.5 Baud Rate Register

Use this register to select different baud rates for the SCI system. The SCP[1:0] bits select the prescaler rate for the SCR[2:0] bits. Together, these five bits provide multiple baud rate combinations for a given crystal frequency. Normally, this register is written once during initialization. The prescaler is set to its fastest rate by default out of reset, and can be changed at any time. Refer to **Table 7-1** and **Table 7-2** for normal baud rate selections.

BAUD — Baud Rate **$102B**

	Bit 7	6	5	4	3	2	1	Bit 0
	TCLR	—	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0
RESET:	0	0	0	0	0	U	U	U

TCLR — Clear Baud Rate Counters (Test)

SCP[1:0] — SCI Baud Rate Prescaler Selects

Refer to the SCI baud rate generator block diagram.

Table 7-1 Baud Rate Prescaler Selection

Prescaler		Divide Internal	Crystal Frequency (MHz)					
SCP1	SCP0	Clock By	4.0	4.9152	8.0	12.0	16.0	20.0
0	0	1	62500	76800	125000	187500	25000	312500
0	1	3	20833	25600	41667	62500	83332	104165
1	0	4	15625	19200	31250	46875	62500	78125
1	1	13	4800	5907	9600	14423	19200	24000

RCKB — SCI Baud Rate Clock Check (Test)

SCR[2:0] — SCI Baud Rate Selects

Selects receiver and transmitter bit rate based on output from baud rate prescaler stage. Refer to the SCI baud rate generator block diagram.

Table 7-2 Baud Rate Selection

SCR[2:0]	Divide Prescaler By	Highest Baud Rate (Prescaler Output from Previous Table)			
		4800	**19200**	**76800**	**312500**
0 0 0	1	4800	19200	76800	312500
0 0 1	2	2400	9600	38400	156250
0 1 0	4	1200	4800	19200	78125
0 1 1	8	600	2400	9600	39063
1 0 0	16	300	1200	4800	19531
1 0 1	32	150	600	2400	9766
1 1 0	64	75	300	1200	4883
1 1 1	128	—	150	600	2441

The prescaler bits, SCP[2:0], determine the highest baud rate, and the SCR[2:0] bits select an additional binary submultiple (≥1, ≥2, ≥4, through ≥128) of this highest baud rate. The result of these two dividers in series is the 16X receiver baud rate clock. The SCR[2:0] bits are not affected by reset and can be changed at any time, although they should not be changed when any SCI transfer is in progress.

Figure 7-3 and **Figure 7-4** illustrate the SCI baud rate timing chain. The prescaler select bits determine the highest baud rate. The rate select bits determine additional divide by two stages to arrive at the receiver timing (RT) clock rate. The baud rate clock is the result of dividing the RT clock by 16.

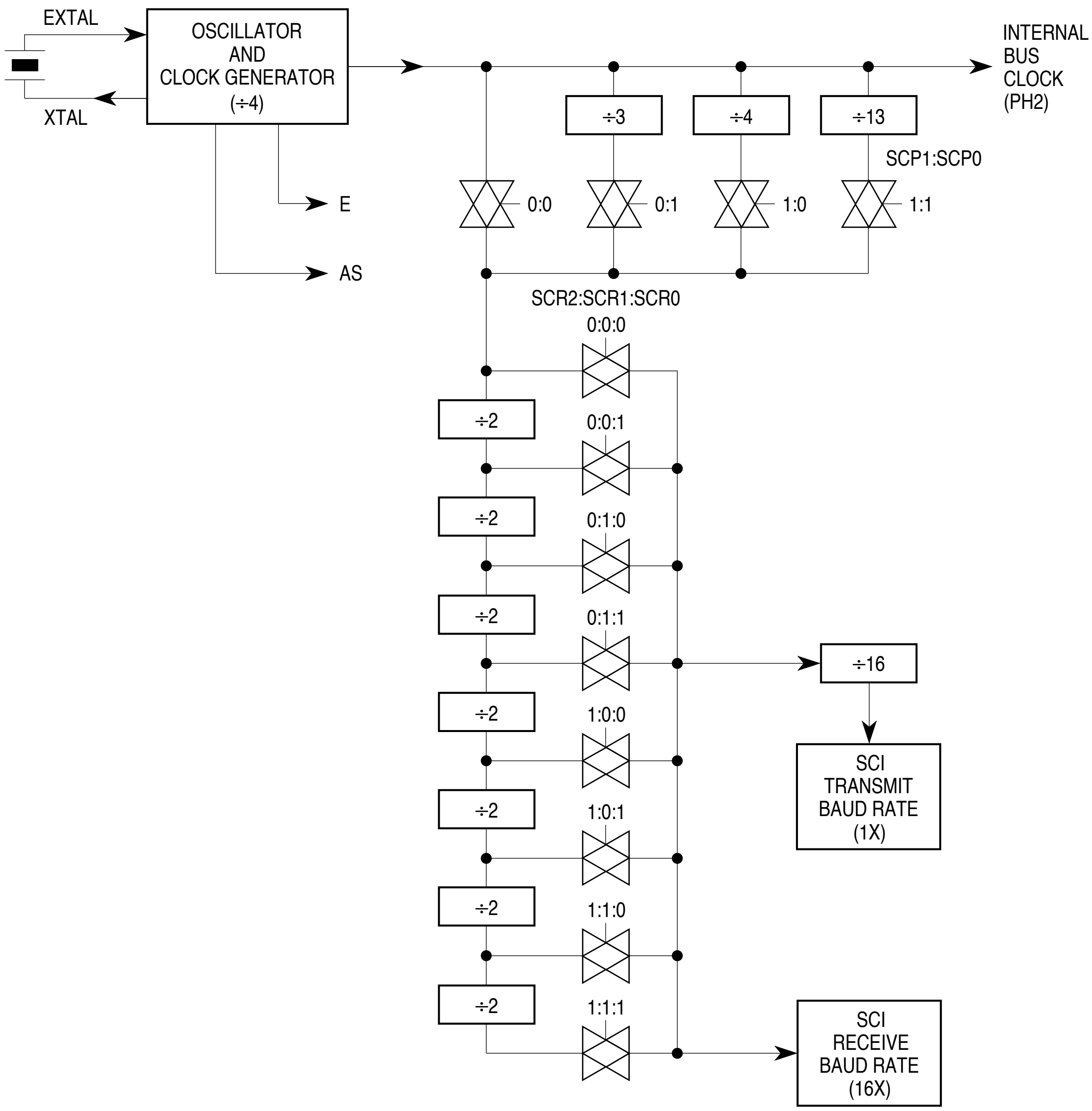


Figure 7-3 SCI Baud Rate Generator Block Diagram

7.7 Status Flags and Interrupts

The SCI transmitter has two status flags. These status flags can be read by software (polled) to tell when the corresponding condition exists. Alternatively, a local interrupt enable bit can be set to enable each of these status conditions to generate interrupt requests when the corresponding condition is present. Status flags are automatically set by hardware logic conditions, but must be cleared by software, which provides an interlock mechanism that enables logic to know when software has noticed the status indication. The software clearing sequence for these flags is automatic — functions that are normally performed in response to the status flags also satisfy the conditions of the clearing sequence.

TDRE and TC flags are normally set when the transmitter is first enabled (TE set to one). The TDRE flag indicates there is room in the transmit queue to store another data character in the TDR. The TIE bit is the local interrupt mask for TDRE. When TIE is zero, TDRE must be polled. When TIE and TDRE are one, an interrupt is requested.

The TC flag indicates the transmitter has completed the queue. The TCIE bit is the local interrupt mask for TC. When TCIE is zero, TC must be polled; when TCIE is one and TC is one, an interrupt is requested.

Writing a zero to TE requests that the transmitter stop when it can. The transmitter completes any transmission in progress before actually shutting down. Only an MCU reset can cause the transmitter to stop and shut down immediately. If TE is written to zero when the transmitter is already idle, the pin reverts to its general-purpose I/O function (synchronized to the bit-rate clock). If anything is being transmitted when TE is written to zero, that character is completed before the pin reverts to general-purpose I/O, but any other characters waiting in the transmit queue are lost. The TC and TDRE flags are set at the completion of this last character, even though TE has been disabled.

7.7.1 Receiver Flags

The SCI receiver has five status flags, three of which can generate interrupt requests. The status flags are set by the SCI logic in response to specific conditions in the receiver. These flags can be read (polled) at any time by software. Refer to Figure 7–4, which shows SCI interrupt arbitration.

When an overrun takes place, the new character is lost, and the character that was in its way in the parallel RDR is undisturbed. RDRF is set when a character has been received and transferred into the parallel RDR. The OR flag is set instead of RDRF if overrun occurs. A new character is ready to be transferred into RDR before a previous character is read from RDR.

The NF and FE flags provide additional information about the character in the RDR, but do not generate interrupt requests.

The last receiver status flag and interrupt source come from the IDLE flag. The RxD line is idle if it has constantly been at logic one for a full character time. The IDLE flag is set only after the RxD line has been busy and becomes idle, which prevents repeated interrupts for the whole time RxD remains idle.

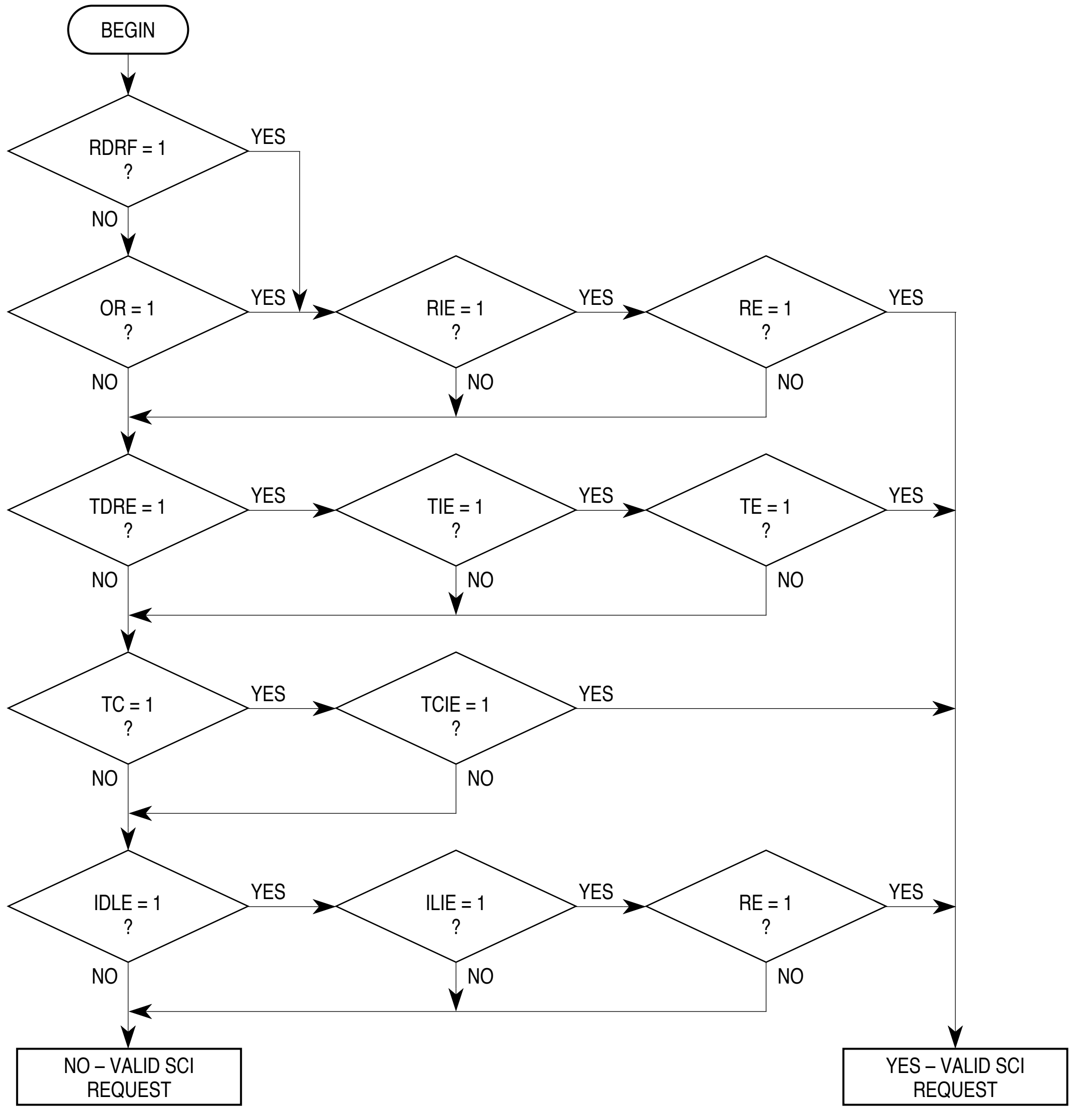


Figure 7-4 Interrupt Source Resolution Within SCI

SECTION 8 SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI), an independent serial communications subsystem, allows the MCU to communicate synchronously with peripheral devices, such as transistor-transistor logic (TTL) shift registers, liquid crystal display (LCD) drivers, analog-to-digital converter subsystems, and other microprocessors. The SPI is also capable of inter-processor communication in a multiple master system. The SPI system can be configured as either a master or a slave device. When configured as a master, data transfer rates can be as high as one-half the E-clock rate (2.5 Mbits per second for a 5-MHz bus frequency). When configured as a slave, data transfers can be as fast as the E-clock rate (5 Mbits per second for a 5-MHz bus frequency).

8.1 Functional Description

The central element in the SPI system is the block containing the shift register and the read data buffer. The system is single buffered in the transmit direction and double buffered in the receive direction. This means that new data for transmission cannot be written to the shifter until the previous transfer is complete; however, received data is transferred into a parallel read data buffer so the shifter is free to accept a second serial character. As long as the first character is read out of the read data buffer before the next serial character is ready to be transferred, no overrun condition occurs. A single MCU register address is used for reading data from the read data buffer and for writing data to the shifter.

The SPI status block represents the SPI status flags (transfer complete, write collision, and mode fault) located in the SPI status register (SPSR). The SPI control block represents those functions that control the SPI system through the serial peripheral control register (SPCR).

Refer to **Figure 8-1**, which shows the SPI block diagram.

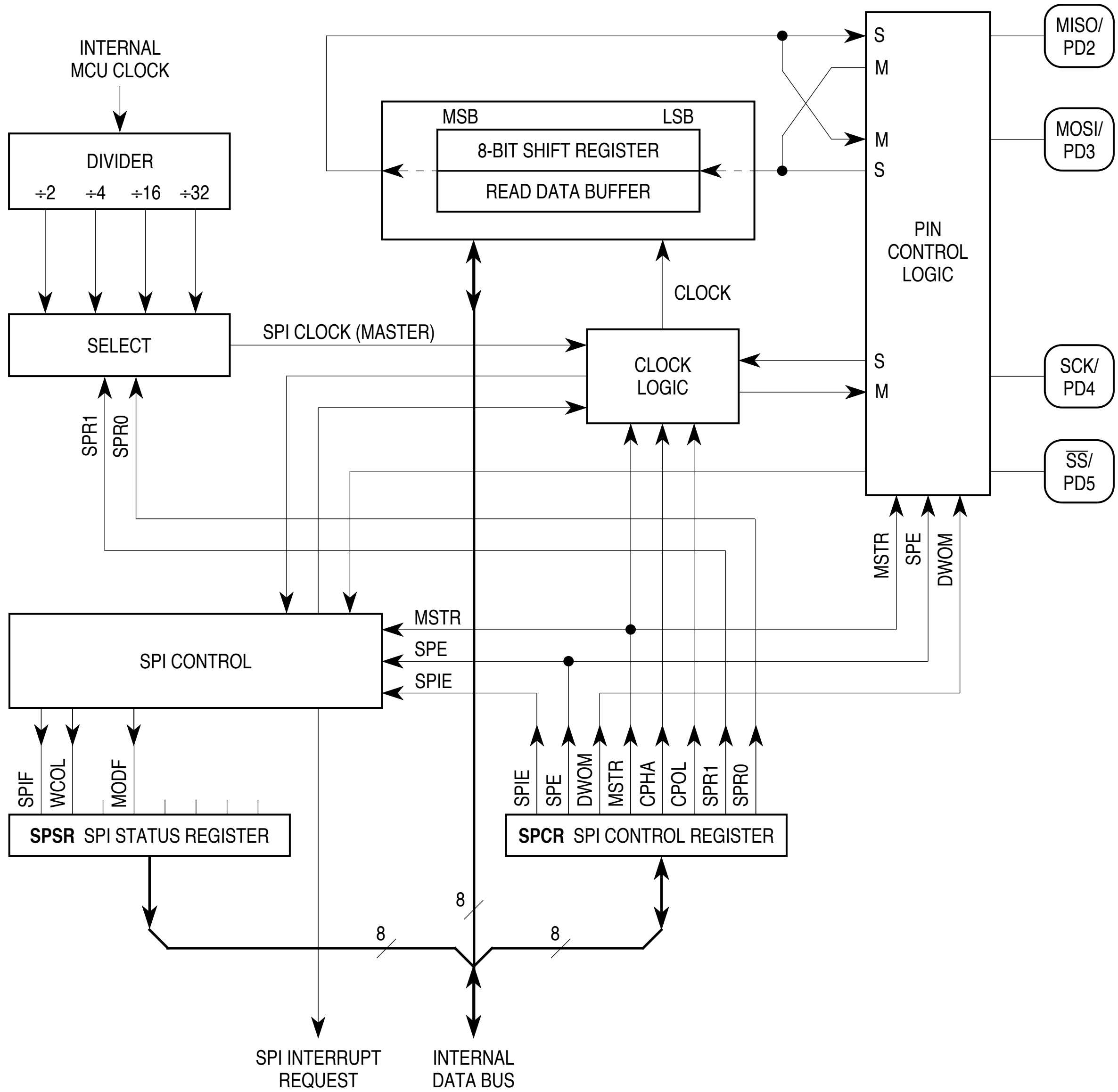


Figure 8-1 SPI Block Diagram

8.2 SPI Transfer Formats

During an SPI transfer, data is simultaneously transmitted and received. A serial clock line synchronizes shifting and sampling of the information on the two serial data lines. A slave select line allows individual selection of a slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. On a master SPI device, the select line can optionally be used to indicate a multiple master bus contention. Refer to **Figure 8-2**.

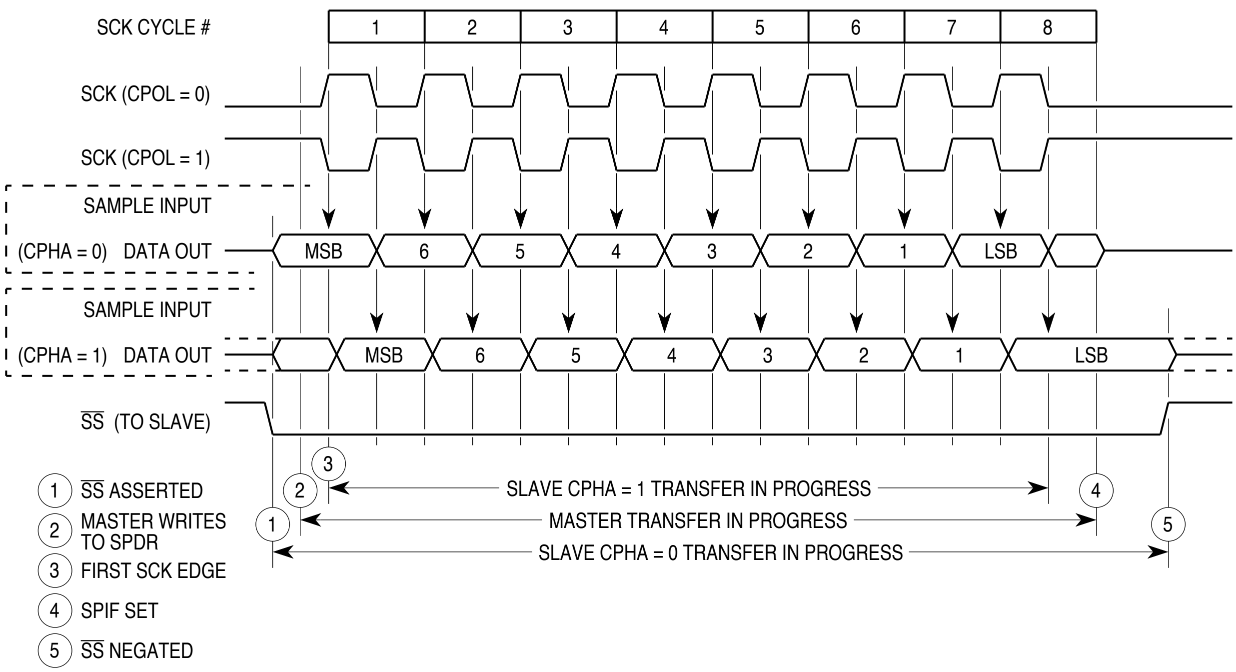


Figure 8-2 SPI Transfer Format

8.2.1 Clock Phase and Polarity Controls

Software can select one of four combinations of serial clock phase and polarity using two bits in the SPI control register (SPCR). The clock polarity is specified by the CPOL control bit, which selects an active high or active low clock, and has no significant effect on the transfer format. The clock phase (CPHA) control bit selects one of two different transfer formats. The clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transfers to allow a master device to communicate with peripheral slaves having different requirements.

When CPHA equals zero, the $\overline{SS}$ line must be negated and reasserted between each successive serial byte. Also, if the slave writes data to the SPI data register (SPDR) while $\overline{SS}$ is low, a write collision error results.

When CPHA equals one, the $\overline{SS}$ line can remain low between successive transfers.

8.3 SPI Signals

The following paragraphs contain descriptions of the four SPI signals: master in slave out (MISO), master out slave in (MOSI), serial clock (SCK), and slave select ($\overline{SS}$).

Any SPI output line must have its corresponding data direction bit in DDRD register set. If the DDR bit is clear, that line is disconnected from the SPI logic and becomes a general-purpose input. All SPI input lines are forced to act as inputs regardless of the state of the corresponding DDR bits in DDRD register.

8.3.1 Master In Slave Out

MISO is one of two unidirectional serial data signals. It is an input to a master device and an output from a slave device. The MISO line of a slave device is placed in the high-impedance state if the slave device is not selected.

8.3.2 Master Out Slave In

The MOSI line is the second of the two unidirectional serial data signals. It is an output from a master device and an input to a slave device. The master device places data on the MOSI line a half-cycle before the clock edge that the slave device uses to latch the data.

8.3.3 Serial Clock

SCK, an input to a slave device, is generated by the master device and synchronizes data movement in and out of the device through the MOSI and MISO lines. Master and slave devices are capable of exchanging a byte of information during a sequence of eight clock cycles.

There are four possible timing relationships that can be chosen by using control bits CPOL and CPHA in the serial peripheral control register (SPCR). Both master and slave devices must operate with the same timing. The SPI clock rate select bits, SPR[1:0], in the SPCR of the master device, select the clock rate. In a slave device, SPR[1:0] have no effect on the operation of the SPI.

8.3.4 Slave Select

The slave select ($\overline{SS}$) input of a slave device must be externally asserted before a master device can exchange data with the slave device. $\overline{SS}$ must be low before data transactions and must stay low for the duration of the transaction.

The $\overline{SS}$ line of the master must be held high. If it goes low, a mode fault error flag (MODF) is set in the serial peripheral status register (SPSR). To disable the mode fault circuit, write a one in bit 5 of the port D data direction register. This sets the $\overline{SS}$ pin to act as a general-purpose output rather than the dedicated input to the slave select circuit, thus inhibiting the mode fault flag. The other three lines are dedicated to the SPI whenever the serial peripheral interface is on.

The state of the master and slave CPHA bits affects the operation of $\overline{SS}$. CPHA settings should be identical for master and slave. When CPHA = 0, the shift clock is the OR of $\overline{SS}$ with SCK. In this clock phase mode, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA = 1, $\overline{SS}$ can be left low between successive SPI characters. In cases where there is only one SPI slave MCU, its $\overline{SS}$ line can be tied to V_{SS} as long as only CPHA = 1 clock mode is used.

8.4 SPI System Errors

Two system errors can be detected by the SPI system. The first type of error arises in a multiple-master system when more than one SPI device simultaneously tries to be a master. This error is called a mode fault. The second type of error, write collision, indicates that an attempt was made to write data to the SPDR while a transfer was in progress.

When the SPI system is configured as a master and the $\overline{SS}$ input line goes to active low, a mode fault error has occurred — usually because two devices have attempted to act as master at the same time. In cases where more than one device is concurrently configured as a master, there is a chance of contention between two pin drivers. For push-pull CMOS drivers, this contention can cause permanent damage. The mode fault mechanism attempts to protect the device by disabling the drivers. The MSTR control bit in the SPCR and all four DDRD control bits associated with the SPI are cleared and an interrupt is generated subject to masking by the SPIE control bit and the I bit in the CCR.

Other precautions may need to be taken to prevent driver damage. If two devices are made masters at the same time, mode fault does not help protect either one unless one of them selects the other as slave. The amount of damage possible depends on the length of time both devices attempt to act as master.

A write collision error occurs if the SPDR is written while a transfer is in progress. Because the SPDR is not double buffered in the transmit direction, writes to SPDR cause data to be written directly into the SPI shift register. Because this write corrupts any transfer in progress, a write collision error is generated. The transfer continues undisturbed, and the write data that caused the error is not written to the shifter.

A write collision is normally a slave error because a slave has no control over when a master initiates a transfer. A master knows when a transfer is in progress, so there is no reason for a master to generate a write-collision error, although the SPI logic can detect write collisions in both master and slave devices.

The SPI configuration determines the characteristics of a transfer in progress. For a master, a transfer begins when data is written to SPDR and ends when SPIF is set. For a slave with CPHA equal to zero, a transfer starts when $\overline{SS}$ goes low and ends when $\overline{SS}$ returns high. In this case, SPIF is set at the middle of the eighth SCK cycle when data is transferred from the shifter to the parallel data register, but the transfer is still in progress until $\overline{SS}$ goes high. For a slave with CPHA equal to one, transfer begins when the SCK line goes to its active level, which is the edge at the beginning of the first SCK cycle. The transfer ends in a slave in which CPHA equals one when SPIF is set.

8.5 SPI Registers

The three SPI registers, SPCR, SPSR, and SPDR, provide control, status, and data storage functions. Refer to the following information for a description of how these registers are organized.

8.5.1 Serial Peripheral Control

SPCR — Serial Peripheral Control Register **$1028**

	Bit 7	6	5	4	3	2	1	Bit 0
	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0
RESET:	0	0	0	0	0	1	U	U

SPIE — Serial Peripheral Interrupt Enable

Set the SPE bit to one to request a hardware interrupt sequence each time the SPIF or MODF status flag is set. SPI interrupts are inhibited if this bit is clear or if the I bit in the condition code register is one.

0 = SPI system interrupts disabled
1 = SPI system interrupts enabled

SPE — Serial Peripheral System Enable

When the SPE bit is set, the port D bit 2, 3, 4, and 5 pins are dedicated to the SPI function. If the SPI is in the master mode and DDRD bit 5 is set, then the port D bit 5 pin becomes a general-purpose output instead of the SS input.

0 = SPI system disabled
1 = SPI system enabled

DWOM — Port D Wired-OR Mode

DWOM affects all port D pins.

0 = Normal CMOS outputs
1 = Open-drain outputs

MSTR — Master Mode Select

0 = Slave mode
1 = Master mode

CPOL — Clock Polarity

When the clock polarity bit is cleared and data is not being transferred, the SCK pin of the master device has a steady state low value. When CPOL is set, SCK idles high. Refer to **Figure 8-2** and **8.2.1 Clock Phase and Polarity Controls**.

CPHA — Clock Phase

The clock phase bit, in conjunction with the CPOL bit, controls the clock-data relationship between master and slave. The CPHA bit selects one of two different clocking protocols. Refer to **Figure 8-2** and **8.2.1 Clock Phase and Polarity Controls**.

SPR[1:0] — SPI Clock Rate Selects

These two bits select the SPI clock (SCK) rate when the device is configured as master. When the device is configured as slave, these bits have no effect. Refer to **Table 8-1**.

Table 8-1 SPI Clock Rates

SPR[1:0]	E Clock Divide By	Frequency at E = 2 MHz	Frequency at E = 3 MHz	Frequency at E = 4 MHz	Frequency at E = 5 MHz
0 0	2	1.0 MHz	1.5 MHz	2.0 MHz	2.5 MHz
0 1	4	500 kHz	750 kHz	1.0 MHz	625 kHz
1 0	16	125 kHz	187.5 kHz	250 kHz	156.25 kHz
1 1	32	62.5 kHz	93.7 kHz	125 kHz	78.125 kHz

8.5.2 Serial Peripheral Status

SPSR — Serial Peripheral Status Register **$1029**

	Bit 7	6	5	4	3	2	1	Bit 0
	SPIF	WCOL	—	MODF	—	—	—	—
RESET:	0	0	0	0	0	0	0	0

SPIF — SPI Interrupt Complete Flag

SPIF is set upon completion of data transfer between the processor and the external device. If SPIF goes high, and if SPIE is set, a serial peripheral interrupt is generated. To clear the SPIF bit, read the SPSR then access the SPDR. Unless SPSR is read (with SPIF set) first, attempts to write SPDR are inhibited.

WCOL — Write Collision

Clearing the WCOL bit is accomplished by reading the SPSR followed by an access of SPDR. Refer to **8.3.4 Slave Select** and **8.4 SPI System Errors**.

0 = No write collision
1 = Write collision

Bit 5 — Not implemented

Always reads zero

MODF — Mode Fault

To clear the MODF bit, read the SPSR then write to the SPCR. Refer to **8.3.4 Slave Select** and **8.4 SPI System Errors**.

0 = No mode fault
1 = Mode fault

Bits [3:0] — Not implemented

Always read zero

8.5.3 Serial Peripheral Data Register

The SPDR is used when transmitting or receiving data on the serial bus. Only a write to this register initiates transmission or reception of a byte, and this only occurs in the master device. At the completion of transferring a byte of data, the SPIF status bit is set in both the master and slave devices.

A read of the SPDR is actually a read of a buffer. To prevent an overrun and the loss of the byte that caused the overrun, the first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated.

SPDR — SPI Data Register **$102A**

Bit 7	6	5	4	3	2	1	Bit 0
Bit 7	6	5	4	3	2	1	Bit 0

SPI is double buffered in and single buffered out.

SECTION 9 TIMING SYSTEM

The M68HC11 timing system is composed of five clock divider chains. The main clock divider chain includes a 16-bit free-running counter, which is driven by a programmable prescaler. The main timer's programmable prescaler provides one of the four clocking rates to drive the 16-bit counter. Two prescaler control bits select the prescale rate.

The prescaler output divides the system clock by 1, 4, 8, or 16. Taps off of this main clocking chain drive circuitry that generates the slower clocks used by the pulse accumulator, the real-time interrupt (RTI), and the computer operating properly (COP) watchdog subsystems, also described in this section. Refer to **Figure 9-1**.

All main timer system activities are referenced to this free-running counter. The counter begins incrementing from $0000 as the MCU comes out of reset, and continues to the maximum count, $FFFF. At the maximum count, the counter rolls over to $0000, sets an overflow flag, and continues to increment. As long as the MCU is running in a normal operating mode, there is no way to reset, change, or interrupt the counting. The capture/compare subsystem features three input capture channels, four output compare channels, and one channel that can be selected to perform either input capture or output compare. Each of the three input capture functions has its own 16-bit input capture register (time capture latch) and each of the output compare functions has its own 16-bit compare register. All timer functions, including the timer overflow and RTI have their own interrupt controls and separate interrupt vectors.

The pulse accumulator contains an 8-bit counter and edge select logic. The pulse accumulator can operate in either event counting mode or gated time accumulation mode. During event counting mode, the pulse accumulator's 8-bit counter increments when a specified edge is detected on an input signal. During gated time accumulation mode, an internal clock source increments the 8-bit counter while an input signal has a predetermined logic level.

The real-time interrupt (RTI) is a programmable periodic interrupt circuit that permits pacing the execution of software routines by selecting one of four interrupt rates.

The COP watchdog clock input ($E \div 2^{15}$) is tapped off of the free-running counter chain. The COP automatically times out unless it is serviced within a specific time by a program reset sequence. If the COP is allowed to time out, a reset is generated, which drives the $\overline{\text{RESET}}$ pin low to reset the MCU and the external system. Refer to **Table 9-1** for crystal related frequencies and periods.

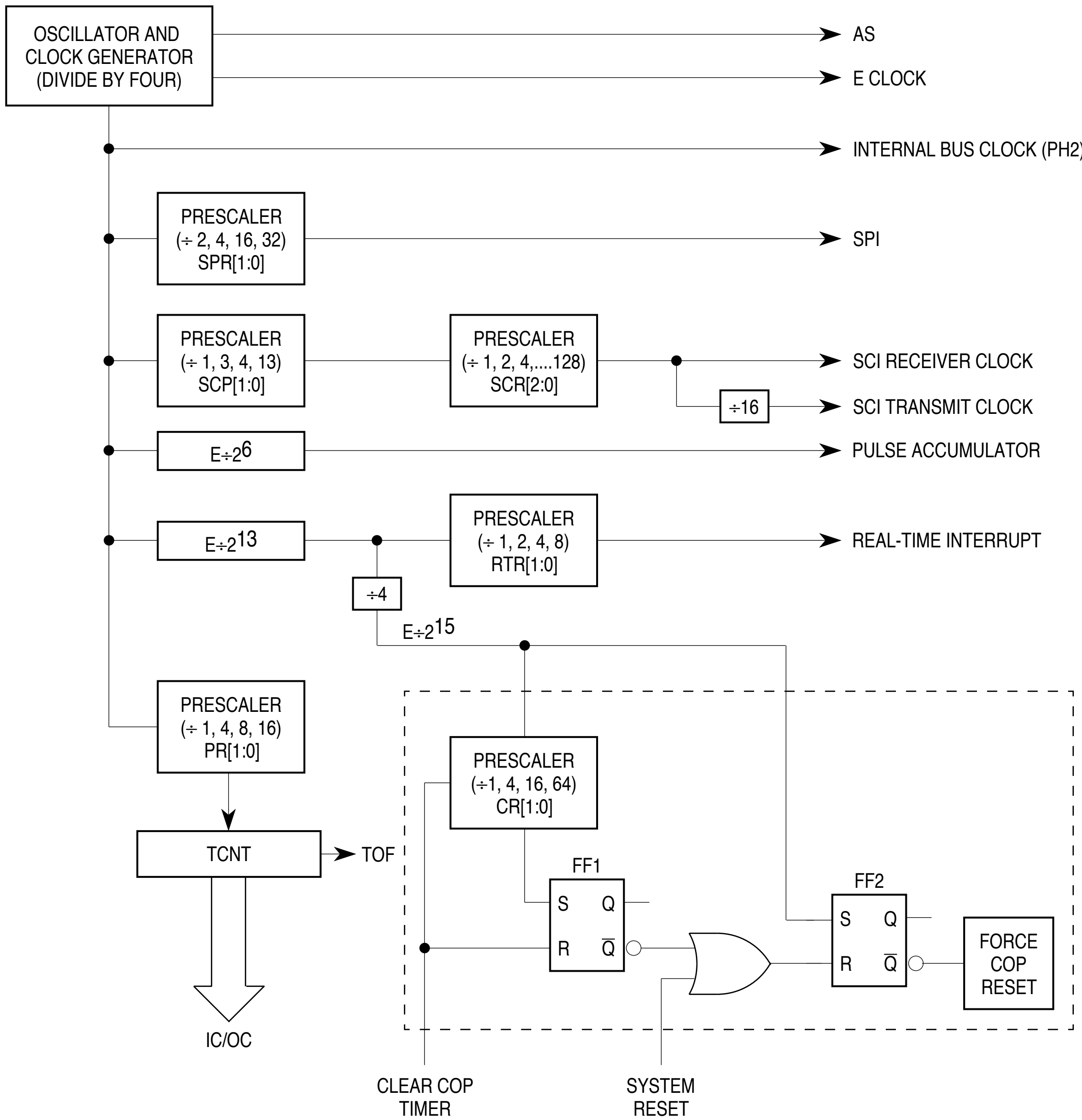


Figure 9-1 Timer Clock Divider Chains

Table 9-1 Timer Summary

Control Bits	XTAL Frequencies				
	4.0 MHz	**8.0 MHz**	**12.0 MHz**	**16.0 MHz**	**Other Rates**
	1.0 MHz	**2.0 MHz**	**3.0 MHz**	**4.0 MHz**	**(E)**
Control Bits	**1000 ns**	**500 ns**	**333 ns**	**250 ns**	**(1/E)**
PR1, PR0	**Main Timer Count Rates**				
0 0 1 count — overflow —	1.0 µs 65.536 ms	500 ns 32.768 ms	333 ns 21.845 ms	250 ns 16.384 ms	(1/E) (2^{16}/E)
0 1 1 count — overflow —	4.0 µs 262.14 ms	2.0 µs 131.07 ms	1.333 µs 87.381 ms	1.0 µs 65.536 ms	(4/E) (2^{18}/E)
1 0 1 count — overflow —	8.0 µs 524.29 ms	4.0 µs 262.14 ms	2.667 µs 174.76 ms	2.0 µs 131.07 ms	(8/E) (2^{19}/E)
1 1 1 count — overflow —	16.0 µs 1.049 s	8.0 µs 524.29 ms	5.333 µs 349.52 ms	4.0 µs 262.14 ms	(16/E) (2^{20}/E)

9.1 Timer Structure

Figure 9-2 shows the capture/compare system block diagram. The port A pin control block includes logic for timer functions and for general-purpose I/O. For pins PA3, PA2, PA1, and PA0, this block contains both the edge-detection logic and the control logic that enables the selection of which edge triggers an input capture. The digital level on PA[3:0] can be read at any time (read PORTA register), even if the pin is being used for the input capture function. Pins PA[6:3] are used for either general-purpose I/O, or as output compare pins. When one of these pins is being used for an output compare function, it cannot be written directly as if it were a general-purpose output. Each of the output compare functions (OC[5:2]) is related to one of the port A output pins. Output compare one (OC1) has extra control logic, allowing it optional control of any combination of the PA[7:3] pins. The PA7 pin can be used as a general-purpose I/O pin, as an input to the pulse accumulator, or as an OC1 output pin.

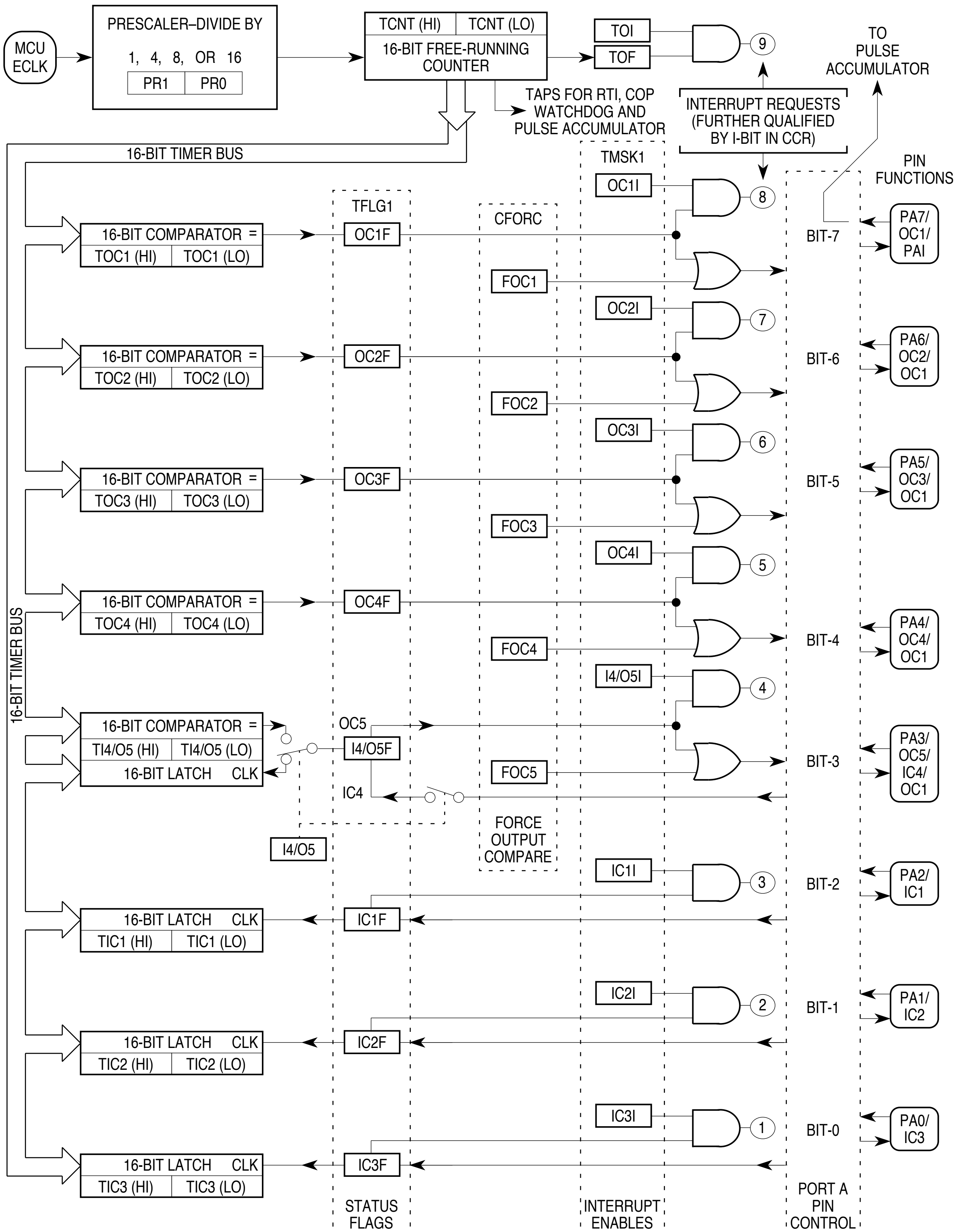


Figure 9-2 Capture/Compare Block Diagram

9.2 Input Capture

The input capture function records the time an external event occurs by latching the value of the free-running counter when a selected edge is detected at the associated timer input pin. Software can store latched values and use them to compute the periodicity and duration of events. For example, by storing the times of successive edges of an incoming signal, software can determine the period and pulse width of a signal. To measure period, two successive edges of the same polarity are captured. To measure pulse width, two alternate polarity edges are captured.

In most cases, input capture edges are asynchronous to the internal timer counter, which is clocked relative to an internal clock (PH2). These asynchronous capture requests are synchronized to PH2 so that the latching occurs on the opposite half cycle of PH2 from when the timer counter is being incremented. This synchronization process introduces a delay from when the edge occurs to when the counter value is detected. Because these delays offset each other when the time between two edges is being measured, the delay can be ignored. When an input capture is being used with an output compare, there is a similar delay between the actual compare point and when the output pin changes state.

The control and status bits that implement the input capture functions are contained in the PACTL, TCTL2, TMSK1, and TFLG1 registers.

To configure port A bit 3 as an input capture, clear the DDA3 bit of the DDRA register. Note that this bit is cleared out of reset. To enable PA3 as the fourth input capture, set the I4/O5 bit in the PACTL register. Otherwise, PA3 is configured as a fifth output compare out of reset, with bit I4/O5 being cleared. If the DDA3 bit is set (configuring PA3 as an output), and IC4 is enabled, then writes to PA3 cause edges on the pin to result in input captures. Writing to TI4/O5 has no effect when the TI4/O5 register is acting as IC4.

9.2.1 Timer Control Register 2

Use the control bits of this register to program input capture functions to detect a particular edge polarity on the corresponding timer input pin. Each of the input capture functions can be independently configured to detect rising edges only, falling edges only, any edge (rising or falling), or to disable the input capture function. The input capture functions operate independently of each other and can capture the same TCNT value if the input edges are detected within the same timer count cycle.

TCTL2 — Timer Control 2 **$1021**

	Bit 7	6	5	4	3	2	1	Bit 0
	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
RESET:	0	0	0	0	0	0	0	0

EDGxB and EDGxA — Input Capture Edge Control

There are four pairs of these bits. Each pair is cleared to zero by reset and must be encoded to configure the corresponding input capture edge detector circuit. IC4 functions only if the I4/O5 bit in the PACTL register is set. Refer to **Table 9-2** for timer control configuration.

9.2.2 Timer Input Capture Registers

When an edge has been detected and synchronized, the 16-bit free-running counter value is transferred into the input capture register pair as a single 16-bit parallel transfer. Timer counter value captures and timer counter incrementing occur on opposite half-cycles of the phase 2 clock so that the count value is stable whenever a capture occurs. The TICx registers are not affected by reset. Input capture values can be read from a pair of 8-bit read-only registers. A read of the high-order byte of an input capture register pair inhibits a new capture transfer for one bus cycle. If a double-byte read instruction, such as LDD, is used to read the captured value, coherency is assured. When a new input capture occurs immediately after a high-order byte read, transfer is delayed for an additional cycle but the value is not lost.

TIC1–TIC3 — Timer Input Capture **$1010–$1015**

$1010	Bit 15	14	13	12	11	10	9	Bit 8	TIC1 (High)
$1011	Bit 7	6	5	4	3	2	1	Bit 0	TIC1 (Low)
$1012	Bit 15	14	13	12	11	10	9	Bit 8	TIC2 (High)
$1013	Bit 7	6	5	4	3	2	1	Bit 0	TIC2 (Low)
$1014	Bit 15	14	13	12	11	10	9	Bit 8	TIC3 (High)
$1015	Bit 7	6	5	4	3	2	1	Bit 0	TIC3 (Low)

TICx not affected by reset.

9.2.3 Timer Input Capture 4/Output Compare 5 Register

Use TI4/O5 as either an input capture register or an output compare register, depending on the function chosen for the PA3 pin. To enable it as an input capture pin, set the I4/O5 bit in the pulse accumulator control register (PACTL) to logic level one. To use it as an output compare register, set the I4/O5 bit to a logic level zero. Refer to **9.6 Pulse Accumulator**.

TI4/O5 — Timer Input Capture 4/Output Compare 5 **$101E, $101F**

$101E	Bit 15	14	13	12	11	10	9	Bit 8	TI4/O5 (High)
$101F	Bit 7	6	5	4	3	2	1	Bit 0	TI4/O5 (Low)

The TI4/O5 register pair resets to ones ($FFFF).

9.3 Output Compare

Use the output compare (OC) function to program an action to occur at a specific time — when the 16-bit counter reaches a specified value. For each of the five output compare functions, there is a separate 16-bit compare register and a dedicated 16-bit comparator. The value in the compare register is compared to the value of the free-running counter on every bus cycle. When the compare register matches the counter value, an output compare status flag is set. The flag can be used to initiate the automatic actions for that output compare function.

To produce a pulse of a specific duration, write a value to the output compare register that represents the time the leading edge of the pulse is to occur. The output compare circuit is configured to set the appropriate output either high or low, depending on the polarity of the pulse being produced. After a match occurs, the output compare register is reprogrammed to change the output pin back to its inactive level at the next match. A value representing the width of the pulse is added to the original value, and then written to the output compare register. Because the pin state changes occur at specific values of the free-running counter, the pulse width can be controlled accurately at the resolution of the free-running counter, independent of software latencies. To generate an output signal of a specific frequency and duty cycle, repeat this pulse-generating procedure.

There are four 16-bit read/write output compare registers: TOC1, TOC2, TOC3, and TOC4, and the TI4/O5 register, which functions under software control as either IC4 or OC5. Each of the OC registers is set to $FFFF on reset. A value written to an OC register is compared to the free-running counter value during each E-clock cycle. If a match is found, the particular output compare flag is set in timer interrupt flag register 1 (TFLG1). If that particular interrupt is enabled in the timer interrupt mask register 1 (TMSK1), an interrupt is generated. In addition to an interrupt, a specified action can be initiated at one or more timer output pins. For OC[5:2], the pin action is controlled by pairs of bits (OMx and OLx) in the TCTL1 register. The output action is taken on each successful compare, regardless of whether or not the OCxF flag in the TFLG1 register was previously cleared.

OC1 is different from the other output compares in that a successful OC1 compare can affect any or all five of the OC pins. The OC1 output action taken when a match is found is controlled by two 8-bit registers with three bits unimplemented: the output compare 1 mask register, OC1M, and the output compare 1 data register, OC1D. OC1M specifies which port A outputs are to be used, and OC1D specifies what data is placed on these port pins.

9.3.1 Timer Output Compare Registers

All output compare registers are 16-bit read-write. Each is initialized to $FFFF at reset. If an output compare register is not used for an output compare function, it can be used as a storage location. A write to the high-order byte of an output compare register pair inhibits the output compare function for one bus cycle. This inhibition prevents inappropriate subsequent comparisons. Coherency requires a complete 16-bit read or write. However, if coherency is not needed, byte accesses can be used.

For output compare functions, write a comparison value to output compare registers TOC1–TOC4 and TI4/O5. When TCNT value matches the comparison value, specified pin actions occur.

TOC1–TOC4 — Timer Output Compare **$1016–$101D**

$1016	Bit 15	14	13	12	11	10	9	Bit 8	TOC1 (High)
$1017	Bit 7	6	5	4	3	2	1	Bit 0	TOC1 (Low)
$1018	Bit 15	14	13	12	11	10	9	Bit 8	TOC2 (High)
$1019	Bit 7	6	5	4	3	2	1	Bit 0	TOC2 (Low)
$101A	Bit 15	14	13	12	11	10	9	Bit 8	TOC3 (High)
$101B	Bit 7	6	5	4	3	2	1	Bit 0	TOC3 (Low)
$101C	Bit 15	14	13	12	11	10	9	Bit 8	TOC4 (High)
$101D	Bit 7	6	5	4	3	2	1	Bit 0	TOC4 (Low)

All TOCx register pairs reset to ones ($FFFF).

9.3.2 Timer Compare Force Register

The CFORC register allows forced early compares. FOC[1:5] correspond to the five output compares. These bits are set for each output compare that is to be forced. The action taken as a result of a forced compare is the same as if there were a match between the OCx register and the free-running counter, except that the corresponding interrupt status flag bits are not set. The forced channels trigger their programmed pin actions to occur at the next timer count transition after the write to CFORC.

The CFORC bits should not be used on an output compare function that is programmed to toggle its output on a successful compare because a normal compare that occurs immediately before or after the force can result in an undesirable operation.

CFORC — Timer Compare Force **$100B**

	Bit 7	6	5	4	3	2	1	Bit 0
	FOC1	FOC2	FOC3	FOC4	FOC5	—	—	—
RESET:	0	0	0	0	0	0	0	0

FOC[1:5] — Force Output Comparison

When the FOC bit associated with an output compare circuit is set, the output compare circuit immediately performs the action it is programmed to do when an output match occurs.

0 = Not affected

1 = Output x action occurs

Bits [2:0] — Not implemented

Always read zero

9.3.3 Output Compare Mask Registers

Use OC1M with OC1 to specify the bits of port A that are affected by a successful OC1 compare. The bits of the OC1M register correspond to PA[7:3].

OC1M — Output Compare 1 Mask **$100C**

	Bit 7	6	5	4	3	2	1	Bit 0
	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	—	—	—
RESET:	0	0	0	0	0	0	0	0

OC1M[7:3] — Output Compare Masks
0 = OC1 is disabled.
1 = OC1 is enabled to control the corresponding pin of port A

Bits [2:0] — Not implemented
Always read zero

9.3.4 Output Compare Data Register

Use this register with OC1 to specify the data that is to be stored on the affected pin of port A after a successful OC1 compare. When a successful OC1 compare occurs, a data bit in OC1D is stored in the corresponding bit of port A for each bit that is set in OC1M.

OC1D — Output Compare 1 Data **$100D**

	Bit 7	6	5	4	3	2	1	Bit 0
	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	—	—	—
RESET:	0	0	0	0	0	0	0	0

If OC1Mx is set, data in OC1Dx is output to port A bit x on successful OC1 compares.

Bits [2:0] — Not implemented
Always read zero

9.3.5 Timer Counter Register

The 16-bit read-only TCNT register contains the prescaled value of the 16-bit timer. A full counter read addresses the most significant byte (MSB) first. A read of this address causes the least significant byte (LSB) to be latched into a buffer for the next CPU cycle so that a double-byte read returns the full 16-bit state of the counter at the time of the MSB read cycle.

TCNT — Timer Counter **$100E, $100F**

$100E	Bit 15	14	13	12	11	10	9	Bit 8	TCNT (High)
$100F	Bit 7	6	5	4	3	2	1	Bit 0	TCNT (Low)

TCNT resets to $0000. In normal modes, TCNT is a read-only register.

9.3.6 Timer Control Register 1

The bits of this register specify the action taken as a result of a successful OCx compare.

TCTL1 — Timer Control 1 **$1020**

	Bit 7	6	5	4	3	2	1	Bit 0
	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5
RESET:	0	0	0	0	0	0	0	0

OM[2:5] — Output Mode

OL[2:5] — Output Level

These control bit pairs are encoded to specify the action taken after a successful OCx compare. OC5 functions only if the I4/O5 bit in the PACTL register is clear. Refer to **Table 9-3** for the coding.

Table 9-2 Timer Output Compare Configuration

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 zero
1	1	Set OCx output line to one

9.3.7 Timer Interrupt Mask Register 1

Use this 8-bit register to enable or inhibit the timer input capture and output compare interrupts.

TMSK1 — Timer Interrupt Mask 1 **$1022**

	Bit 7	6	5	4	3	2	1	Bit 0
	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I
RESET:	0	0	0	0	0	0	0	0

OC1I–OC4I — Output Compare x Interrupt Enable

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

I4/O5I — Input Capture 4/Output Compare 5 Interrupt Enable

When I4/O5 in PACTL is one, I4/O5I is the input capture 4 interrupt enable bit. When I4/O5 in PACTL is zero, I4/O5I is the output compare 5 interrupt enable bit.

IC1I–IC3I — Input Capture x Interrupt Enable

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

NOTE

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

9.3.8 Timer Interrupt Flag Register 1

Bits in this register indicate when timer system events have occurred. Coupled with the bits of TMSK1, the bits of TFLG1 allow the timer subsystem to operate in either a polled or interrupt driven system. Each bit of TFLG1 corresponds to a bit in TMSK1 in the same position.

TFLG1 — Timer Interrupt Flag 1 **$1023**

	Bit 7	6	5	4	3	2	1	Bit 0
	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F
RESET:	0	0	0	0	0	0	0	0

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

OC1F–OC4F — Output Compare x Flag
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 the function enabled by I4/O5 bit in PACTL

IC1F–IC3F — Input Capture x Flag
Set each time a selected active edge is detected on the ICx input line

9.3.9 Timer Interrupt Mask Register 2

Use this 8-bit register to enable or inhibit timer overflow and real-time interrupts. The timer prescaler control bits are included in this register.

TMSK2 — Timer Interrupt Mask 2 **$1024**

	Bit 7	6	5	4	3	2	1	Bit 0
	TOI	RTII	PAOVI	PAII	—	—	PR1	PR0
RESET:	0	0	0	0	0	0	0	0

TOI — Timer Overflow Interrupt Enable
0 = TOF interrupts disabled
1 = Interrupt requested when TOF is set to one

RTII — Real-Time Interrupt Enable
Refer to **9.4 Real-Time Interrupt**.

PAOVI — Pulse Accumulator Overflow Interrupt Enable
Refer to **9.6.3 Pulse Accumulator Status and Interrupt Bits**.

PAII — Pulse Accumulator Input Edge Interrupt Enable
Refer to **9.6.3 Pulse Accumulator Status and Interrupt Bits**.

PR[1:0] — Timer Prescaler Select
These bits are used to select the prescaler divide-by ratio. In normal modes, PR[1:0] can only be written once, and the write must be within 64 cycles after reset. Refer to **Table 9-1** and **Table 9-4** for specific timing values.

Table 9-3 Timer Prescaler Selection

PR[1:0]	Prescaler
0 0	1
0 1	4
1 0	8
1 1	16

NOTE

Bits in TMSK2 correspond bit for bit with flag bits in TFLG2. Ones in TMSK2 enable the corresponding interrupt sources.

9.3.10 Timer Interrupt Flag Register 2

Bits in this register indicate when certain timer system events have occurred. Coupled with the four high-order bits of TMSK2, the bits of TFLG2 allow the timer subsystem to operate in either a polled or interrupt driven system. Each bit of TFLG2 corresponds to a bit in TMSK2 in the same position.

TFLG2 — Timer Interrupt Flag 2 **$1025**

	Bit 7	6	5	4	3	2	1	Bit 0
	TOF	RTIF	PAOVF	PAIF	—	—	—	—
RESET:	0	0	0	0	0	0	0	0

Clear flags by writing a one 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
Refer to **9.4 Real-Time Interrupt**.

PAOVF — Pulse Accumulator Overflow Interrupt Flag
Refer to **9.6 Pulse Accumulator**.

PAIF — Pulse Accumulator Input Edge Interrupt Flag
Refer to **9.6 Pulse Accumulator**.

Bits [3:0] — Not implemented
Always read zero

9.4 Real-Time Interrupt

The real-time interrupt (RTI) feature, used to generate hardware interrupts at a fixed periodic rate, is controlled and configured by two bits (RTR1 and RTR0) in the pulse accumulator control (PACTL) register. The RTII bit in the TMSK2 register enables the interrupt capability. The four different rates available are a product of the MCU oscillator frequency and the value of bits RTR[1:0]. Refer to **Table 9-4**, which shows the periodic real-time interrupt rates.

Table 9-4 RTI Rate Selection

RTR[1:0]	E = 1 MHz	E = 2 MHz	E = 3 MHz	E = 4 MHz	E = X MHz
0 0	8.192 ms	4.096 ms	2.731 ms	2.048 ms	$(2^{13}/E)$
0 1	16.384 ms	8.192 ms	5.461 ms	4.096 ms	$(2^{14}/E)$
1 0	32.768 ms	16.384 ms	10.923 ms	8.192 ms	$(2^{15}/E)$
1 1	65.536 ms	32.768 ms	21.845 ms	16.384 ms	$(2^{16}/E)$

The clock source for the RTI function is a free-running clock that cannot be stopped or interrupted except by reset. This clock causes the time between successive RTI time-outs to be a constant that is independent of the software latencies associated with flag clearing and service. For this reason, an RTI period starts from the previous time-out, not from when RTIF is cleared.

Every time-out causes the RTIF bit in TFLG2 to be set, and if RTII is set, an interrupt request is generated. After reset, one entire RTI period elapses before the RTIF flag is set for the first time. Refer to the TMSK2, TFLG2, and PACTL registers.

9.4.1 Timer Interrupt Mask Register 2

This register contains the real-time interrupt enable bits.

TMSK2 — Timer Interrupt Mask Register 2 **$1024**

	Bit 7	6	5	4	3	2	1	Bit 0
	TOI	RTII	PAOVI	PAII	—	—	PR1	PR0
RESET:	0	0	0	0	0	0	0	0

TOI — Timer Overflow Interrupt Enable
- 0 = TOF interrupts disabled
- 1 = Interrupt requested when TOF is set to one

RTII — Real-Time Interrupt Enable
- 0 = RTIF interrupts disabled
- 1 = Interrupt requested when RTIF set to one

PAOVI — Pulse Accumulator Overflow Interrupt Enable
Refer to **9.6 Pulse Accumulator**.

PAII — Pulse Accumulator Input Edge
Refer to **9.6 Pulse Accumulator**.

PR[1:0] — Timer Prescaler Select
Refer to **Table 9-4**.

NOTE

Bits in TMSK2 correspond bit for bit with flag bits in TFLG2. Ones in TMSK2 enable the corresponding interrupt sources.

9.4.2 Timer Interrupt Flag Register 2

Bits of this register indicate the occurrence of timer system events. Coupled with the four high-order bits of TMSK2, the bits of TFLG2 allow the timer subsystem to operate in either a polled or interrupt driven system. Each bit of TFLG2 corresponds to a bit in TMSK2 in the same position.

TFLG2 — Timer Interrupt Flag 2 **$1025**

	Bit 7	6	5	4	3	2	1	Bit 0
	TOF	RTIF	PAOVF	PAIF	—	—	—	—
RESET:	0	0	0	0	0	0	0	0

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

TOF — Timer Overflow Interrupt Flag
Set when TCNT changes from $FFFF to $0000

RTIF — Real-Time Interrupt Flag
The RTIF status bit is automatically set to one at the end of every RTI period. To clear RTIF, write a byte to TFLG2 with bit 6 set.

PAOVF — Pulse Accumulator Overflow Interrupt Flag
Refer to **9.6 Pulse Accumulator**.

PAIF — Pulse Accumulator Input Edge Interrupt Flag
Refer to **9.6 Pulse Accumulator**.

Bits [3:0] — Not implemented
Always read zero

9.4.3 Pulse Accumulator Control Register

Bits RTR[1:0] of this register select the rate for the RTI system. The remaining bits control the pulse accumulator and IC4/OC5 functions.

PACTL — Pulse Accumulator Control **$1026**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	PAEN	PAMOD	PEDGE	—	I4/O5	RTR1	RTR0
RESET:	0	0	0	0	0	0	0	0

Bit 7 — Not implemented
Always reads zero

PAEN — Pulse Accumulator System Enable
Refer to **9.6 Pulse Accumulator**.

PAMOD — Pulse Accumulator Mode
Refer to **9.6 Pulse Accumulator**.

PEDGE — Pulse Accumulator Edge Control
Refer to **9.6 Pulse Accumulator**.

Bit 3 — Not implemented
Always reads zero

I4/O5 — Input Capture 4/Output Compare
Refer to **9.6 Pulse Accumulator**.

RTR[1:0] — RTI Interrupt Rate Select
These two bits determine the rate at which the RTI system requests interrupts. The RTI system is driven by an E divided by 2^{13} rate clock that is compensated so it is independent of the timer prescaler. These two control bits select an additional division factor. Refer to **Table 9-5**.

9.5 Computer Operating Properly Watchdog Function

The clocking chain for the COP function, tapped off of the main timer divider chain, is only superficially related to the main timer system. The CR[1:0] bits in the OPTION register and the NOCOP bit in the CONFIG register determine the status of the COP function. One additional register, COPRST, is used to arm and clear the COP watchdog reset system. Refer to **SECTION 5 RESETS AND INTERRUPTS** for a more detailed discussion of the COP function.

9.6 Pulse Accumulator

The MC68HC11F1 MCUs have an 8-bit counter that can be configured to operate either as a simple event counter, or for gated time accumulation, depending on the state of the PAMOD bit in the PACTL register. Refer to the pulse accumulator block diagram, **Figure 9-3**.

In the event counting mode, the 8-bit counter is incremented by pulses on an external pin (PAI). The maximum clocking rate for the external event counting mode is the E clock divided by two. In gated time accumulation mode, a free-running E-clock ÷ 64 signal drives the 8-bit counter, but only while the external PAI pin is activated. Refer to **Table 9-6**. The pulse accumulator counter can be read or written at any time.

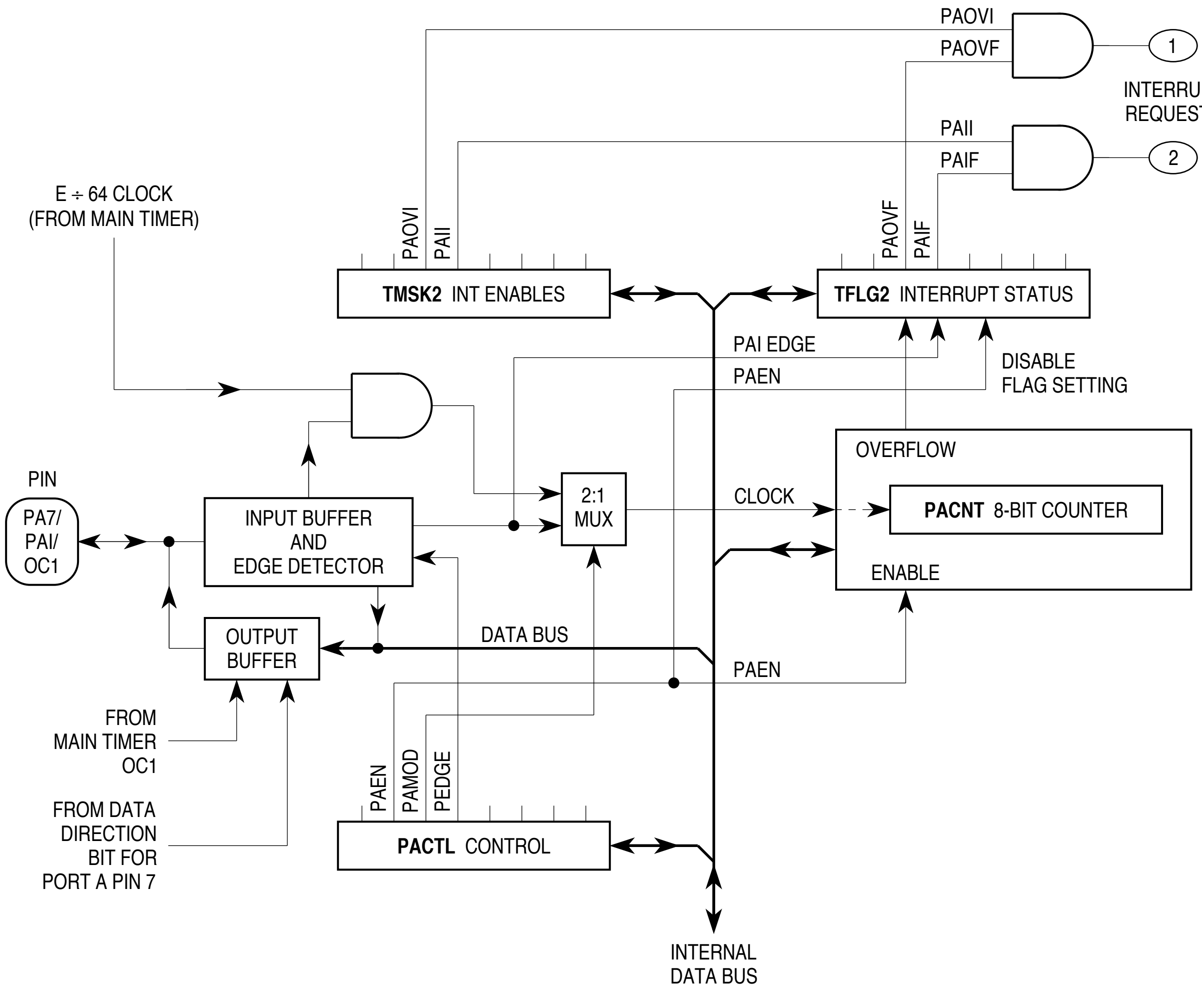


Figure 9-3 Pulse Accumulator

Table 9-5 Pulse Accumulator Timing

Crystal Frequency (4∗E)	E Clock (E)	Cycle Time (1/E)	2^6/E (64/E)	PACNT Overflow (16384/E)
4.0 MHz	1.0 MHz	1000 ns	64 µs	16.384 ms
8.0 MHz	2.0 MHz	500 ns	32 µs	8.192 ms
12.0 MHz	3.0 MHz	333 ns	21.33 µs	5.461 ms
16.0 MHz	4.0 MHz	250 ns	16.0 µs	4.096 ms

Pulse accumulator control bits are also located within two timer registers, TMSK2 and TFLG2, as described in the following paragraphs.

9.6.1 Pulse Accumulator Control Register

Four of this register's bits control an 8-bit pulse accumulator system. Another bit enables either the OC5 function or the IC4 function, while two other bits select the rate for the real-time interrupt system.

PACTL — Pulse Accumulator Control **$1026**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	PAEN	PAMOD	PEDGE	—	I4/O5	RTR1	RTR0
RESET:	0	0	0	0	0	0	0	0

Bit 7 — Not implemented
Always reads zero

PAEN — Pulse Accumulator System Enable
0 = Pulse accumulator disabled
1 = Pulse accumulator enabled

PAMOD — Pulse Accumulator Mode
0 = Event counter
1 = Gated time accumulation

PEDGE — Pulse Accumulator Edge Control
This bit has different meanings depending on the state of the PAMOD bit, as shown in **Table 9-6**.

Table 9-6 Pulse Accumulator Edge Detection Control

PAMOD	PEDGE	Action on Clock
0	0	PAI falling edge increments the counter.
0	1	PAI rising edge increments the counter.
1	0	A zero on PAI inhibits counting.
1	1	A one on PAI inhibits counting.

Bit 3 — Not implemented
Always reads zero

I4/O5 — Input Capture 4/Output Compare 5
0 = Output compare 5 function enable (No IC4)
1 = Input capture 4 function enable (No OC5)

RTR[1:0] — RTI Interrupt Rate Selects
Refer to **9.4 Real-Time Interrupt**.

9.6.2 Pulse Accumulator Count Register

This 8-bit read/write register contains the count of external input events at the PAI input, or the accumulated count. The PACNT is readable even if PAI is not active in gated time accumulation mode. The counter is not affected by reset and can be read or written at any time. Counting is synchronized to the internal PH2 clock so that incrementing and reading occur during opposite half cycles.

PACNT — Pulse Accumulator Count **$1027**

Bit 7	6	5	4	3	2	1	Bit 0
Bit 7	6	5	4	3	2	1	Bit 0

9.6.3 Pulse Accumulator Status and Interrupt Bits

The pulse accumulator control bits, PAOVI, PAII, PAOVF, and PAIF are located within timer registers TMSK2 and TFLG2.

TMSK2 — Timer Interrupt Mask 2 Register **$1024**

	Bit 7	6	5	4	3	2	1	Bit 0
	TOI	RTII	PAOVI	PAII	—	—	PR1	PR0
RESET:	0	0	0	0	0	0	0	0

TFLG2 — Timer Interrupt Flag 2 Register **$1025**

	Bit 7	6	5	4	3	2	1	Bit 0
	TOF	RTIF	PAOVF	PAIF	—	—	—	—
RESET:	0	0	0	0	0	0	0	0

PAOVI and PAOVF — Pulse Accumulator Interrupt Enable and Overflow Flag

The PAOVF status bit is set each time the pulse accumulator count rolls over from $FF to $00. To clear this status bit, write a one in the corresponding data bit position (bit 5) of the TFLG2 register. The PAOVI control bit allows configuring the pulse accumulator overflow for polled or interrupt-driven operation and does not affect the state of PAOVF. When PAOVI is zero, pulse accumulator overflow interrupts are inhibited, and the system operates in a polled mode, which requires that PAOVF be polled by user software to determine when an overflow has occurred. When the PAOVI control bit is set, a hardware interrupt request is generated each time PAOVF is set. Before leaving the interrupt service routine, software must clear PAOVF by writing to the TFLG2 register.

PAII and PAIF — Pulse Accumulator Input Edge Interrupt Enable and Flag

The PAIF status bit is automatically set each time a selected edge is detected at the PA7/PAI/OC1 pin. To clear this status bit, write to the TFLG2 register with a one in the corresponding data bit position (bit 4). The PAII control bit allows configuring the pulse accumulator input edge detect for polled or interrupt-driven operation but does not affect setting or clearing the PAIF bit. When PAII is zero, pulse accumulator input interrupts are inhibited, and the system operates in a polled mode. In this mode, the PAIF bit must be polled by user software to determine when an edge has occurred. When the PAII control bit is set, a hardware interrupt request is generated each time PAIF is set. Before leaving the interrupt service routine, software must clear PAIF by writing to the TFLG2 register.

SECTION 10 ANALOG-TO-DIGITAL CONVERTER

The analog-to-digital (A/D) system, a successive approximation converter, uses an all-capacitive charge redistribution technique to convert analog signals to digital values.

10.1 Overview

The A/D system is an 8-channel, 8-bit, multiplexed-input converter. The AV_{DD} pin is used to input supply voltage to the A/D converter. This allows the supply voltage to be bypassed independently. The converter does not require external sample and hold circuits because of the type of charge redistribution technique used. A/D converter timing can be synchronized to the system E clock, or to an internal resistor capacitor (RC) oscillator. The A/D converter system consists of four functional blocks: multiplexer, analog converter, digital control, and result storage. Refer to **Figure 10-1**.

10.1.1 Multiplexer

The multiplexer selects one of 16 inputs for conversion. Input selection is controlled by the value of bits CD–CA in the ADCTL register. The eight port E pins are fixed-direction analog inputs to the multiplexer, and internal analog signal lines are routed to it.

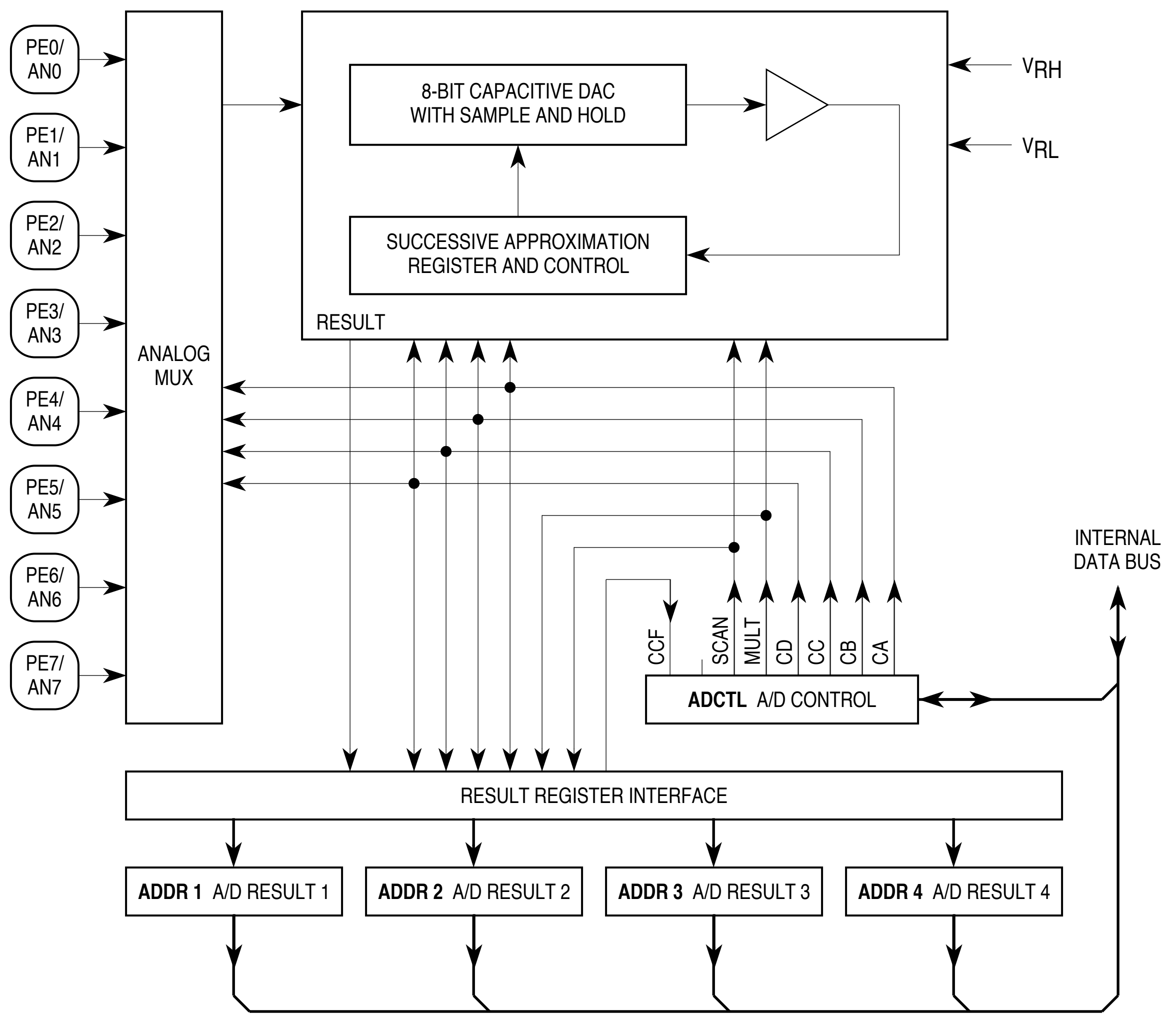
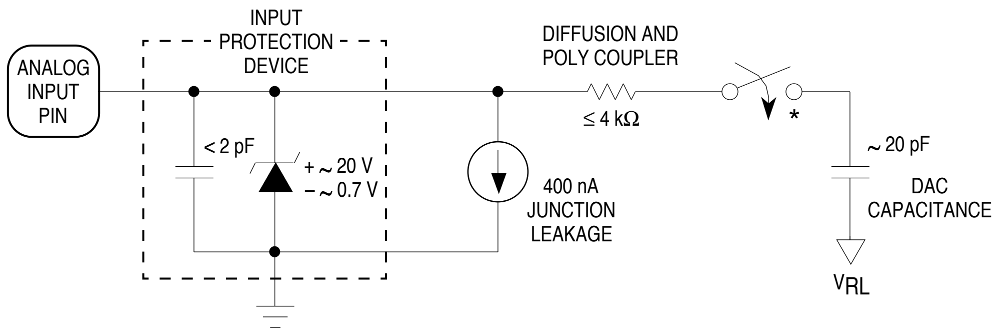


Figure 10-1 A/D Converter Block Diagram

Port E pins can also be used as digital inputs. Reads of port E pins are not recommended during the sample portion of an A/D conversion cycle, when the gate signal to the N-channel input gate is on. Because no P-channel devices are directly connected to either input pins or reference voltage pins, voltages above V_{DD} do not cause a latchup problem, although current should be limited according to maximum ratings. Refer to **Figure 10-2**, which is a functional diagram of an input pin.



* This analog switch is closed only during the 12-cycle sample time.

Figure 10-2 Electrical Model of an A/D Input Pin (Sample Mode)

10.1.2 Analog Converter

Conversion of an analog input selected by the multiplexer occurs in this block. It contains a digital-to-analog capacitor (DAC) array, a comparator, and a successive approximation register (SAR). Each conversion is a sequence of eight comparison operations, beginning with the most significant bit (MSB). Each comparison determines the value of a bit in the successive approximation register.

The DAC array performs two functions. It acts as a sample and hold circuit during the entire conversion sequence, and provides comparison voltage to the comparator during each successive comparison.

The result of each successive comparison is stored in the SAR. When a conversion sequence is complete, the contents of the SAR are transferred to the appropriate result register.

A charge pump provides switching voltage to the gates of analog switches in the multiplexer. Charge pump output must stabilize between 7 and 8 volts, thus a delay of up to 100 µs must be imposed after setting ADPU before the converter can be used. The charge pump is enabled by the ADPU bit in the OPTION register.

Power is provided to the A/D converter system through the AV_{DD} and AV_{SS} pins.

10.1.3 Digital Control

All A/D converter operations are controlled by bits in register ADCTL. In addition to selecting the analog input to be converted, ADCTL bits indicate conversion status, and control whether single or continuous conversions are performed. Finally, the ADCTL bits determine whether conversions are performed on single or multiple channels.

10.1.4 Result Registers

Four 8-bit registers (ADR1–ADR4) store conversion results. Each of these registers can be accessed by the processor in the CPU. The conversion complete flag (CCF) indicates when valid data is present in the result registers. The result registers are written during a portion of the system clock cycle when reads do not occur, so there is no conflict.

10.1.5 A/D Converter Clocks

The CSEL bit in the OPTION register selects whether the A/D converter uses the system E clock or an internal RC oscillator for synchronization. When the A/D system is operating with the MCU E clock, all switching and comparator functions are synchronized to the MCU clocks. This allows the comparator results to be sampled at relatively quiet clock times to minimize noise errors.

When E-clock frequency is below 750 kHz, charge leakage in the capacitor array can cause errors, and the internal oscillator should be used. The RC clock is asynchronous to the MCU internal E clock. Therefore, when the RC clock is used, additional errors can occur because the comparator is sensitive to the additional system clock noise.

10.1.6 Conversion Sequence

A/D converter operations are performed in sequences of four conversions each. A conversion sequence can repeat continuously or stop after one iteration. The conversion complete flag (CCF) is set after the fourth conversion in a sequence to show the availability of data in the result registers. **Figure 10-3** shows the timing of a typical sequence. Synchronization is referenced to the system E clock.

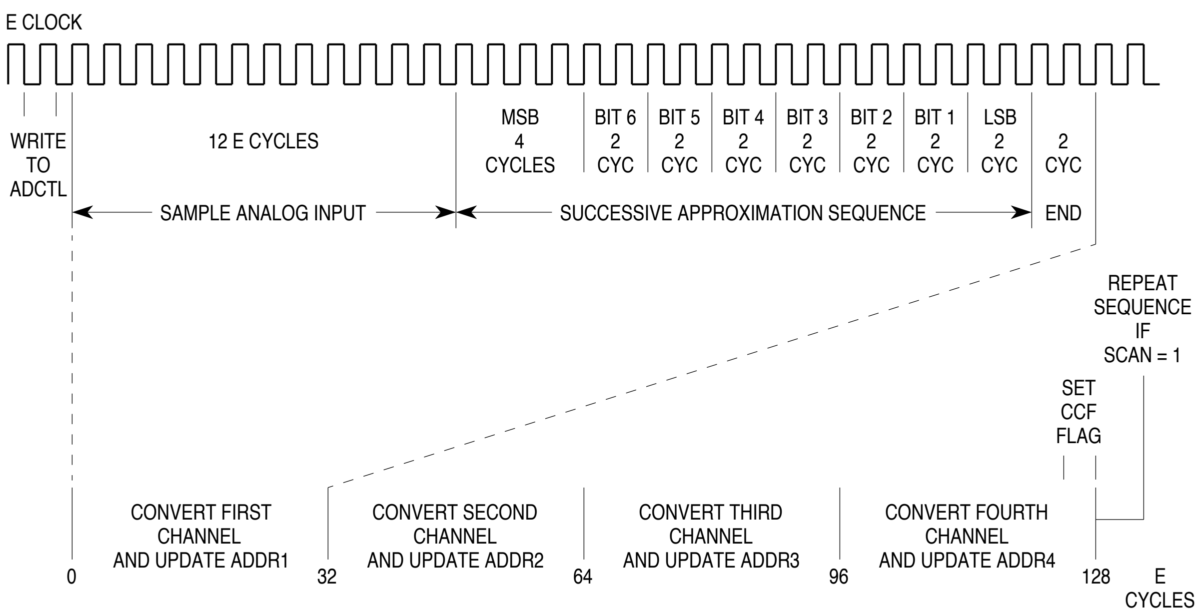


Figure 10-3 A/D Conversion Sequence

10.2 A/D Converter Power-Up and Clock Select

Bit 7 of the OPTION register controls A/D converter power up. Clearing ADPU removes power from and disables the A/D converter system. Setting ADPU enables the A/D converter system. Stabilization of the analog bias voltages requires a delay of as much as 100 µs after turning on the A/D converter. When the A/D converter system is operating with the MCU E clock, all switching and comparator operations are synchronized to the MCU clocks. This allows the comparator results to be sampled at quiet times, which minimizes noise errors. The internal RC oscillator is asynchronous to the MCU clock, so noise affects A/D converter results, which lowers accuracy slightly while CSEL = 1.

OPTION — System Configuration Options **$1039**

	Bit 7	6	5	4	3	2	1	Bit 0
	ADPU	CSEL	IRQE*	DLY*	CME	FCME*	CR1*	CR0*
RESET:	0	0	0	1	0	0	0	0

*Can be written only once in first 64 cycles out of reset in normal modes, or at any time in special modes

ADPU — A/D Power-Up
    0 = A/D powered down
    1 = A/D powered up

CSEL — Clock Select
    0 = A/D and EEPROM use system E clock
    1 = A/D and EEPROM use internal RC clock

IRQE — Configure $\overline{\text{IRQ}}$ for Edge-Sensitive Only Operation
    Refer to **SECTION 5 RESETS AND INTERRUPTS**.

DLY — Enable Oscillator Start-up Delay
    Refer to **SECTION 5 RESETS AND INTERRUPTS**.

CME — Clock Monitor Enable
    Refer to **SECTION 5 RESETS AND INTERRUPTS**.

FCME — Force Clock Monitor Enable
    Refer to **SECTION 5 RESETS AND INTERRUPTS**.

CR[1:0] — COP Timer Rate Select Bits
    Refer to **SECTION 5 RESETS AND INTERRUPTS** and **SECTION 9 TIMING SYSTEM**.

10.3 Conversion Process

The A/D conversion sequence begins one E-clock cycle after a write to the A/D control/status register, ADCTL. The bits in ADCTL select the channel and the mode of conversion.

An input voltage equal to V_{RL} converts to $00 and an input voltage equal to V_{RH} converts to $FF (full scale), with no overflow indication. For ratiometric conversions of this type, the source of each analog input should use V_{RH} as the supply voltage and be referenced to V_{RL}.

10.4 Channel Assignments

The multiplexer allows the A/D converter to select one of sixteen analog signals. Eight of these channels correspond to port E input lines, four of the channels are internal reference points or test functions, and four channels are reserved. Refer to **Table 10-1**.

Table 10-1 A/D Converter Channel Assignments

Channel Number	Channel Signal	Result in ADRx if MULT = 1
1	AN0	ADR1
2	AN1	ADR2
3	AN2	ADR3
4	AN3	ADR4
5	AN4	ADR1
6	AN5	ADR2
7	AN6	ADR3
8	AN7	ADR4
9	Reserved	—
10	Reserved	—
11	Reserved	—
12	Reserved	—
13	V_{RH}*	ADR1
14	V_{RL}*	ADR2
15	$(V_{RH})/2$*	ADR3
16	Reserved*	ADR4

*Used for factory testing

10.5 Single-Channel Operation

There are two types of single-channel operation. When SCAN = 0, the first type, the single selected channel is converted four consecutive times. The first result is stored in A/D result register 1 (ADR1), and the fourth result is stored in ADR4. After the fourth conversion is complete, all conversion activity is halted until a new conversion command is written to the ADCTL register. In the second type of single-channel operation, SCAN = 1, conversions continue to be performed on the selected channel with the fifth conversion being stored in register ADR1 (overwriting the first conversion result), the sixth conversion overwriting ADR2, and so on.

10.6 Multiple-Channel Operation

There are two types of multiple-channel operation. When SCAN = 0, the first type, a selected group of four channels is converted one time each. The first result is stored in A/D result register 1 (ADR1), and the fourth result is stored in ADR4. After the fourth conversion is complete, all conversion activity is halted until a new conversion command is written to the ADCTL register. In the second type of multiple-channel operation, SCAN = 1, conversions continue to be performed on the selected group of channels with the fifth conversion being stored in register ADR1 (replacing the earlier conversion result for the first channel in the group), the sixth conversion overwriting ADR2, and so on.

10.7 Operation in STOP and WAIT Modes

If a conversion sequence is in progress when either the STOP or WAIT mode is entered, the conversion of the current channel is suspended. When the MCU resumes normal operation, that channel is resampled and the conversion sequence is resumed. As the MCU exits the WAIT mode, the A/D circuits are stable and valid results can be obtained on the first conversion. However, in STOP mode, all analog bias currents are disabled and it is necessary to allow a stabilization period when leaving the STOP mode. If the STOP mode is exited with a delay (DLY = 1), there is enough time for these circuits to stabilize before the first conversion. If the STOP mode is exited with no delay (DLY bit in OPTION register = 0), allow 10 ms for the A/D circuitry to stabilize to avoid invalid results.

10.8 A/D Control/Status Registers

All bits in this register can be read or written, except CCF (bit 7), which is a read-only status indicator, and bit 6, which always reads as zero. Write to ADCTL to initiate a conversion. To quit a conversion in progress, write to this register and a new conversion sequence begins immediately.

ADCTL — A/D Control/Status **$1030**

	Bit 7	6	5	4	3	2	1	Bit 0
	CCF	—	SCAN	MULT	CD	CC	CB	CA
RESET:	1	0	I	I	I	I	I	I

CCF — Conversions Complete Flag

A read-only status indicator, this bit is set when all four A/D result registers contain valid conversion results. Each time the ADCTL register is overwritten, this bit is automatically cleared to zero and a conversion sequence is started. In the continuous mode, CCF is set at the end of the first conversion sequence.

Bit 6 — Not implemented

Always reads zero

SCAN — Continuous Scan Control

When this control bit is clear, the four requested conversions are performed once to fill the four result registers. When this control bit is set, conversions continue in a round-robin fashion with the result registers updated as data becomes available.

MULT — Multiple Channel/Single Channel Control

When this bit is clear, the A/D converter system is configured to perform four consecutive conversions on the single channel specified by the four channel select bits CD–CA (bits [3:0] of the ADCTL register). When this bit is set, the A/D system is configured to perform a conversion on each of four channels where each result register corresponds to one channel.

NOTE

When the multiple-channel continuous scan mode is used, extra care is needed in the design of circuitry driving the A/D inputs. The charge on the capacitive DAC array before the sample time is related to the voltage on the previously converted channel. A charge share situation exists between the internal DAC capacitance and the external circuit capacitance. Although the amount of charge involved is small, the rate at which it is repeated is every 64 µs for an E clock of 2 MHz. The RC charging rate of the external circuit must be balanced against this charge sharing effect to avoid errors in accuracy. Refer to *M68HC11 Reference Manual* (M68HC11RM/AD) for further information.

CD–CA — Channel Selects D–A

Refer to **Table 10-2**. When a multiple channel mode is selected (MULT = 1), the two least significant channel select bits (CB and CA) have no meaning and the CD and CC bits specify which group of four channels is to be converted.

Table 10-2 A/D Converter Channel Selection

Channel Select Control Bits	Channel Signal	Result in ADRx if MULT = 1
CD:CC:CB:CA		
0000	AN0	ADR1
0001	AN1	ADR2
0010	AN2	ADR3
0011	AN3	ADR4
0100	AN4	ADR1
0101	AN5	ADR2
0110	AN6	ADR3
0111	AN7	ADR4
1000	Reserved	—
1001	Reserved	—
1010	Reserved	—
1011	Reserved	—
1100	V_{RH}*	ADR1
1101	V_{RL}*	ADR2
1110	$(V_{RH})/2$*	ADR3
1111	Reserved*	ADR4

*Used for factory testing

10.9 A/D Converter Result Registers

These read-only registers hold an 8-bit conversion result. Writes to these registers have no effect. Data in the A/D converter result registers is valid when the CCF flag in the ADCTL register is set, indicating a conversion sequence is complete. If conversion results are needed sooner, refer to **Figure 10-3**, which shows the A/D conversion sequence diagram.

ADR1–ADR4 — A/D Results **$1031–$1034**

$1031	Bit 7	6	5	4	3	2	1	Bit 0	ADR1
$1032	Bit 7	6	5	4	3	2	1	Bit 0	ADR2
$1033	Bit 7	6	5	4	3	2	1	Bit 0	ADR3
$1034	Bit 7	6	5	4	3	2	1	Bit 0	ADR4

APPENDIX A ELECTRICAL CHARACTERISTICS

This appendix contains electrical parameters for the MC68HC11F1 microcontroller.

Table A-1 Maximum Ratings

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	– 0.3 to + 7.0	V
Input Voltage	V_{in}	– 0.3 to + 7.0	V
Operating Temperature Range MC68HC11F1 MC68HC11F1C MC68HC11F1V MC68HC11F1M	T_A	T_L to T_H 0 to + 70 – 40 to + 85 – 40 to + 105 – 40 to + 125	°C
Storage Temperature Range	T_{stg}	– 55 to + 150	°C
Current Drain per Pin* Excluding V_{DD}, V_{SS}, AV_{DD}, V_{RH}, and V_{RL}	I_D	25	mA

*One pin at a time, observing maximum power dissipation limits.

Internal circuitry protects the inputs against damage caused by high static voltages or electric fields; however, normal precautions are necessary to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Extended operation at the maximum ratings can adversely affect device reliability. Tying unused inputs to an appropriate logic voltage level (either GND or V_{DD}) enhances reliability of operation.

Table A-2 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Average Junction Temperature	T_J	$T_A + (P_D \times Q_{JA})$	°C
Ambient Temperature	T_A	User-determined	°C
Package Thermal Resistance (Junction-to-Ambient) 68-Pin Plastic Leaded Chip Carrier 80-Pin Low Profile Quad Flat Pack (LQFP, 1.4 mm Thick)	Q_{JA}	 50 80	 °C/W °C/W
Total Power Dissipation (Note 1)	P_D	$P_{INT} + P_{I/O}$ $K / (T_J + 273°C)$	W
Device Internal Power Dissipation	P_{INT}	$I_{DD} \times V_{DD}$	W
I/O Pin Power Dissipation (Note 2)	$P_{I/O}$	User-determined	W
A Constant (Note 3)	K	$P_D \times (T_A + 273°C) +$ $Q_{JA} \times P_D^2$	W x °C

NOTES:

1 This is an approximate value, neglecting $P_{I/O}$.

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

3. K is a constant pertaining to the device. Solve for K with a known T_A and a measured P_D (at equilibrium. Use this value of K to solve for P_D and T_J iteratively for any value of T_A

Table A-3 DC Electrical Characteristics

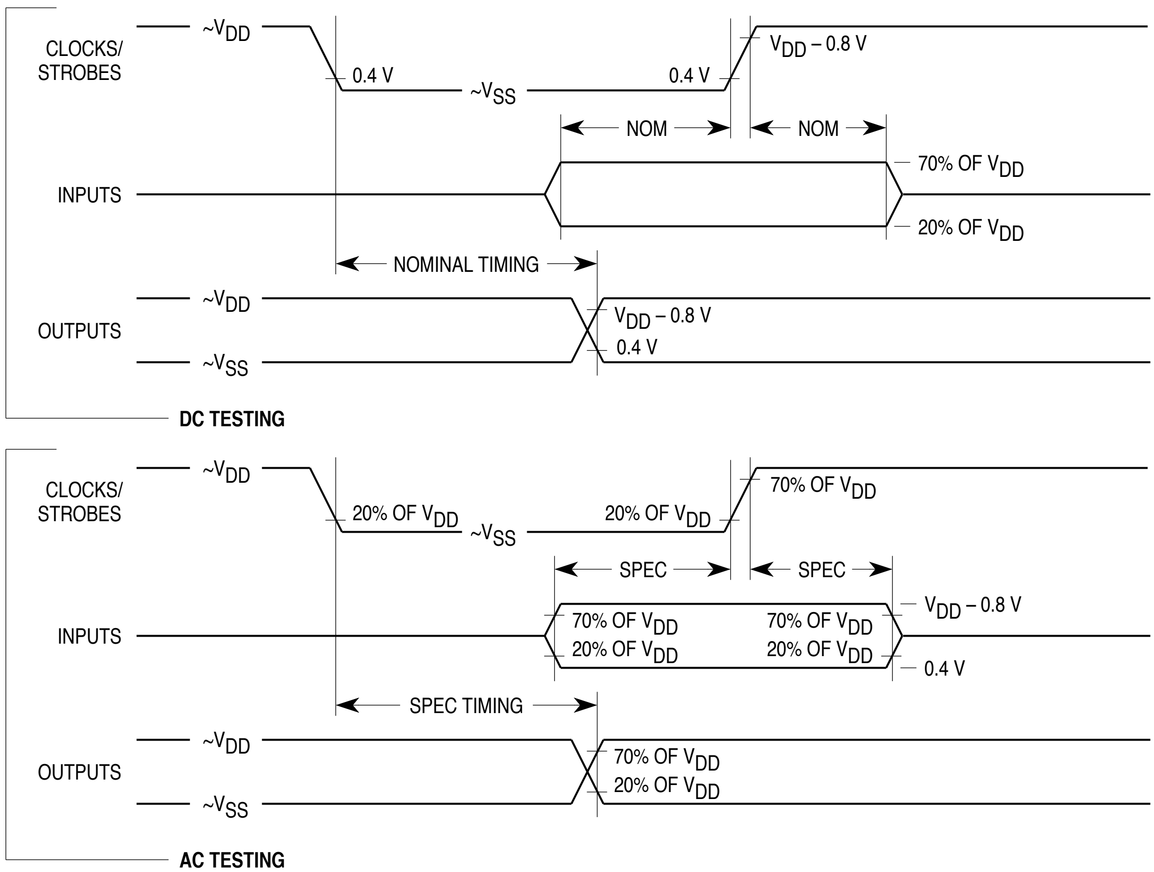
V_{DD} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted

Characteristic	Symbol	Min	Max	Unit
Output Voltage (Note 1) — All Outputs except XTAL All Outputs Except XTAL, $\overline{RESET}$, and MODA I_{Load} = ± 10.0 µA	V_{OL} V_{OH}	— V_{DD} – 0.1	0.1 —	V V
Output High Voltage (Note 1) All Outputs Except XTAL, $\overline{RESET}$, and MODA I_{Load} = – 0.8 mA, V_{DD} = 4.5 V	V_{OH}	V_{DD} – 0.8	—	V
Output Low Voltage — All Outputs Except XTAL I_{Load} = 1.6 mA	V_{OL}	—	0.4	V
Input High Voltage — All Inputs Except $\overline{RESET}$ $\overline{RESET}$	V_{IH}	0.7 x V_{DD} 0.8 x V_{DD}	V_{DD} + 0.3 V_{DD} + 0.3	V V
Input Low Voltage — All Inputs	V_{IL}	V_{SS} – 0.3	0.2 x V_{DD}	V
I/O Ports, Three-State Leakage — Ports A, B, C, D, F, G $V_{in} = V_{IH}$ or V_{IL} — MODA/$\overline{LIR}$, $\overline{RESET}$	I_{OZ}	—	±10	µA
Input Leakage Current (Note 2) $V_{in} = V_{DD}$ or V_{SS} — $\overline{IRQ}$, $\overline{XIRQ}$ on standard devices $V_{in} = V_{DD}$ or V_{SS} — MODB/V_{STBY}, $\overline{XIRQ}$ on EPROM devices	I_{in}	— —	±1 ±10	µA µA
Input Current with Pull-Up Resistors $V_{in} = V_{IL}$ — Ports B, F, and G	I_{ipr}	100	500	µA
RAM Standby Voltage — Power down	V_{SB}	4.0	V_{DD}	V
RAM Standby Current — Power down	I_{SB}	—	20	µA
Input Capacitance — PE[7:0], $\overline{IRQ}$, $\overline{XIRQ}$, EXTAL Ports A, B, C, D, F, G, MODA/$\overline{LIR}$, $\overline{RESET}$	C_{in}	— —	8 12	pF pF
Output Load Capacitance — All Outputs Except PD[4:1], 4XOUT, XTAL, MODA/$\overline{LIR}$ PD[4:1] 4XOUT	C_L	— — —	90 200 30	pF pF pF

Characteristic	Symbol	2 MHz	3 MHz	4 MHz	Unit
Maximum Total Supply Current (Note 3) **RUN:** Expanded Mode	I_{DD}	27	38	50	mA
WAIT: (All Peripheral Functions Shut Down) Expanded Mode	W_{IDD}	15	20	25	mA
STOP: No Clocks, Expanded Mode	S_{IDD}	50	50	50	µA
Maximum Power Dissipation Expanded Mode	P_D	149	209	275	mW

NOTES:

1. V_{OH} specification for $\overline{RESET}$ and MODA is not applicable because they are open-drain pins. V_{OH} specification not applicable to ports C and D in wired-OR mode.
2. Refer to A/D specification for leakage current for port E.
3. EXTAL is driven with a square wave, and
   t_{cyc} = 500 ns for 2 MHz rating;
   t_{cyc} = 333 ns for 3 MHz rating;
   t_{cyc} = 250 ns for 4 MHz rating;
   $V_{IL} \leq 0.2$ V; $V_{IH} \geq V_{DD} – 0.2$ V; No dc loads.



NOTES:

1. Full test loads are applied during all DC electrical tests and AC timing measurements.
2. During AC timing measurements, inputs are driven to 0.4 volts and V_{DD} – 0.8 volts while timing measurements are taken at the 20% and 70% of V_{DD} points.

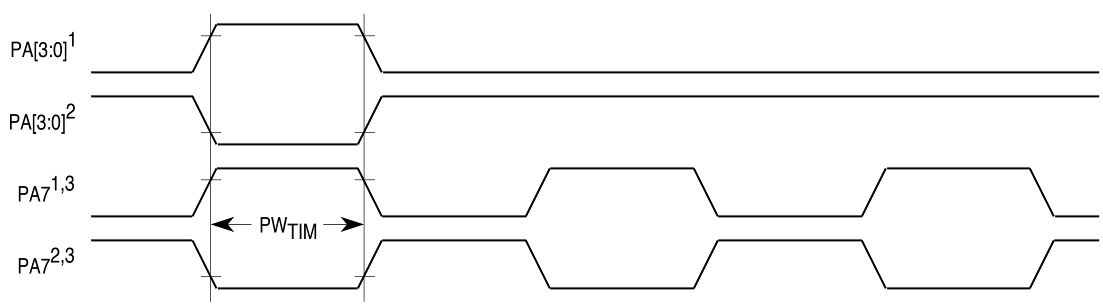
Figure A-1 Test Methods

Table A-4 Control Timing

V_{DD} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H

Characteristic	Symbol	2.0 MHz		3.0 MHz		4.0 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_o	dc	2.0	dc	3.0	dc	4.0	MHz
E-Clock Period	t_{cyc}	500	—	333	—	250	—	ns
Crystal Frequency	f_{XTAL}	—	8.0	—	12.0	—	16.0	MHz
External Oscillator Frequency	4 f_o	dc	8.0	dc	12.0	dc	16.0	MHz
Processor Control Setup Time t_{PCSU} = 1/4 t_{cyc} + 50 ns	t_{PCSU}	175	—	133	—	113	—	ns
Reset Input Pulse Width (Notes 2, 3) (To Guarantee External Reset Vector) (Minimum Input Time; Can Be Preempted by Internal Reset)	PW_{RSTL}	16 1	— —	16 1	— —	16 1	— —	t_{cyc} t_{cyc}
Mode Programming Setup Time	t_{MPS}	2	—	2	—	2	—	t_{cyc}
Mode Programming Hold Time	t_{MPH}	10	—	10	—	10	—	ns
Interrupt Pulse Width, $\overline{IRQ}$ Edge-Sensitive Mode $PW_{IRQ} = t_{cyc}$ + 20 ns	PW_{IRQ}	520	—	353	—	270	—	ns
Wait Recovery Startup Time	t_{WRS}	—	4	—	4	—	4	t_{cyc}
Timer Pulse Width, Input Capture Pulse Accumulator Input $PW_{TIM} = t_{cyc}$ + 20 ns	PW_{TIM}	520	—	353	—	270	—	ns

NOTES:
1. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. $\overline{RESET}$ is recognized during the first clock cycle it is held low. Internal circuitry then drives the pin low for four clock cycles, releases the pin, and samples the pin level two cycles later to determine the source of the interrupt. Refer to **SECTION 5 RESETS AND INTERRUPTS** for further detail.
3. $PW_{RSTL} = 8\ t_{cyc}$ minimum on mask set C94R only.



NOTES:
1. Rising edge sensitive input.
2. Falling edge sensitive input.
3. Maximum pulse accumulator clocking rate is E-clock frequency divided by 2.

Figure A-2 Timer Inputs

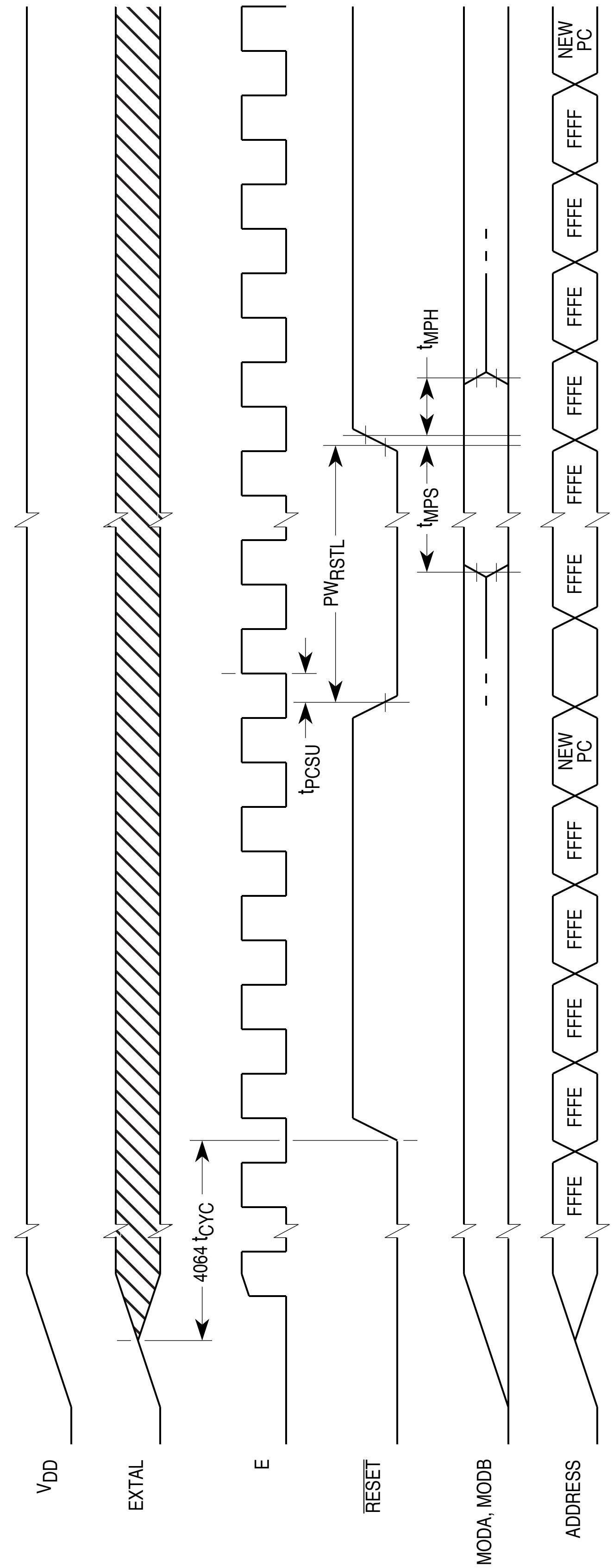
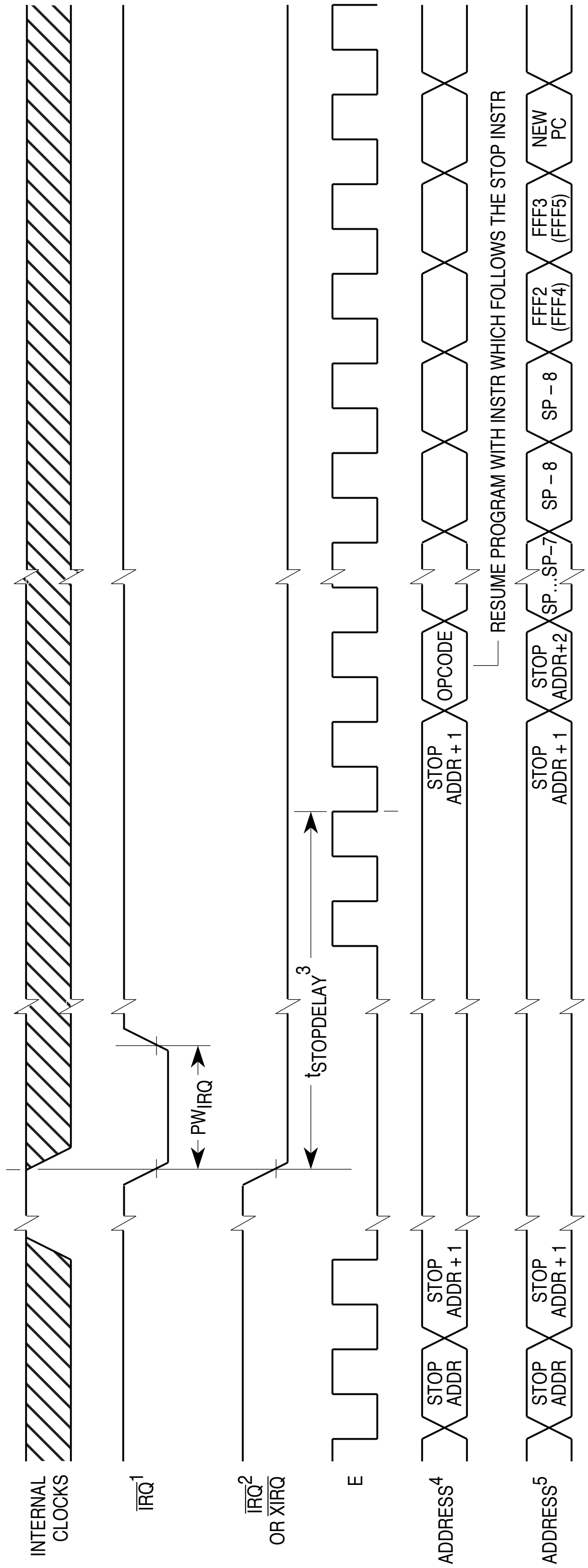


Figure A-3 POR External Reset Timing Diagram



NOTES:

1. Edge Sensitive $\overline{IRQ}$ pin (IRQE bit = 1)
2. Level sensitive $\overline{IRQ}$ pin (IRQE bit = 0)
3. $t_{STOPDELAY}$ = 4064 t_{CYC} if DLY bit = 1 or 4 t_{CYC} if DLY = 0.
4. $\overline{XIRQ}$ with X bit in CCR = 1.
5. $\overline{IRQ}$ or ($\overline{XIRQ}$ with X bit in CCR = 0).

Figure A-4 STOP Recovery Timing Diagram

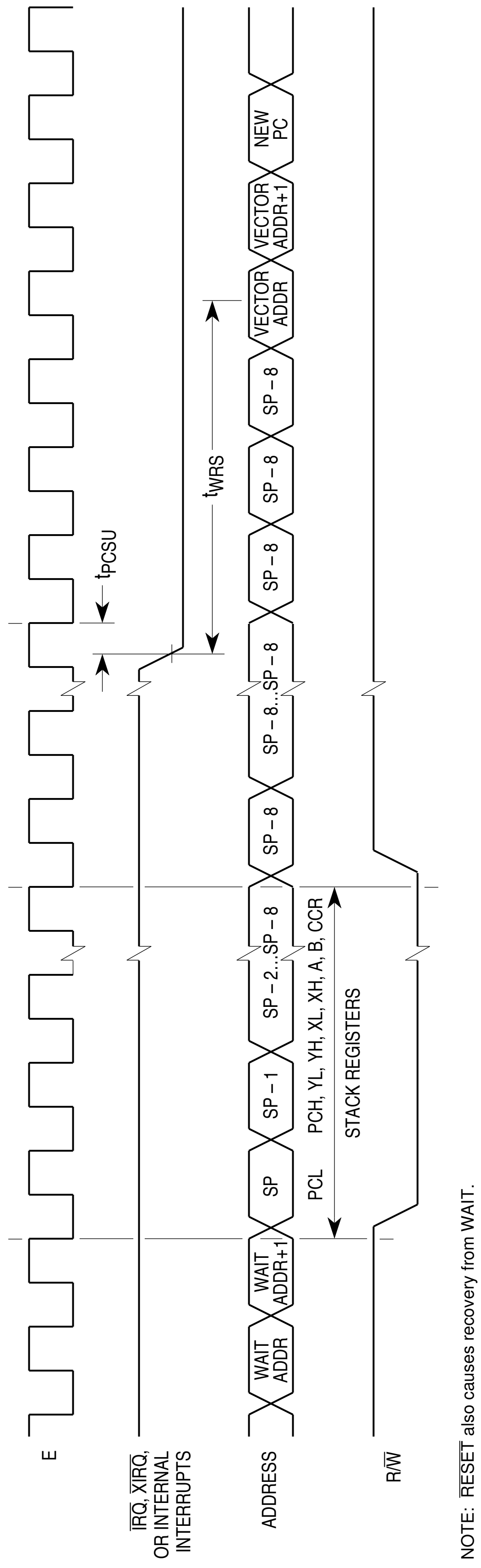
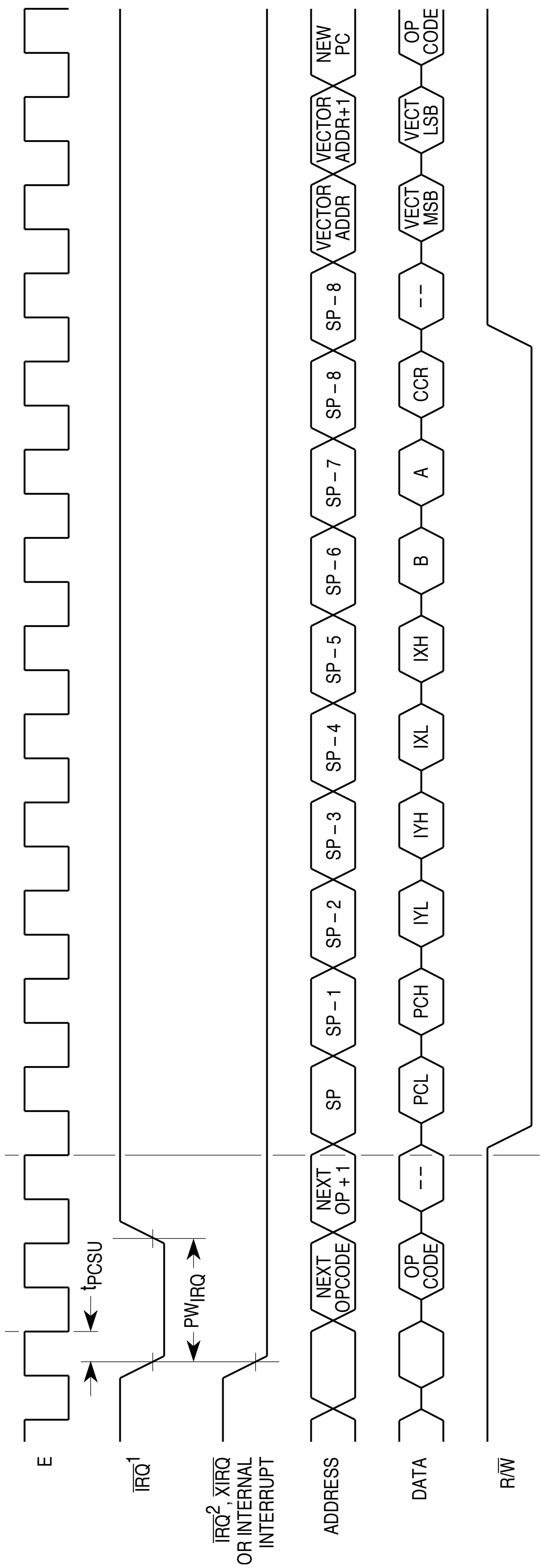


Figure A-5 WAIT Recovery from Interrupt Timing Diagram



NOTES:
1. Edge sensitive $\overline{IRQ}$ pin (IRQE bit = 1).
2. Level sensitive $\overline{IRQ}$ pin (IRQE bit = 0).

Figure A-6 Interrupt Timing Diagram

Table A-5 Peripheral Port Timing

V_{DD} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H

Characteristic	Symbol	2.0 MHz		3.0 MHz		4.0 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation (E-Clock Frequency)	f_o	dc	2.0	dc	3.0	dc	4.0	MHz
E-Clock Period	t_{cyc}	500	—	333	—	250	—	ns
Peripheral Data Setup Time (MCU Read of Ports A, C, D, E, G)	t_{PDSU}	100	—	100	—	100	—	ns
Peripheral Data Hold Time (MCU Read of Ports A, C, D, E, G)	t_{PDH}	50	—	50	—	50	—	ns
Delay Time, Peripheral Data Write	t_{PWD}							ns
(MCU Write to Port A)		—	200	—	200	—	200	
(MCU Write to Ports B, C, D, F, and G t_{PWD} = 1/4 t_{cyc} + 100 ns)		—	225	—	183	—	162	

NOTES:
1. Ports C, D, and G timing is valid for active drive (CWOM, DWOM, and GWOM bits **cleared**).
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.

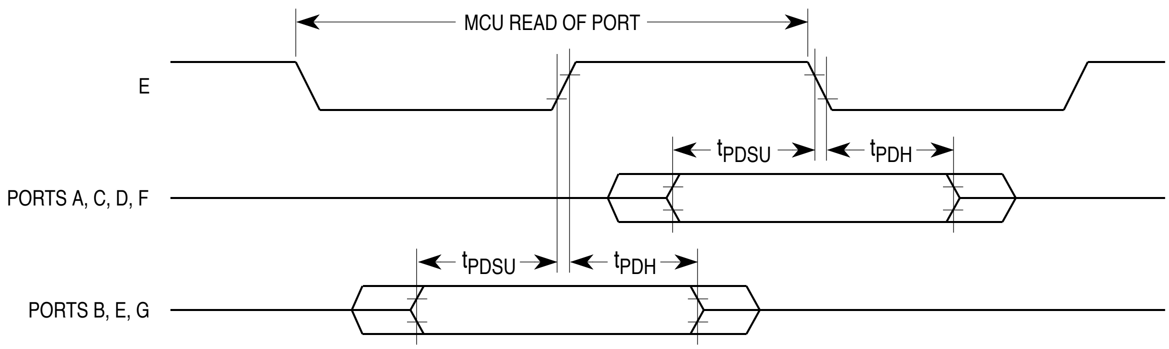


Figure A-7 Port Read Timing Diagram

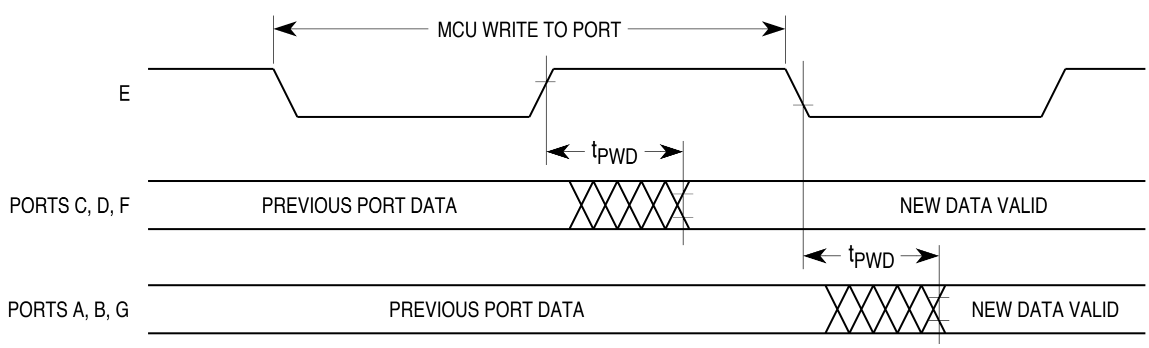


Figure A-8 Port Write Timing Diagram

Table A-6 Analog-To-Digital Converter Characteristics

V_{DD} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, 750 kHz ≤ E ≤ 3.0 MHz, unless otherwise noted

Characteristic	Parameter	Min	Absolute	2.0 MHz	3.0 MHz	4.0 MHz	Unit
				Max	Max	Max	
Resolution	Number of Bits Resolved by A/D Converter	—	8	—	—	—	Bits
Non-Linearity	Maximum Deviation from the Ideal A/D Transfer Characteristics	—	—	± 1	± 1	± 1	LSB
Zero Error	Difference Between the Output of an Ideal and an Actual for Zero Input Voltage	—	—	± 1	± 1	± 1	LSB
Full Scale Error	Difference Between the Output of an Ideal and an Actual A/D for Full-Scale Input Voltage	—	—	± 1	± 1	± 1	LSB
Total Unadjusted Error	Maximum Sum of Non-Linearity, Zero Error, and Full-Scale Error	—	—	± 1/2	± 1 1/2	± 1 1/2	LSB
Quantization Error	Uncertainty Because of Converter Resolution	—	—	± 1/2	± 1/2	± 1/2	LSB
Absolute Accuracy	Difference Between the Actual Input Voltage and the Full-Scale Weighted Equivalent of the Binary Output Code, All Error Sources Included	—	—	± 1	± 2	± 2	LSB
Conversion Range	Analog Input Voltage Range	V_{RL}	—	V_{RH}	V_{RH}	V_{RH}	V
V_{RH}	Maximum Analog Reference Voltage (Note 2)	V_{RL}	—	V_{DD} + 0.1	V_{DD} + 0.1	V_{DD} + 0.1	V
V_{RL}	Minimum Analog Reference Voltage (Note 2)	V_{SS} –0.1	—	V_{RH}	V_{RH}	V_{RH}	V
ΔV_R	Minimum Difference between V_{RH} and V_{RL} (Note 2)	3	—	—	—	—	V
Conversion Time	Total Time to Perform a Single Analog-to-Digital Conversion: E Clock Internal RC Oscillator	 — —	 32 —	 — t_{cyc} + 32	 — t_{cyc} + 32	 — t_{cyc} + 32	 t_{cyc} μs
Monotonicity	Conversion Result Never Decreases with an Increase in Input Voltage and has no Missing Codes		Guaranteed				
Zero Input Reading	Conversion Result when $V_{in} = V_{RL}$	00	—	—	—	—	Hex
Full Scale Reading	Conversion Result when $V_{in} = V_{RH}$	—	—	FF	FF	FF	Hex
Sample Acquisition Time	Analog Input Acquisition Sampling Time: E Clock Internal RC Oscillator	 — —	 12 —	 — 12	 — 12	 — 12	 t_{cyc} μs
Sample/Hold Capacitance	Input Capacitance during Sample PE[7:0]	—	20 (Typ)	—	—	—	pF
Input Leakage	Input Leakage on A/D Pins PE[7:0] V_{RL}, V_{RH}	— —	— —	400 1.0	400 1.0	400 1.0	nA μA

NOTES:

1. For f_{op} < 2 MHz, source impedances should equal approximately 10 kΩ. For f_{op} ≥ 2 MHz, source impedances should equal approximately 5 kΩ – 10 kΩ. Source impedances greater than 10 kΩ affect accuracy adversely because of input leakage.
2. Performance verified down to 2.5 V ΔV_R, but accuracy is tested and guaranteed at ΔV_R = 5 V ± 10%

Table A-7 Expansion Bus Timing

V_{DD} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H

Num	Characteristic	Symbol	2.0 MHz		3.0 MHz		4.0 MHz		Unit
			Min	Max	Min	Max	Min	Max	
	Frequency of Operation (E-Clock Frequency)	f_o	dc	2.0	dc	3.0	dc	4.0	MHz
1	Cycle Time $t_{cyc} = 1/f_o$	t_{cyc}	500	—	333	—	250	—	ns
2	Pulse Width, E Low $PW_{EL} = 1/2\ t_{cyc} - 20$ ns	PW_{EL}	230	—	147	—	105	—	ns
3	Pulse Width, E High $PW_{EH} = 1/2\ t_{cyc} - 25$ ns (Note 2)	PW_{EH}	225	—	142	—	100	—	ns
4A 4B	E Clock Rise Time Fall Time	t_r t_f	— —	20 20	— —	20 18	— —	20 15	ns ns
9	Address Hold Time $t_{AH} = 1/8\ t_{cyc} - 10$ ns	t_{AH}	53	—	32	—	21	—	ns
11	Address Delay Time $t_{AD} = 1/8\ t_{cyc} + 40$ ns	t_{AD}	—	103	—	82	—	71	ns
12	Address Valid Time to E Rise $t_{AV} = PW_{EL} - t_{AD}$	t_{AV}	128	—	65	—	34	—	ns
17	Read Data Setup Time	t_{DSR}	30	—	30	—	20	—	ns
18	Read Data Hold Time	t_{DHR}	0	—	0	—	0	—	ns
19	Write Data Delay Time	t_{DDW}	—	40	—	40	—	40	ns
21	Write Data Hold Time $t_{DHW} = 1/8\ t_{cyc}$	t_{DHW}	63	—	42	—	31	—	ns
29	MPU Address Access Time $t_{ACCA} = t_{cyc} - t_f - t_{DSR} - t_{AD}$ (Note 2)	t_{ACCA}	348	—	203	—	144	—	ns
39	Write Data Setup Time $t_{DSW} = PW_{EH} - t_{DDW}$ (Note 2)	t_{DSW}	185	—	102	—	60	—	ns
50	E Valid Chip Select Delay Time	t_{ECSD}	—	40	—	40	—	40	ns
51	E Valid Chip Select Access Time $t_{ECSA} = PW_{EH} - t_{ECSD} - t_{DSR}$ (Note 2)	t_{ECSA}	155	—	72	—	40	—	ns
52	Chip Select Hold Time	t_{CH}	0	20	0	20	0	20	ns
54	Address Valid Chip Select Delay Time $t_{ACSD} = 1/4\ t_{cyc} + 40$ ns	t_{ACSD}	—	165	—	123	—	103	ns
55	Address Valid Chip Select Access Time $t_{ACSA} = t_{cyc} - t_f - t_{DSR} - t_{ACSD}$ (Note 2)	t_{ACSA}	285	—	162	—	113	—	ns
56	Address Valid to Chip Select Time	t_{AVCS}	10	—	10	—	10	—	ns
57	Address Valid to Data Three-State Time	t_{AVDZ}	—	10	—	10	—	10	ns

NOTES:
1. Input clocks with duty cycles other than 50% affect bus performance.
2. Indicates a parameter affected by clock stretching. Add $n(t_{cyc})$ to parameter value, where:
   n = 1, 2, or 3 depending on values written to CSSTRH register.
3. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.

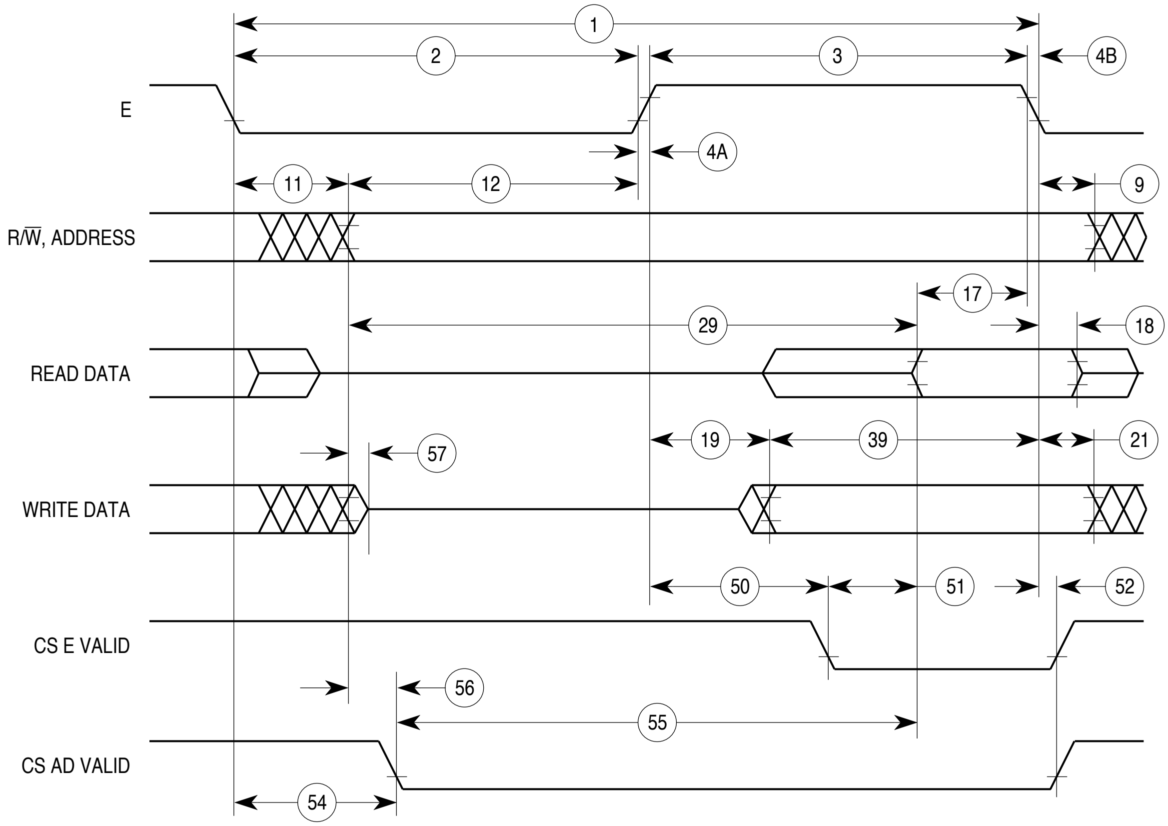


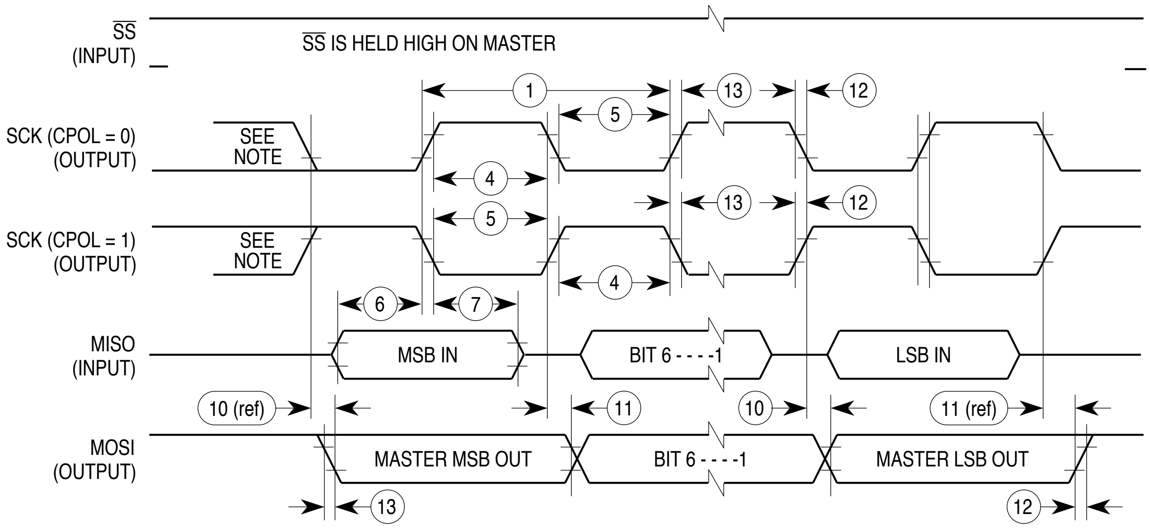
Figure A-9 Expansion Bus Timing

Table A-8 Serial Peripheral Interface Timing

V_{DD} = 5.0 Vdc ± 5%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H

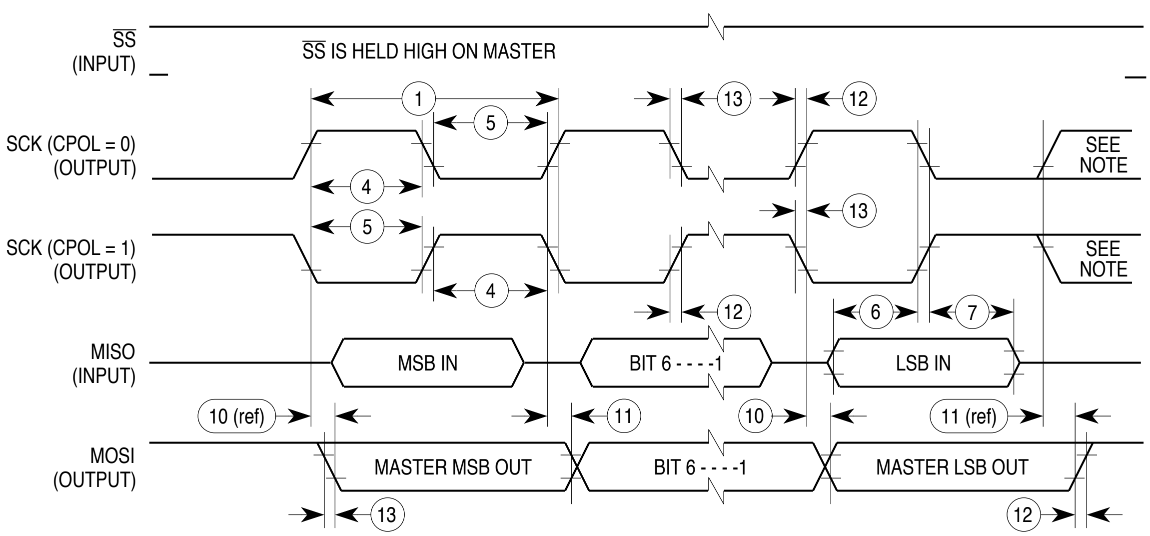
Num	Characteristic	Symbol	2.0 MHz		3.0 MHz		4.0 MHz		Unit
			Min	Max	Min	Max	Min	Max	
	Operating Frequency Master Slave	$f_{op(m)}$ $f_{op(s)}$	dc dc	1.0 2.0	dc dc	1.5 3.0	dc dc	2.0 4.0	MHz MHz
1	Cycle Time Master Slave	$t_{cyc(m)}$ $t_{cyc(s)}$	2.0 500	— —	2.0 333	— —	2.0 250	— —	t_{cyc} ns
2	Enable Lead Time Master (Note 2) Slave	$t_{lead(m)}$ $t_{lead(s)}$	— 250	— —	— 240	— —	— 200	— —	ns ns
3	Enable Lag Time Master (Note 2) Slave	$t_{lag(m)}$ $t_{lag(s)}$	— 250	— —	— 240	— —	— 200	— —	ns ns
4	Clock (SCK) High Time Master Slave	$t_{w(SCKH)m}$ $t_{w(SCKH)s}$	340 190	— —	227 127	— —	130 85	— —	ns ns
5	Clock (SCK) Low Time Master Slave	$t_{w(SCKL)m}$ $t_{w(SCKL)s}$	340 190	— —	227 127	— —	130 85	— —	ns ns
6	Data Setup Time (Inputs) Master Slave	$t_{su(m)}$ $t_{su(s)}$	100 100	— —	100 100	— —	100 100	— —	ns ns
7	Data Hold Time (Inputs) Master Slave	$t_{h(m)}$ $t_{h(s)}$	100 100	— —	100 100	— —	100 100	— —	ns ns
8	Access Time (Time to Data Active from High-Impedance State) Slave	t_a	0	120	0	120	0	120	ns
9	Disable Time (Hold Time to High-Impedance State) Slave	t_{dis}	—	240	—	167	—	125	ns
10	Data Valid (After Enable Edge) (Note 3)	$t_{v(s)}$	—	240	—	167	—	125	ns
11	Data Hold Time (Outputs) (After Enable Edge)	t_{ho}	0	—	0	—	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and $\overline{SS}$)	t_{rm} t_{rs}	— —	100 2.0	— —	100 2.0	— —	100 2.0	ns µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and $\overline{SS}$)	t_{fm} t_{fs}	— —	100 2.0	— —	100 2.0	— —	100 2.0	ns µs

NOTES:
1. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. Signal production depends on software.
3. Assumes 200 pF load on all SPI pins.



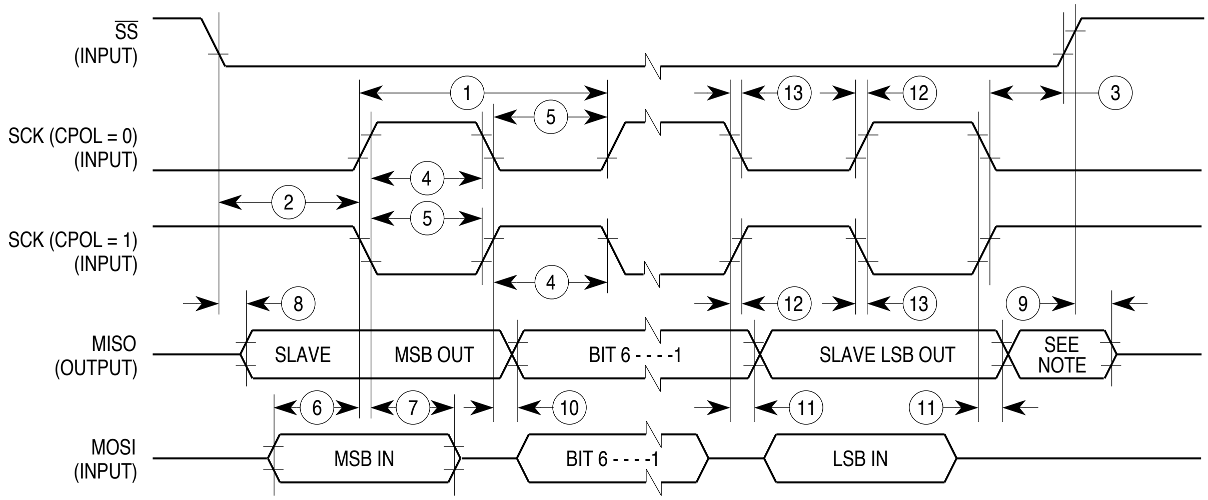
NOTE: This first clock edge is generated internally but is not seen at the SCK pin.

Figure A-10 SPI Master Timing (CPHA = 0)



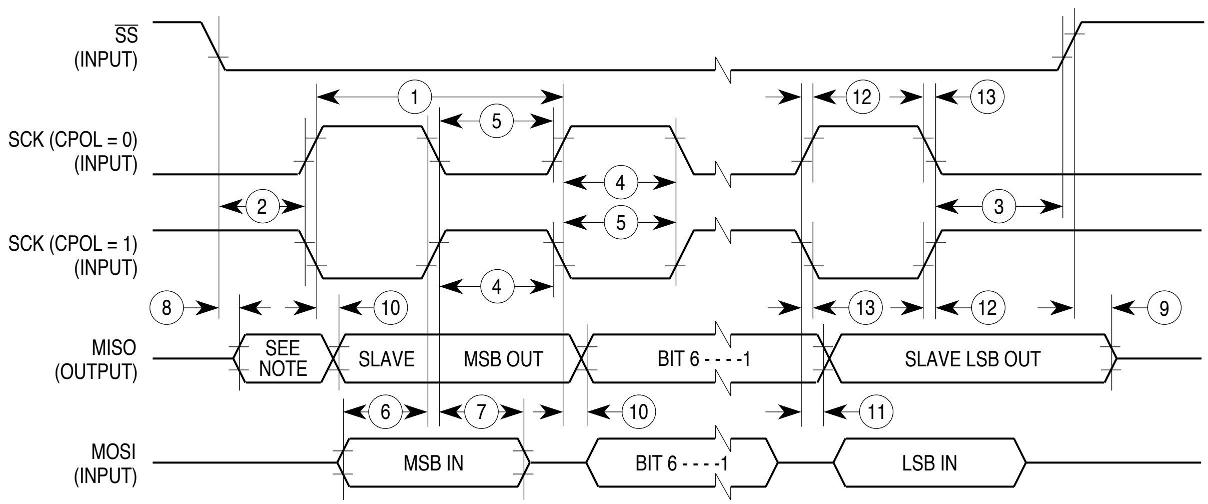
NOTE: This last clock edge is generated internally but is not seen at the SCK pin.

Figure A-11 SPI Master Timing (CPHA = 1)



NOTE: Not defined but normally MSB of character just received.

Figure A-12 SPI Slave Timing (CPHA = 0)



NOTE: Not defined but normally LSB of character previously transmitted.

Figure A-13 SPI Slave Timing (CPHA = 1)

Table A-9 EEPROM Characteristics

V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H

Characteristic		Temperature Range		Unit
	0 to 70, – 40 to 85	**–40 to 105**	**–40 to 125**	**° C**
Programming Time (Note 1) <1.0 MHz, RCO Enabled 1.0 to 2.0 MHz, RCO Disabled ≥ 2.0 MHz (or Anytime RCO Enabled)	10 20 10	15 Must use RCO 15	20 Must use RCO 20	ms
Erase Time (Note 1) Byte, Row and Bulk	10	10	10	ms
Write/Erase Endurance (Note 2)	10,000	10,000	10,000	Cycles
Data Retention (Note 2)	10	10	10	Years

NOTES:
1. The RC oscillator (RCO) must be enabled (by setting the CSEL bit in the OPTION register) for EEPROM programming and erasure when the E-clock frequency is below 1.0 MHz.
2. Refer to Reliability Monitor Report (current quarterly issue) for current failure rate information.

APPENDIX BMECHANICAL DATA AND ORDERING INFORMATION

B.1 Pin Assignments

The MC68HC11F1 is available in the 80-pin plastic low profile quad flat pack (LQFP) or the 68-pin plastic leaded chip carrier (PLCC). Refer to **Table B-1** for ordering information.

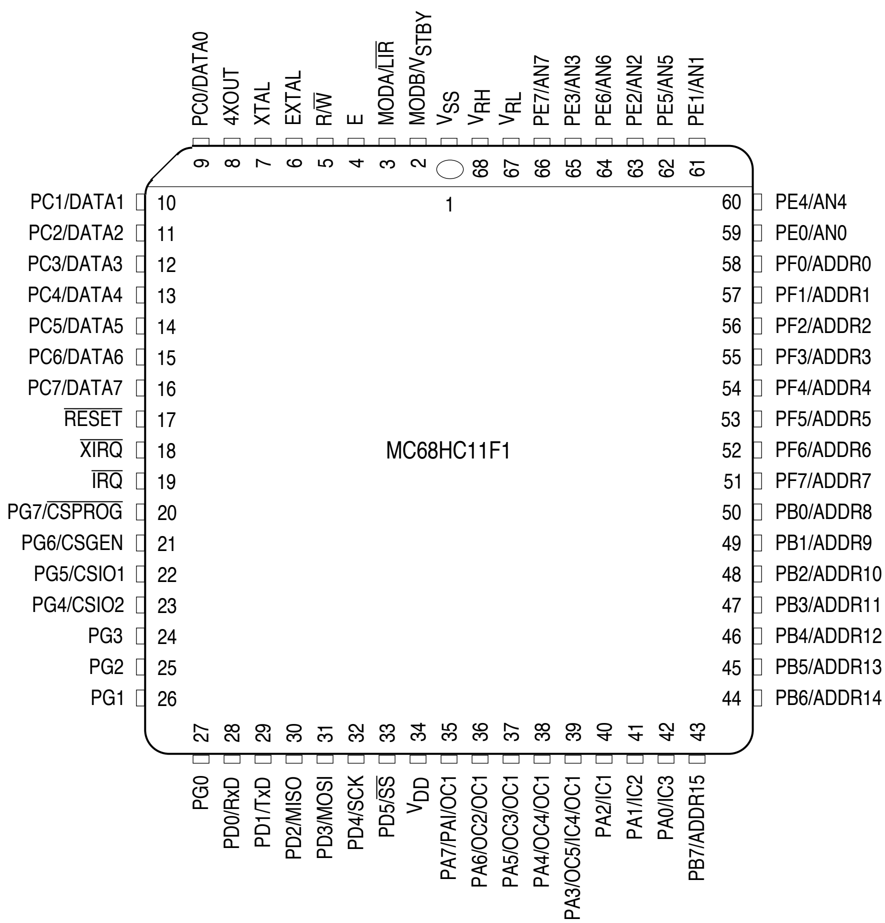


Figure B-1 MC68HC11F1 68-Pin PLCC

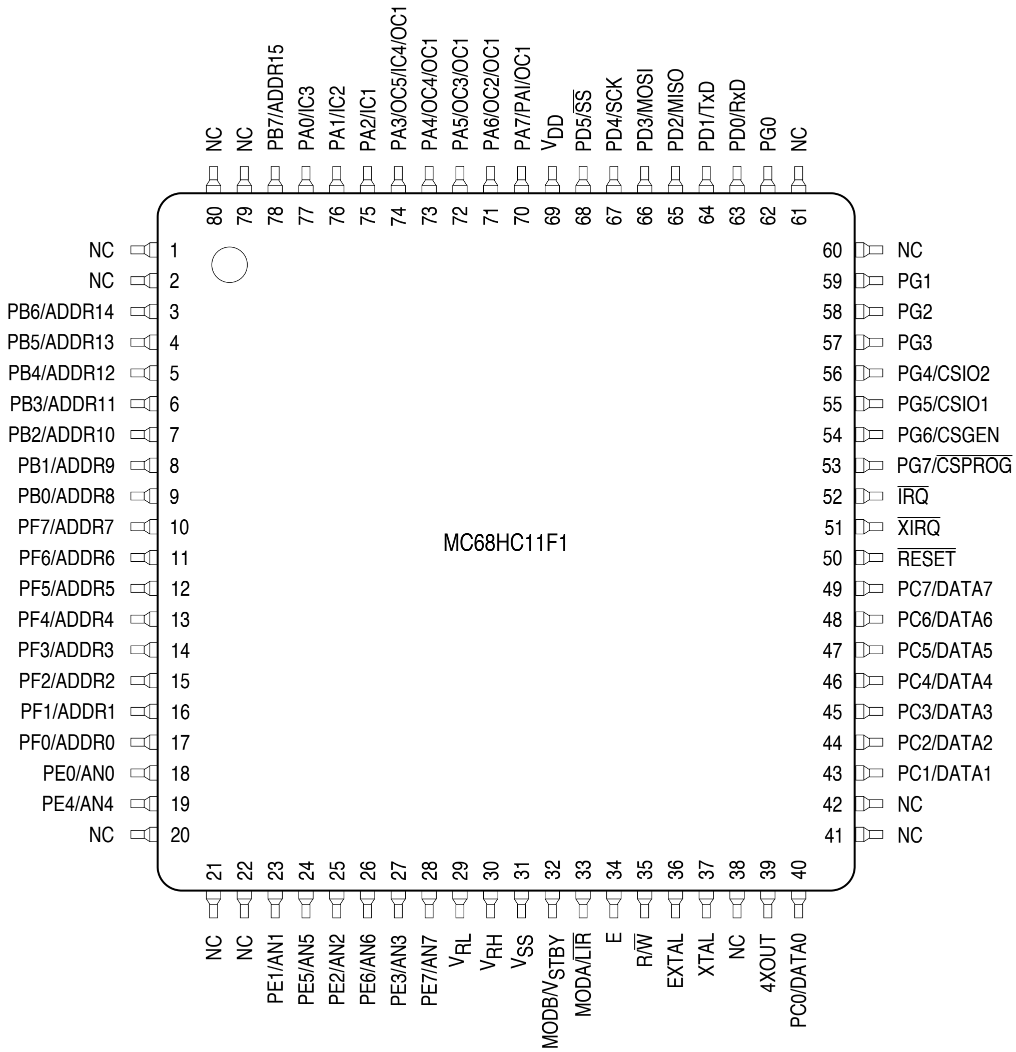


Figure B-2 MC68HC11F1 80-Pin Quad Flat Pack

B.2 Package Dimensions

For case outlines please visit our website at http://design-net.sps.mot.com.

B.3 Ordering Information

Use the information in **Table B-1** to specify the appropriate device when placing an order.

Table B-1 Device Ordering Information

Description	Package	Temperature	Frequency	MC Order Number
NO ROM, 512 Bytes EEPROM, 1024 Bytes RAM	80-Pin LQFP (14 mm X 14 mm, 1.4 mm thick)	– 40° to + 85° C	2 MHz	MC68HC11F1CPU2
			3 MHz	MC68HC11F1CPU3
			4 MHz	MC68HC11F1CPU4
		– 40° to + 105° C	2 MHz	MC68HC11F1VPU2
			3 MHz	MC68HC11F1VPU3
			4 MHz	MC68HC11F1VPU4
		– 40° to + 125° C	2 MHz	MC68HC11F1MPU2
			3 MHz	MC68HC11F1MPU3
	68-Pin PLCC	– 40° to + 85° C	2 MHz	MC68HC11F1CFN2
			3 MHz	MC68HC11F1CFN3
			4 MHz	MC68HC11F1CFN4
		– 40° to + 105° C	2 MHz	MC68HC11F1VFN2
			3 MHz	MC68HC11F1VFN3
			4 MHz	MC68HC11F1VFN4
		– 40° to + 125° C	2 MHz	MC68HC11F1MFN2
			3 MHz	MC68HC11F1MFN3

APPENDIX CDEVELOPMENT SUPPORT

C.1 MC68HC11F1 Development Tools

The following table and text provide a reference to development tools for the MC68HC11F1 microcontrollers. For more complete information refer to the appropriate manual for each system.

Table C-1 MC68HC11F1 Development Tools

Device	Evaluation Systems	Modular Development Systems
MC68HC11F1	M68HC11F1EVS	MMDS11*

* For MC68HC11F1 support, the MMDS11 must be used with an MC68HC11F1 emulator module.

C.2 MC68HC11EVS — Evaluation System

The EVS is an economical tool for designing, debugging, and evaluating target systems based on the MC68HC11F1 MCU. The two printed circuit boards that comprise the EVS are the MC68HC11F1EM emulator module (EM) and the M68HC11PFB platform board (PFB).

- Monitor/debugger firmware
- One-line assembler/disassembler
- Host computer download capability
- Dual memory maps:
  - 64 Kbyte monitor map that includes 16 Kbytes of monitor EPROM
  - MC68HC11F1 user map that includes 64 Kbytes of emulation RAM
- OTPROM, EPROM, and EEPROM MCU programmer
- MCU extension I/O port for single-chip, expanded, and special-test operation modes
- RS-232C terminal and host I/O ports
- Logic analyzer connector

C.3 M68MMDS11 — Modular Development System for M68HC11 Devices

The M68MMDS11 Motorola Modular Development System (MMDS11) is a tool for developing embedded systems based on M68HC11 MCUs. The MMDS11 is an emulator system that provides an on-screen bus state analyzer and real-time memory monitoring windows. An integrated design environment includes an editor, an assembler, the user interface, and source-level debug capability. These features significantly reduce the time necessary to develop and debug an embedded MCU system. The compact unit requires minimum space.

- Real-time, non-intrusive, in-circuit emulation
- Assembly-language source-level debugging

- Built-in real-time bus state analyzer:
  - 8 Kbyte x 64 real-time trace buffer
  - Four hardware triggers control real-time bus analysis, provide breakpoints
  - Nine triggering modes
  - Display of real-time trace data as raw data, disassembled instructions, raw data and disassembled instructions, or assembly-language source code
  - As many as 8190 pre- or post-trigger points
  - Trace buffer can be filled while single-stepping through user software
  - 16-bit or 24-bit time tag
  - Programmable time tag clock source
  - 16 general-purpose logic clips, four can trigger bus state analyzer sequencer
- 64 Kbytes of emulation memory
- 32-byte block of real-time memory, can be mapped within a 1 Kbyte window anywhere in the 64 Kbyte M68HC11 memory map.
- Six software-selectable oscillator clock sources: five internally generated frequencies or an external frequency on a logic clip
- 64 possible hardware instruction breakpoints over the 64 Kbyte M68HC11 memory map or a one megabyte bank selected memory map.
- Four data breakpoints (hardware breakpoints). A data breakpoint can be qualified by an address, an address range, data, or clips.
- SCRIPT command for automatic execution of MMDS11 command sequences
- Command and response logging
- A DOS personality file for each EM that the MMDS supports. Each personality file provides a foreground memory-map description and a chip information file.
- CHIPINFO command provides memory-map, vectors, registers, and pin-out information contained in the personality file
- Latch-up resistant design makes power-up sequencing unimportant.
- RS-232 operation speeds as high as 57.6 Kbaud
- On-screen, context-sensitive help
- Mouse or keyboard control of software
- Built-in power supply
- Compact size: 15.38 inches (390.6 mm) deep, 10.19 inches (258.83 mm) wide, and 2.75 inches (69.85 mm) high. Station module weighs 6.0 pounds (2.72 kg).