MC68HC11E/D
REV 1

HC11

M68HC11 E SERIES

TECHNICAL
DATA

MOTOROLA

M68HC11 E SERIES

HCMOS MICROCONTROLLER UNIT

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

TABLE OF CONTENTS

Paragraph **Title** **Page**

SECTION 1
INTRODUCTION

SECTION 2
PIN DESCRIPTIONS

SECTION 3
CENTRAL PROCESSING UNIT

TABLE OF CONTENTS
(Continued)

SECTION 4
OPERATING MODES AND ON–CHIP MEMORY

SECTION 5
RESETS AND INTERRUPTS

TABLE OF CONTENTS
(Continued)

SECTION 6
PARALLEL INPUT/OUTPUT

SECTION 7
SERIAL COMMUNICATIONS INTERFACE

TABLE OF CONTENTS
(Continued)

Paragraph Title Page

SECTION 8
SERIAL PERIPHERAL INTERFACE

SECTION 9
TIMING SYSTEM

TABLE OF CONTENTS
(Continued)

SECTION 10
ANALOG-TO-DIGITAL CONVERTER

APPENDIX A
ELECTRICAL CHARACTERISTICS

APPENDIX B
MECHANICAL DATA AND ORDERING INFORMATION

APPENDIX C
DEVELOPMENT SUPPORT

SUMMARY OF CHANGES

LIST OF ILLUSTRATIONS

Figure **Title** **Page**

LIST OF ILLUSTRATIONS
(Continued)

Figure Title Page

LIST OF TABLES

LIST OF TABLES (Continued)

SECTION 1
INTRODUCTION

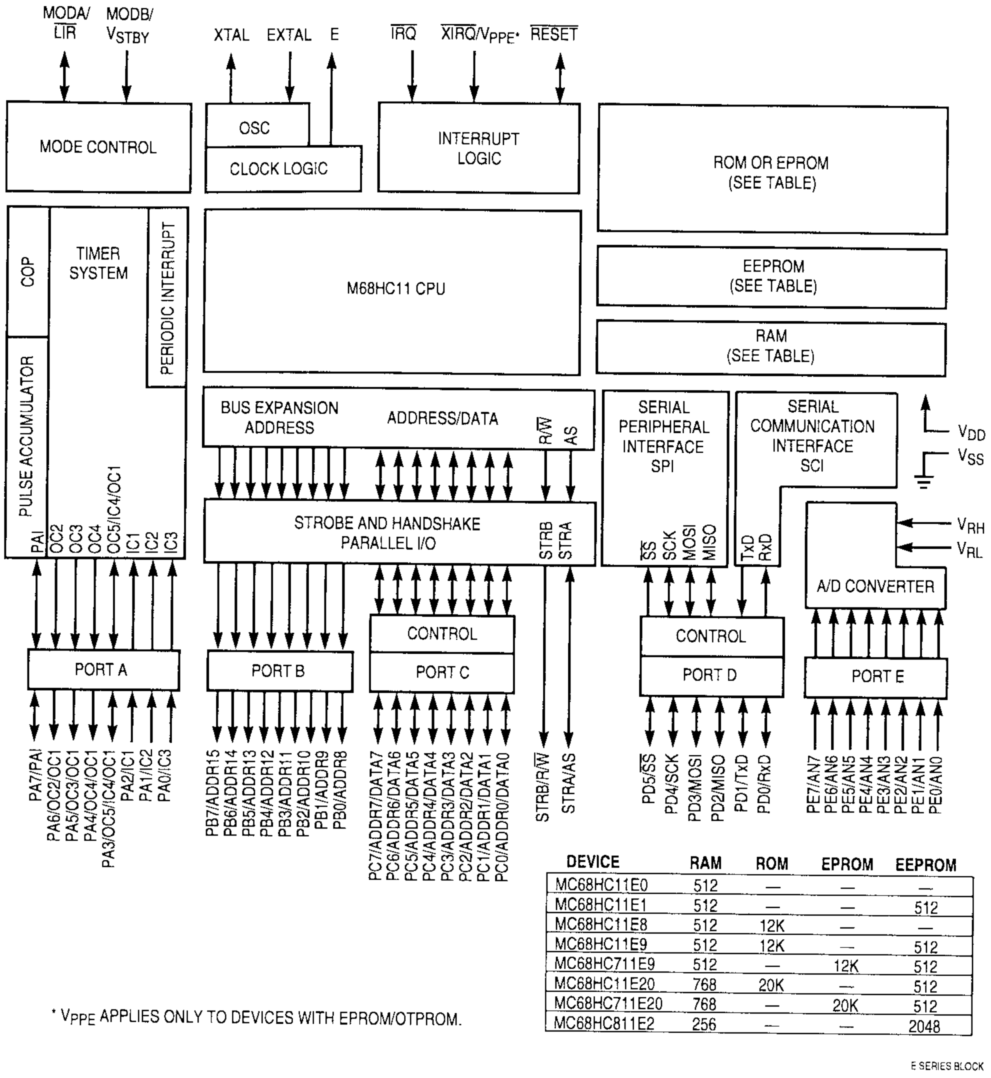
This document contains a detailed description of the M68HC11 E series of 8-bit microcontroller units (MCUs). These MCUs all combine the M68HC11 CPU with high-performance, on-chip peripherals. The E series is comprised of many devices with various configurations of RAM, ROM or EPROM, and EEPROM. Several low-voltage devices are also available. With the exception of a few minor differences, the operation of all E-series MCUs is identical. A fully static design and high-density complementary metal-oxide semiconductor (HCMOS) fabrication process allow E-series devices to operate at frequencies from 3 MHz to dc, with very low power consumption.

1.1 Features

- M68HC11 CPU
- Power Saving STOP and WAIT Modes
- Low-Voltage Devices Available (3.0 – 5.5 Vdc or 2.7 – 5.5 Vdc)
- 0, 256, 512, or 768 Bytes of On-Chip RAM, Data Retained During Standby
- 0, 12, or 20 Kbytes of On-Chip ROM or EPROM
- 0, 512, or 2048 Bytes of On-Chip EEPROM with Block Protect for Security
  - — 2048 bytes of EEPROM with selectable base address in MC68HC811E2
- Asynchronous Nonreturn to Zero (NRZ) Serial Communications Interface (SCI)
  - — Additional Baud Rates Available on MC68HC(7)11E20
- Synchronous Serial Peripheral Interface (SPI)
- 8-Channel 8-Bit Analog-to-Digital (A/D) Converter
- 16-Bit Timer System
  - — 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 System
- 38 General-Purpose Input/Output (I/O) Pins
  - — 16 Bidirectional I/O Pins
  - — 11 Input-Only Pins
  - — 11 Output-Only Pins
- Several Packaging Options
  - — 52-Pin Plastic Leaded Chip Carrier (PLCC)
  - — 52-Pin Windowed Ceramic Leaded Chip Carrier (CLCC)
  - — 52-Pin Plastic Thin Quad Flat Pack, 10 mm X 10 mm (TQFP)
  - — 64-Pin Plastic Quad Flat Pack (QFP)
  - — 48-Pin Plastic Dual In-Line Package (DIP), MC68HC811E2 only
  - — 56-Pin Plastic Dual In-Line Package, .070" Lead Spacing (SDIP)

1.2 Structure

Below is the functional diagram of the E-series MCUs. Differences among devices are noted in the table.



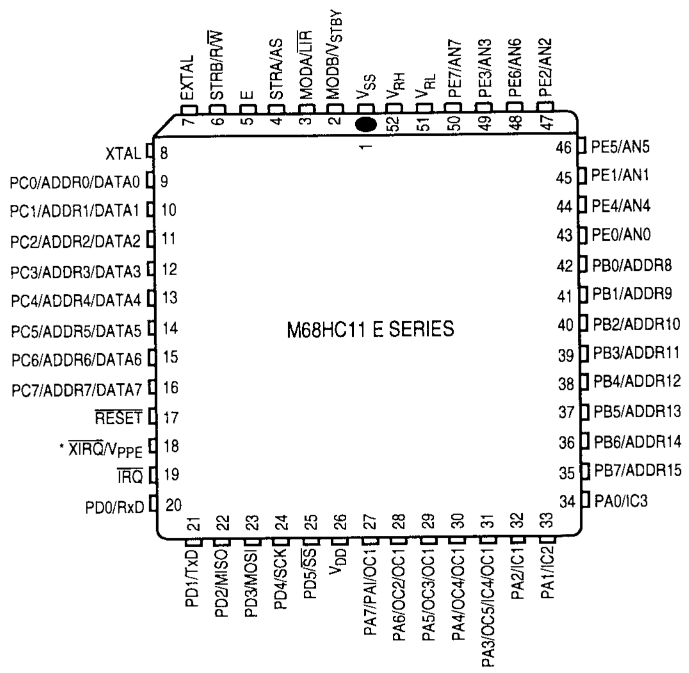
DEVICE	RAM	ROM	EPROM	EEPROM
MC68HC11E0	512	—	—	—
MC68HC11E1	512	—	—	512
MC68HC11E8	512	12K	—	—
MC68HC11E9	512	12K	—	512
MC68HC711E9	512	—	12K	512
MC68HC11E20	768	20K	—	512
MC68HC711E20	768	—	20K	512
MC68HC811E2	256	—	—	2048

* V_{PPE} APPLIES ONLY TO DEVICES WITH EPROM/OTPROM.

Figure 1–1 M68HC11 E-Series Block Diagram

SECTION 2
PIN DESCRIPTIONS

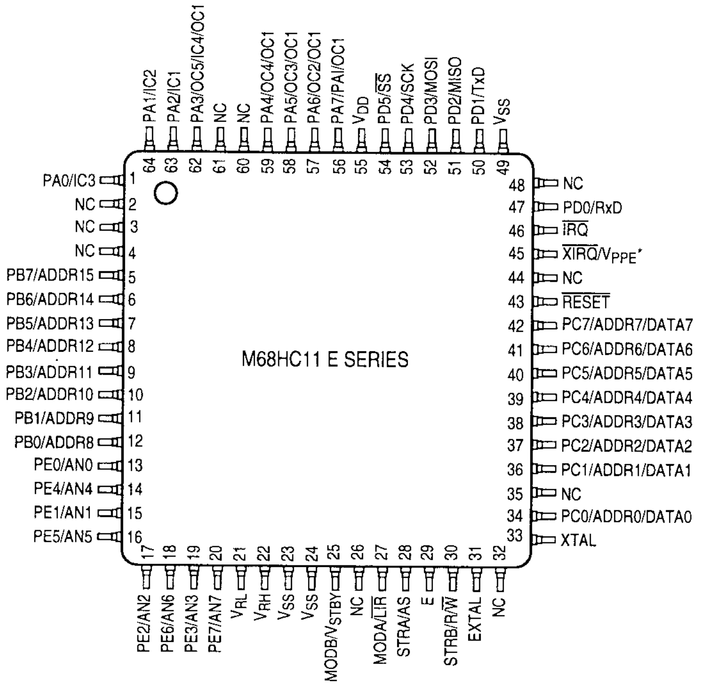
M68HC11 E-series MCUs are available packaged in 52-pin PLCC, 52-pin windowed CLCC, 64-pin QFP, 52-pin thin QFP, 56-pin SDIP, and 48-pin DIP (MC68HC811E2 only). Most pins on these MCUs serve two or more functions, as described in the following paragraphs. Refer to **Figure 2–1**, **Figure 2–2**, **Figure 2–3**, **Figure 2–4**, and **Figure 2–5**, which show the M68HC11 E-series pin assignments for the PLCC/CLCC, QFP, and DIP packages.



* V_{PPE} APPLIES ONLY TO DEVICES WITH EPROM/OTPROM.

E SERIES 52-PIN PLCC

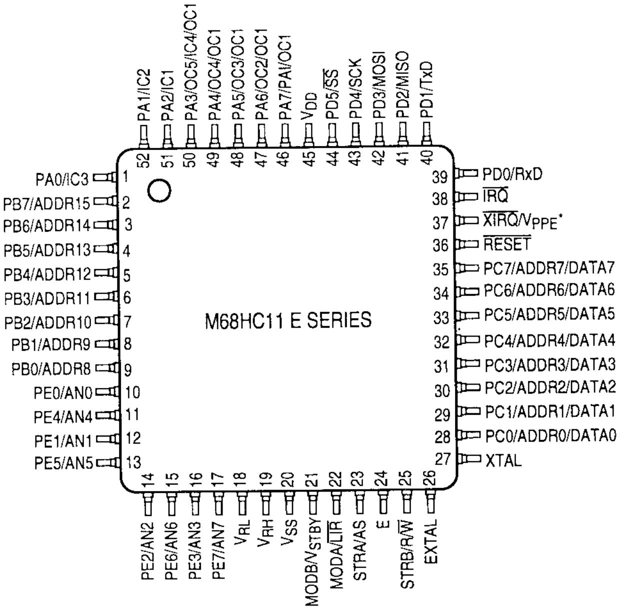
Figure 2–1 Pin Assignments for 52-Pin PLCC and CLCC



* V_{PPE} APPLIES ONLY TO DEVICES WITH EPROM/OTPROM.

E SERIES 64-PIN QFP

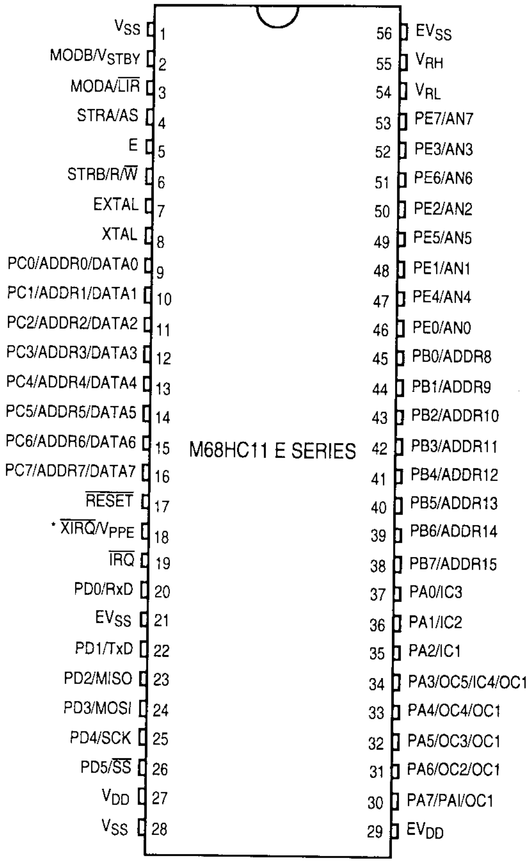
Figure 2–2 Pin Assignments for 64-Pin QFP



* V_{PPE} APPLIES ONLY TO DEVICES WITH EPROM/OTPROM.

E SERIES 52-PIN QFP

Figure 2–3 Pin Assignments for 52-Pin Thin QFP



* V_{PPE} APPLIES ONLY TO DEVICES WITH EPROM/OTPROM.

E SERIES 56-PIN DIP

Figure 2–4 Pin Assignments for 56-Pin SDIP

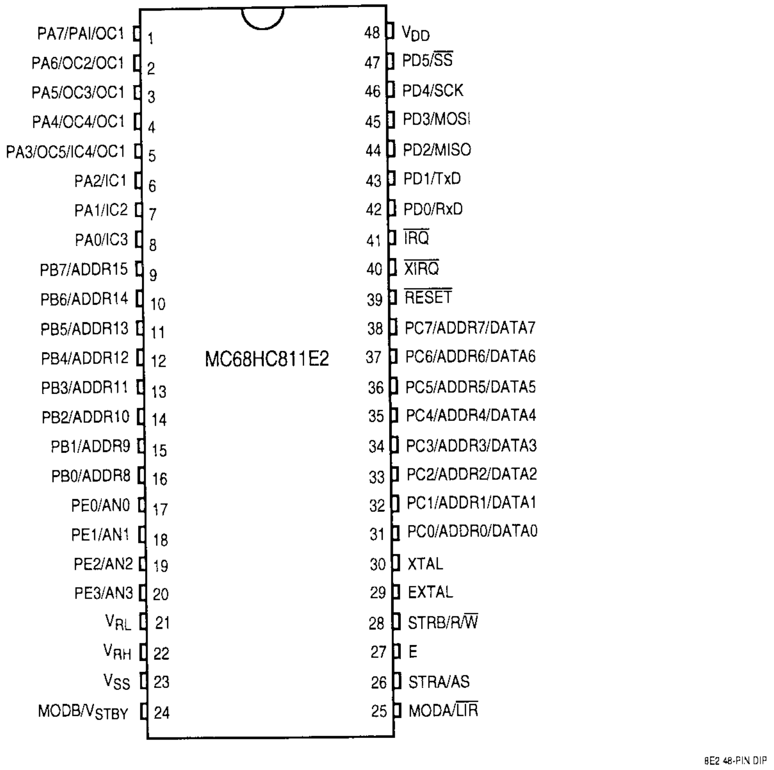


Figure 2–5 Pin Assignments for 48-Pin DIP (MC68HC811E2)

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, V_{SS} is ground. The MCU operates from a single 5-volt (nominal) power supply. Low-voltage devices in the E series operate at either 3.0 – 5.5 volts or 2.7 – 5.5 volts. 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 $\overline{\text{RESET}}$

A bidirectional control signal, $\overline{\text{RESET}}$, acts as an input to initialize the MCU to a known startup 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. Do not 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.

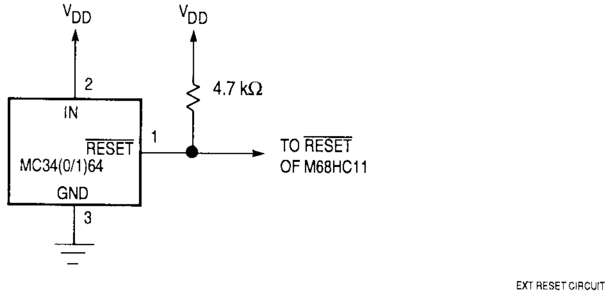


Figure 2–6 External Reset Circuit

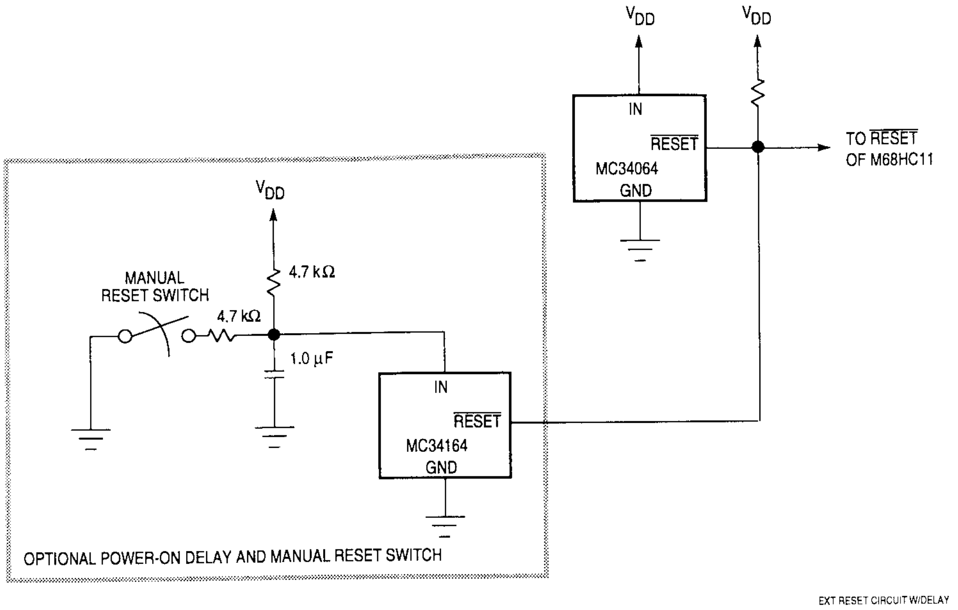


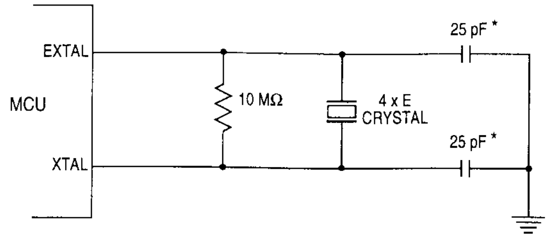
Figure 2–8 External Reset Circuit with Delay

2.3 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. The frequency applied to these pins is four times higher than the desired E-clock rate.

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

In all cases, use caution around the oscillator pins. Load capacitances shown in the oscillator circuits include all stray layout capacitances. Refer to **Figure 2–9**, **Figure 2–10**, and **Figure 2–11**.



* THIS VALUE INCLUDES ALL STRAY CAPACITANCES.

Figure 2–9 Common Crystal Connections

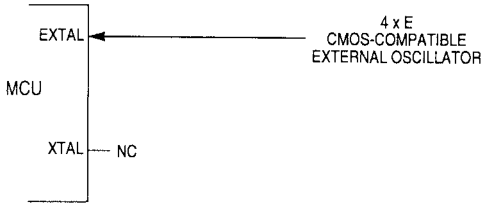
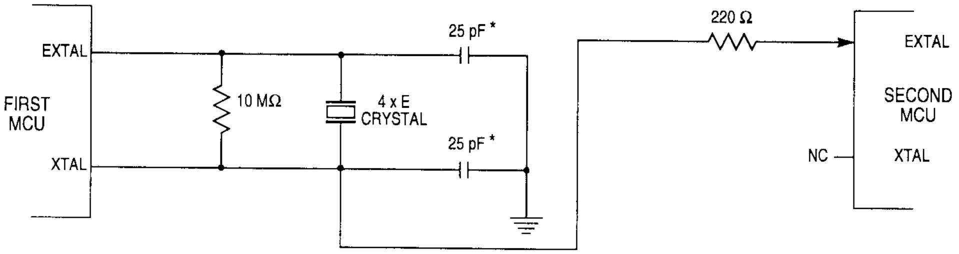


Figure 2–10 External Oscillator Connections



* THIS VALUE INCLUDES ALL STRAY CAPACITANCES.

Figure 2–11 One Crystal Driving Two MCUs

2.4 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 XTAL and EXTAL pins. 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. To reduce RFI emissions, the E-clock output of most E-series devices can be disabled while operating in single-chip modes. The E clock signal is always enabled on the MC68HC811E2.

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

The $\overline{\text{IRQ}}$ input provides a means of applying asynchronous interrupt requests to the MCU. Either negative edge-sensitive triggering or level-sensitive triggering is program selectable (OPTION register). $\overline{\text{IRQ}}$ is always configured to level-sensitive triggering at reset. When using $\overline{\text{IRQ}}$ in a level-sensitive wired-OR configuration, connect an external pull-up resistor, typically 4.7 kΩ, to V_{DD}.

2.6 Nonmaskable Interrupt ($\overline{\text{XIRQ}}$/V_{PPE})

The $\overline{\text{XIRQ}}$ input provides a means of requesting a nonmaskable 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 MCU is cleared (normally upon return from an interrupt). Refer to **SECTION 5 RESETS AND INTERRUPTS**.

V_{PPE} is the input for the 12 volt nominal programming voltage required for EPROM/OTPROM programming. On devices without EPROM/OTPROM this pin is only $\overline{\text{XIRQ}}$ input.

2.7 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 load instruction register ($\overline{\text{LIR}}$) pin provides an open-drain output to indicate that execution of an instruction has begun. A series of E-clock cycles occurs during execution of each instruction. The $\overline{\text{LIR}}$ signal goes 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. When the voltage on this pin is more than one MOS threshold (about 0.7 volts) above the V_{DD} voltage, the internal RAM and part of the reset logic are powered from this signal rather than the V_{DD} input. 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.

2.8 V_{RL} and V_{RH}

These two inputs provide the reference voltages for the analog-to-digital converter circuitry. V_{RL} is the low reference, typically 0 Vdc. V_{RH} is the high reference. For proper A/D converter operation, V_{RH} should be at least 3 Vdc greater than V_{RL}, and both V_{RL} and V_{RH} should be between V_{SS} and V_{DD}.

2.9 STRA/AS

This pin performs either of two separate functions, depending on the operating mode. In single-chip mode, STRA performs an input handshake (strobe input) function. In the expanded multiplexed mode, AS provides an address strobe function. AS can be used to demultiplex the address and data signals at port C. Refer to **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY**.

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

The strobe B and read/write pin acts as either an output strobe, or as a data bus direction indicator, depending on the operating mode. In single-chip operating mode, STRB acts as a programmable strobe for handshake with other parallel devices. Refer to **SECTION 6 PARALLEL INPUT/OUTPUT** for further information.

In expanded multiplexed operating mode, R/$\overline{\text{W}}$ is used to indicate the direction of transfers on the external data bus. A low on the R/$\overline{\text{W}}$ pin indicates data is being written to the external data bus. A high on this pin indicates that a read cycle is in progress. R/$\overline{\text{W}}$ stays low during consecutive data bus write cycles, such as a double-byte store. It is possible for data to be driven out port C, if internal read visibility is enabled and an internal address is read, even though R/$\overline{\text{W}}$ is in a high-impedance state. Refer to **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY** for more information about IRVNE.

2.11 Port Signals

Port pins have different functions in different operating modes. Pin functions for ports A, D, and E are independent of operating modes. Ports B and C, however, are affected by operating mode. Port B provides eight general-purpose output signals in single-chip operating modes. When the microcontroller is in expanded multiplexed operating mode, port B pins are the eight high-order address lines. Port C provides eight general-purpose input/output signals when the MCU is in the single-chip operating mode. When the microcontroller is in the expanded multiplexed operating mode, port C pins are a multiplexed address/data bus. Refer to **Table 2–1** for details about functions of the 40 port signals within different operating modes. Terminate unused inputs and I/O pins configured as inputs high or low.

Table 2–1 Port Signal Functions

Port/Bit	Single-Chip and Bootstrap Modes	Expanded and Test Modes
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	
PB0	PB0	ADDR8
PB1	PB1	ADDR9
PB2	PB2	ADDR10
PB3	PB3	ADDR11
PB4	PB4	ADDR12
PB5	PB5	ADDR13
PB6	PB6	ADDR14
PB7	PB7	ADDR15
PC0	PC0	ADDR0/DATA0
PC1	PC1	ADDR1/DATA1
PC2	PC2	ADDR2/DATA2
PC3	PC3	ADDR3/DATA3
PC4	PC4	ADDR4/DATA4
PC5	PC5	ADDR5/DATA5
PC6	PC6	ADDR6/DATA6
PC7	PC7	ADDR7/DATA7
PD0	PD0/RxD	
PD1	PD1/TxD	
PD2	PD2/MISO	
PD3	PD3/MOSI	
PD4	PD4/SCK	
PD5	PD5/$\overline{\text{SS}}$	
—	STRA	AS
—	STRB	R/$\overline{\text{W}}$
PE0	PE0/AN0	
PE1	PE1/AN1	
PE2	PE3/AN2	
PE3	PE3/AN3	
PE4	PE4/AN4	
PE5	PE5/AN5	
PE6	PE6/AN6	
PE7	PE7/AN7	

2.11.1 Port A

In all operating modes, port A can be configured for three timer input capture (IC) functions and four timer output compare (OC) functions. An additional pin can be configured as either the fourth IC or the fifth OC. Any port A pin that is not currently being used for a timer function can be used as either a general-purpose input or output line. Only port A pins PA7 and PA3 have an associated data direction control bit that allows the pin to be selectively configured as input or output. Bits DDRA7 and DDRA3 located in PACTL register control data direction for PA7 and PA3, respectively. All other port A pins are fixed as either input or output.

PA7 can function as general-purpose I/O or as timer output compare for OC1. PA7 is also the input to the pulse accumulator, even while functioning as a general-purpose I/O or an OC1 output.

PA[6:4] serve as either general-purpose outputs, timer input captures or timer output compare 2–4. In addition, PA[6:4] can be controlled by OC1.

PA3 can be a general-purpose I/O pin or a timer IC/OC pin. Timer functions associated with this pin include OC1 and IC4/OC5. IC4/OC5 is software selectable as either a fourth input capture or a fifth output compare. PA3 can also be configured to allow OC1 edges to trigger IC4 captures.

PA[2:0] serve as general-purpose inputs or as IC[1:3].

PORTA can be read at any time. Reads of pins configured as inputs return the logic level present on the pin. Pins configured as outputs return the logic level present at the pin driver input. If written, PORTA stores the data in an internal latch, bits 7 and 3. It drives the pins only if they are configured as outputs. Writes to PORTA do not change the pin state when pins are configured for timer input captures or output compares. Refer to **SECTION 6 PARALLEL INPUT/OUTPUT**.

2.11.2 Port B

During single-chip operating modes, all port B pins are general-purpose output pins. During MCU reads of this port, the level sensed at the input side of the port B output drivers is read. Port B can also be used in simple strobed output mode. In this mode, an output pulse appears at the STRB signal each time data is written to port B.

In expanded multiplexed operating modes, all of the port B pins act as high order address output signals. During each MCU cycle, bits [15:8] of the address bus are output on the PB[7:0] pins. The PORTB register is treated as an external address in expanded modes.

2.11.3 Port C

While in single-chip operating modes, all port C pins are general-purpose input/output pins. Port C inputs can be latched into an alternate PORTCL register by providing an input transition to the STRA signal. Port C can also be used in full handshake modes of parallel I/O where the STRA input and STRB output act as handshake control lines.

When in expanded multiplexed modes, all port C pins are configured as multiplexed address/data signals. During the address portion of each MCU cycle, bits [7:0] of the address are output on the PC[7:0] pins. During the data portion of each MCU cycle (E high), PC[7:0] are bidirectional data signals, DATA[7:0]. The direction of data at the port C pins is indicated by the $R/\overline{W}$ signal.

The CWOM control bit in the PIOC register disables the port C P-channel output driver. CWOM simultaneously affects all eight bits of port C. 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, when a port C bit is at logic level zero, it is driven low by the N-channel driver. When a port C bit is at logic level one, the associated pin has high-impedance, as neither the N- 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 mode. Refer to **SECTION 6 PARALLEL INPUT/OUTPUT** for additional information about port C functions.

2.11.4 Port D

Pins PD[5:0] can be used for general-purpose I/O signals. These pins alternately serve as the serial communication interface (SCI) and serial peripheral interface (SPI) signals when those subsystems are enabled.

Pin PD0 is the receive data input (RxD) signal for the SCI.

Pin PD1 is the transmit data output (TxD) signal for the SCI.

Pins PD[5:2] are dedicated to the serial peripheral interface (SPI). PD2 is the master in slave out (MISO) signal. PD3 is the master out slave in (MOSI) signal. PD4 is the serial clock (SCK) signal and PD5 is the slave select ($\overline{SS}$) input.

2.11.5 Port E

Use port E for general-purpose or analog-to-digital (A/D) inputs. If high accuracy is required for A/D conversions, avoid reading port E during sampling, as small disturbances can reduce the accuracy of that result.

SECTION 3
CENTRAL PROCESSING UNIT

This section presents information on M68HC11 central processing unit (CPU) architecture, data types, addressing modes, the instruction set, and 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**.

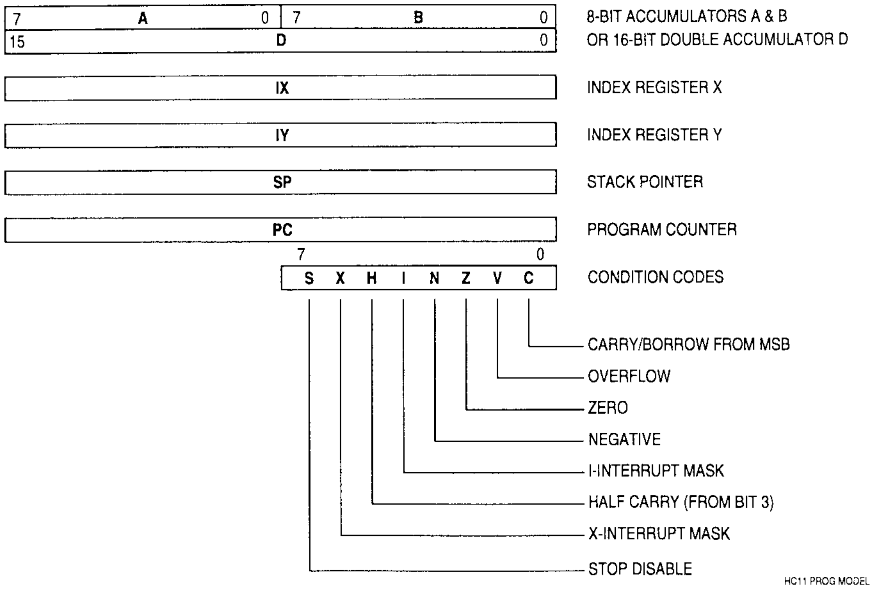


Figure 3–1 Programming Model

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)

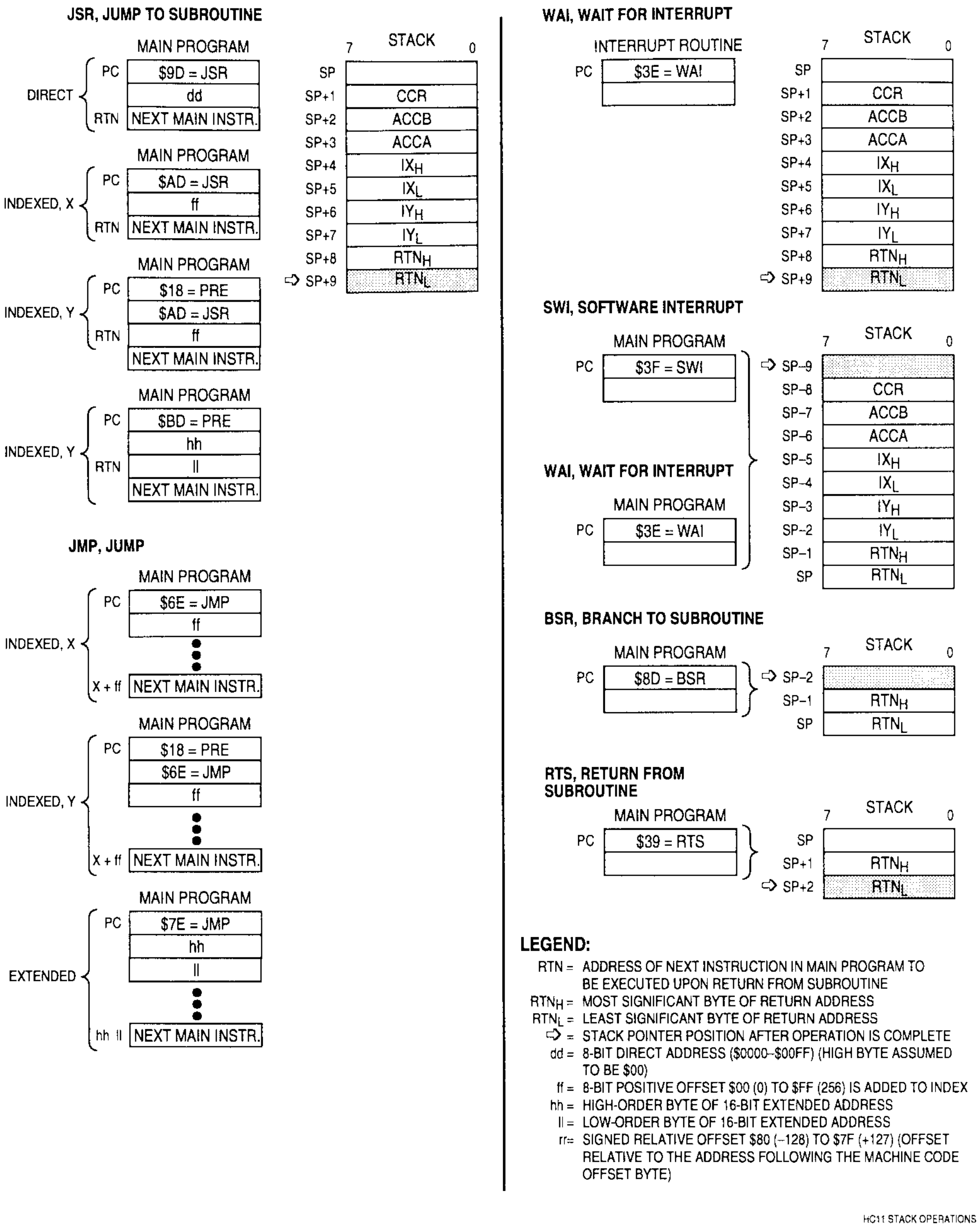
The IX register provides a 16-bit indexing value that can be added to the 8-bit offset provided in an instruction to create an effective address. The IX register can also be used as a counter or as a temporary storage register.

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.



LEGEND:

RTN = ADDRESS OF NEXT INSTRUCTION IN MAIN PROGRAM TO BE EXECUTED UPON RETURN FROM SUBROUTINE
$\mathrm{RTN_H}$ = MOST SIGNIFICANT BYTE OF RETURN ADDRESS
$\mathrm{RTN_L}$ = LEAST SIGNIFICANT BYTE OF RETURN ADDRESS
⇨ = STACK POINTER POSITION AFTER OPERATION IS COMPLETE
dd = 8-BIT DIRECT ADDRESS ($0000–$00FF) (HIGH BYTE ASSUMED TO BE $00)
ff = 8-BIT POSITIVE OFFSET $00 (0) TO $FF (256) IS ADDED TO INDEX
hh = HIGH-ORDER BYTE OF 16-BIT EXTENDED ADDRESS
ll = LOW-ORDER BYTE OF 16-BIT EXTENDED ADDRESS
rr= SIGNED RELATIVE OFFSET $80 (–128) TO $7F (+127) (OFFSET RELATIVE TO THE ADDRESS FOLLOWING THE MACHINE CODE OFFSET BYTE)

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, ($\overline{\text{IRQ}}$ and $\overline{\text{XIRQ}}$) 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 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 STOP instruction is encountered by the CPU while the S bit is set, it is treated as a no-operation (NOP) instruction, and processing continues to the next instruction. S is set by reset — STOP 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)

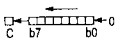
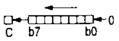
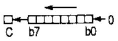
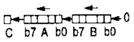
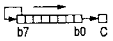
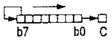
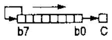
Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
ABA	Add Accumulators	A + B ⇒ A	INH	1B	—	2	—	—	Δ	—	Δ	Δ	Δ	Δ
ABX	Add B to X	IX + (00 : B) ⇒ IX	INH	3A	—	3	—	—	—	—	—	—	—	—
ABY	Add B to Y	IY + (00 : B) ⇒ IY	INH	18 3A	—	4	—	—	—	—	—	—	—	—
ADCA (opr)	Add with Carry to A	A + M + C ⇒ A	A IMM	89	ii	2	—	—	Δ	—	Δ	Δ	Δ	Δ
			A DIR	99	dd	3								
			A EXT	B9	hh ll	4								
			A IND,X	A9	ff	4								
			A IND,Y	18 A9	ff	5								
ADCB (opr)	Add with Carry to B	B + M + C ⇒ B	B IMM	C9	ii	2	—	—	Δ	—	Δ	Δ	Δ	Δ
			B DIR	D9	dd	3								
			B EXT	F9	hh ll	4								
			B IND,X	E9	ff	4								
			B IND,Y	18 E9	ff	5								
ADDA (opr)	Add Memory to A	A + M ⇒ A	A IMM	8B	ii	2	—	—	Δ	—	Δ	Δ	Δ	Δ
			A DIR	9B	dd	3								
			A EXT	BB	hh ll	4								
			A IND,X	AB	ff	4								
			A IND,Y	18 AB	ff	5								
ADDB (opr)	Add Memory to B	B + M ⇒ B	B IMM	CB	ii	2	—	—	Δ	—	Δ	Δ	Δ	Δ
			B DIR	DB	dd	3								
			B EXT	FB	hh ll	4								
			B IND,X	EB	ff	4								
			B IND,Y	18 EB	ff	5								
ADDD (opr)	Add 16-Bit to D	D + (M : M + 1) ⇒ D	IMM	C3	jj kk	4	—	—	—	—	Δ	Δ	Δ	Δ
			DIR	D3	dd	5								
			EXT	F3	hh ll	6								
			IND,X	E3	ff	6								
			IND,Y	18 E3	ff	7								
ANDA (opr)	AND A with Memory	A • M ⇒ A	A IMM	84	ii	2	—	—	—	—	Δ	Δ	0	—
			A DIR	94	dd	3								
			A EXT	B4	hh ll	4								
			A IND,X	A4	ff	4								
			A IND,Y	18 A4	ff	5								
ANDB (opr)	AND B with Memory	B • M ⇒ B	B IMM	C4	ii	2	—	—	—	—	Δ	Δ	0	—
			B DIR	D4	dd	3								
			B EXT	F4	hh ll	4								
			B IND,X	E4	ff	4								
			B IND,Y	18 E4	ff	5								
ASL (opr)	Arithmetic Shift Left		EXT	78	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND,X	68	ff	6								
			IND,Y	18 68	ff	7								
ASLA	Arithmetic Shift Left A		A INH	48	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLB	Arithmetic Shift Left B		B INH	58	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLD	Arithmetic Shift Left D		INH	05	—	3	—	—	—	—	Δ	Δ	Δ	Δ
ASR	Arithmetic Shift Right		EXT	77	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND,X	67	ff	6								
			IND,Y	18 67	ff	7								
ASRA	Arithmetic Shift Right A		A INH	47	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASRB	Arithmetic Shift Right B		B INH	57	—	2	—	—	—	—	Δ	Δ	Δ	Δ
BCC (rel)	Branch if Carry Clear	? C = 0	REL	24	rr	3	—	—	—	—	—	—	—	—
BCLR (opr) (msk)	Clear Bit(s)	M • (mm) ⇒ M	DIR	15	dd mm	6	—	—	—	—	Δ	Δ	0	—
			IND,X	1D	ff mm	7								
			IND,Y	18 1D	ff mm	8								
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	85	ii	2	—	—	—	—	Δ	Δ	0	—
			A DIR	95	dd	3								
			A EXT	B5	hh ll	4								
			A IND,X	A5	ff	4								
			A IND,Y	18 A5	ff	5								
BITB (opr)	Bit(s) Test B with Memory	B • M	B IMM	C5	ii	2	—	—	—	—	Δ	Δ	0	—
			B DIR	D5	dd	3								
			B EXT	F5	hh ll	4								
			B IND,X	E5	ff	4								
			B IND,Y	18 E5	ff	5								
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	13	dd mm rr	6	—	—	—	—	—	—	—	—
			IND,X	1F	ff mm rr	7								
			IND,Y	18 1F	ff mm rr	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	12	dd mm rr	6	—	—	—	—	—	—	—	—
			IND,X	1E	ff mm rr	7								
			IND,Y	18 1E	ff mm rr	8								
BSET (opr) (msk)	Set Bit(s)	M + mm ⇒ M	DIR	14	dd mm	6	—	—	—	—	Δ	Δ	0	—
			IND,X	1C	ff mm	7								
			IND,Y	18 1C	ff mm	8								
BSR (rel)	Branch to Subroutine	See Figure 3–2	REL	8D	rr	6	—	—	—	—	—	—	—	—
BVC (rel)	Branch if Overflow Clear	? V = 0	REL	28	rr	3	—	—	—	—	—	—	—	—
BVS (rel)	Branch if Overflow Set	? V = 1	REL	29	rr	3	—	—	—	—	—	—	—	—
CBA	Compare A to B	A – B	INH	11	—	2	—	—	—	—	Δ	Δ	Δ	Δ
CLC	Clear Carry Bit	0 ⇒ C	INH	0C	—	2	—	—	—	—	—	—	—	0
CLI	Clear Interrupt Mask	0 ⇒ I	INH	0E	—	2	—	—	—	0	—	—	—	—
CLR (opr)	Clear Memory Byte	0 ⇒ M	EXT	7F	hh ll	6	—	—	—	—	0	1	0	0
			IND,X	6F	ff	6								
			IND,Y	18 6F	ff	7								
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	81	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
			A DIR	91	dd	3								
			A EXT	B1	hh ll	4								
			A IND,X	A1	ff	4								
			A IND,Y	18 A1	ff	5								
CMPB (opr)	Compare B to Memory	B – M	B IMM	C1	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
			B DIR	D1	dd	3								
			B EXT	F1	hh ll	4								
			B IND,X	E1	ff	4								
			B IND,Y	18 E1	ff	5								

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

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	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)

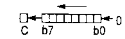
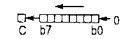
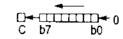
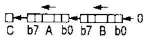
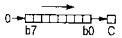
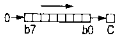
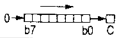
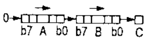
Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
INY	Increment Index Register Y	IY + 1 ⇒ IY	INH	18 08	—	4	—	—	—	—	—	Δ	—	—
JMP (opr)	Jump	See Figure 3–2	EXT	7E	hh ll	3	—	—	—	—	—	—	—	—
			IND,X	6E	ff	3								
			IND,Y	18 6E	ff	4								
JSR (opr)	Jump to Subroutine	See Figure 3–2	DIR	9D	dd	5	—	—	—	—	—	—	—	—
			EXT	BD	hh ll	6								
			IND,X	AD	ff	6								
			IND,Y	18 AD	ff	7								
LDAA (opr)	Load Accumulator A	M ⇒ A	A IMM	86	ii	2	—	—	—	—	Δ	Δ	0	—
			A DIR	96	dd	3								
			A EXT	B6	hh ll	4								
			A IND,X	A6	ff	4								
			A IND,Y	18 A6	ff	5								
LDAB (opr)	Load Accumulator B	M ⇒ B	B IMM	C6	ii	2	—	—	—	—	Δ	Δ	0	—
			B DIR	D6	dd	3								
			B EXT	F6	hh ll	4								
			B IND,X	E6	ff	4								
			B IND,Y	18 E6	ff	5								
LDD (opr)	Load Double Accumulator D	M ⇒ A,M + 1 ⇒ B	IMM	CC	jj kk	3	—	—	—	—	Δ	Δ	0	—
			DIR	DC	dd	4								
			EXT	FC	hh ll	5								
			IND,X	EC	ff	5								
			IND,Y	18 EC	ff	6								
LDS (opr)	Load Stack Pointer	M : M + 1 ⇒ SP	IMM	8E	jj kk	3	—	—	—	—	Δ	Δ	0	—
			DIR	9E	dd	4								
			EXT	BE	hh ll	5								
			IND,X	AE	ff	5								
			IND,Y	18 AE	ff	6								
LDX (opr)	Load Index Register X	M : M + 1 ⇒ IX	IMM	CE	jj kk	3	—	—	—	—	Δ	Δ	0	—
			DIR	DE	dd	4								
			EXT	FE	hh ll	5								
			IND,X	EE	ff	5								
			IND,Y	CD EE	ff	6								
LDY (opr)	Load Index Register Y	M : M + 1 ⇒ IY	IMM	18 CE	jj kk	4	—	—	—	—	Δ	Δ	0	—
			DIR	18 DE	dd	5								
			EXT	18 FE	hh ll	6								
			IND,X	1A EE	ff	6								
			IND,Y	18 EE	ff	6								
LSL (opr)	Logical Shift Left		EXT	78	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND,X	68	ff	6								
			IND,Y	18 68	ff	7								
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	74	hh ll	6	—	—	—	—	0	Δ	Δ	Δ
			IND,X	64	ff	6								
			IND,Y	18 64	ff	7								
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	70	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND,X	60	ff	6								
			IND,Y	18 60	ff	7								
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)

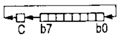
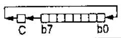
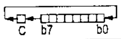
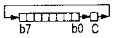
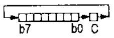
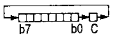
Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
NOP	No operation	No Operation	INH	01	—	2	—	—	—	—	—	—	—	—
ORAA (opr)	OR Accumulator A (Inclusive)	A + M ⇒ A	A IMM A DIR A EXT A IND,X A IND,Y	8A 9A BA AA 18 AA	ii dd hh ll ff ff	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			Condition Codes							
				Opcode	Operand	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	—	—	—	—	—	—	—	—

Cycle

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

Operands

dd	= 8-Bit Direct Address ($0000 –$00FF) (High Byte Assumed to be $00)
ff	= 8-Bit Positive Offset $00 (0) to $FF (255) (Is Added to Index)
hh	= High-Order Byte of 16-Bit Extended Address
ii	= One Byte of Immediate Data
jj	= High-Order Byte of 16-Bit Immediate Data
kk	= Low-Order Byte of 16-Bit Immediate Data
ll	= Low-Order Byte of 16-Bit Extended Address
mm	= 8-Bit Mask (Set Bits to be Affected)
rr	= Signed Relative Offset $80 (−128) to $7F (+127) (Offset Relative to Address Following Machine Code Offset Byte))

Operators

( )	Contents of register shown inside parentheses
⇐	Is transferred to
⇑	Is pulled from stack
⇓	Is pushed onto stack
•	Boolean AND
+	Arithmetic Addition Symbol except where used as Inclusive-OR symbol in Boolean Formula
⊕	Exclusive-OR
*	Multiply
:	Concatenation
−	Arithmetic subtraction symbol or Negation symbol (Two's Complement)

Condition Codes

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

SECTION 4
OPERATING MODES AND ON-CHIP MEMORY

This section contains information about the operating modes and the on-chip memory for M68HC11 E-series MCUs. Except for a few minor differences, operation is identical for all devices in the E series. Differences are noted where necessary.

4.1 Operating Modes

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

4.1.1 Single-Chip Mode

In single-chip mode, ports B and C and strobe pins A (STRA) and B (STRB) are available for general-purpose parallel I/O. In this mode, all software needed to control the MCU is contained in internal resources. ROM/EPROM (if present) will always be enabled out of reset, ensuring that the reset and interrupt vectors will be available at locations $FFC0–$FFFF. For the MC68HC811E2, the vector locations are the same, however, they are contained in the 2048-byte EEPROM array.

4.1.2 Expanded Mode

In expanded operating mode, the MCU can access the full 64-Kbyte address space. The space includes the same on-chip memory addresses used for single-chip mode as well as addresses for external peripherals and memory devices. The expansion bus is made up of ports B and C, and control signals AS and R/$\overline{W}$. The R/$\overline{W}$ (read/write) and AS (address strobe) allow the low-order address and the 8-bit data bus to be multiplexed on the same pins. During the first half of each bus cycle address information is present. During the second half of each bus cycle the pins become the bidirectional data bus. AS is an active-high latch enable signal for an external address latch. Address information is allowed through the transparent latch while AS is high and is latched when AS drives low.

The address, R/$\overline{W}$, and AS signals are active and valid for all bus cycles, including accesses to internal memory locations. The E clock is used to enable external devices to drive data onto the internal data bus during the second half of a read bus cycle (E clock high). R/$\overline{W}$ controls the direction of data transfers. R/$\overline{W}$ drives low when data is being written to the internal data bus. R/$\overline{W}$ will remain low during consecutive data bus write cycles, such as when a double-byte store occurs. Notice that the write enable signal for an external memory is the NAND of the E clock and the inverted R/$\overline{W}$ signal. Refer to the example diagram of address and data demultiplexing.

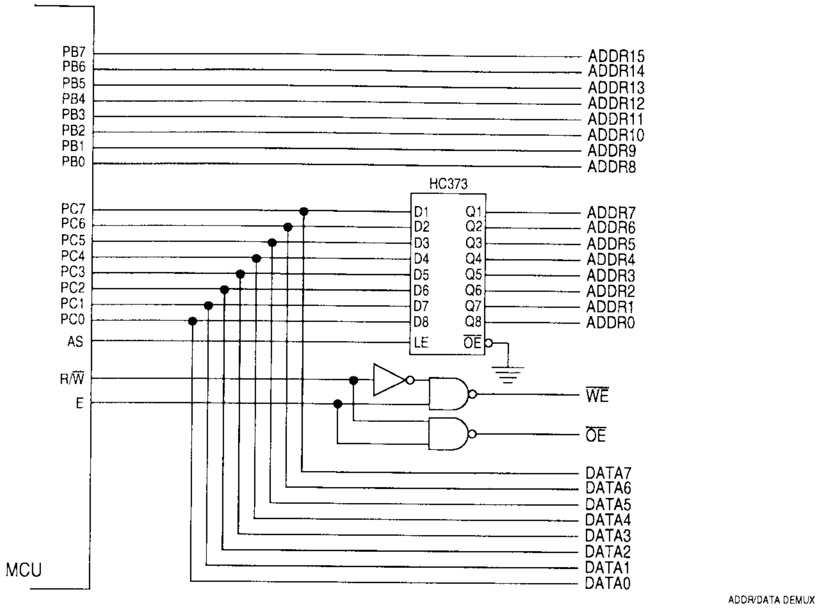


Figure 4–1 Address/Data Demultiplexing

4.1.3 Test Mode

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 Bootstrap Mode

When the MCU is reset in special bootstrap mode, a small on-chip ROM is enabled at address $BF00–$BFFF. The ROM contains a bootloader program and a special set of interrupt and reset vectors. The MCU fetches the reset vector, then executes the bootloader. 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 this ROM at $BFC0–$BFFF. The bootstrap ROM contains a small program which initializes the SCI and allows the user to download a program into on-chip RAM. The size of the downloaded program can be as large as the size of the on-chip RAM. After a four-character delay, or after receiving the character for the highest address in RAM, control passes to the loaded program at $0000. Refer to the memory map diagram.

Use of an external pull-up resistor is required when using the SCI transmitter pin because port D pins are configured for wired-OR operation by the bootloader. In bootstrap mode, the interrupt vectors are directed to RAM. This allows the use of interrupts through a jump table. Refer to Motorola application note AN1060, M68HC11 Bootstrap Mode.

4.2 Memory Map

The operating mode determines memory mapping and whether external addresses can be accessed. Refer to **Figure 4–2**, **Figure 4–3**, and **Figure 4–4**, which illustrate the memory maps for each of the three families comprising the M68HC11 E series of MCUs. Memory locations for on-chip resources are the same for both expanded and single-chip modes. Control bits in the CONFIG register allow EPROM and EEPROM (if present) to be disabled from the memory map. The RAM is mapped to $0000 after reset. It can be placed at any 4-Kbyte boundary ($x000) by writing an appropriate value to the INIT register. The 64-byte register block is mapped to $1000 after reset and can also be placed at any 4-Kbyte boundary ($x000) by writing an appropriate value to the INIT register. If RAM and registers are mapped to the same boundary, the first 64 bytes of RAM will be inaccessible. Refer to **Table 4–1**, which details the MCU register and control bit assignments.

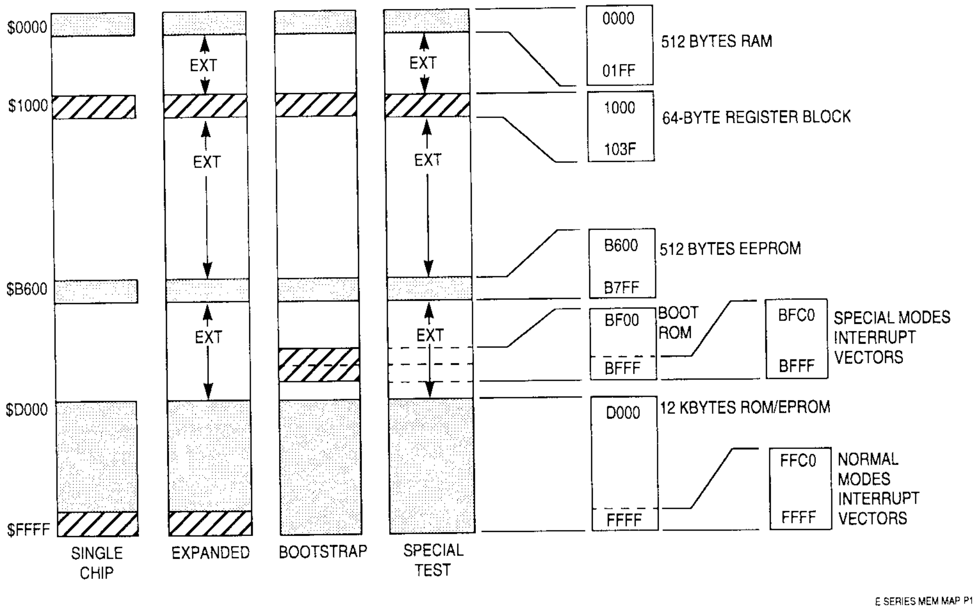


Figure 4–2 Memory Map for MC68HC11E0, MC68HC11E1, MC68HC11E8, and MC68HC(7)11E9

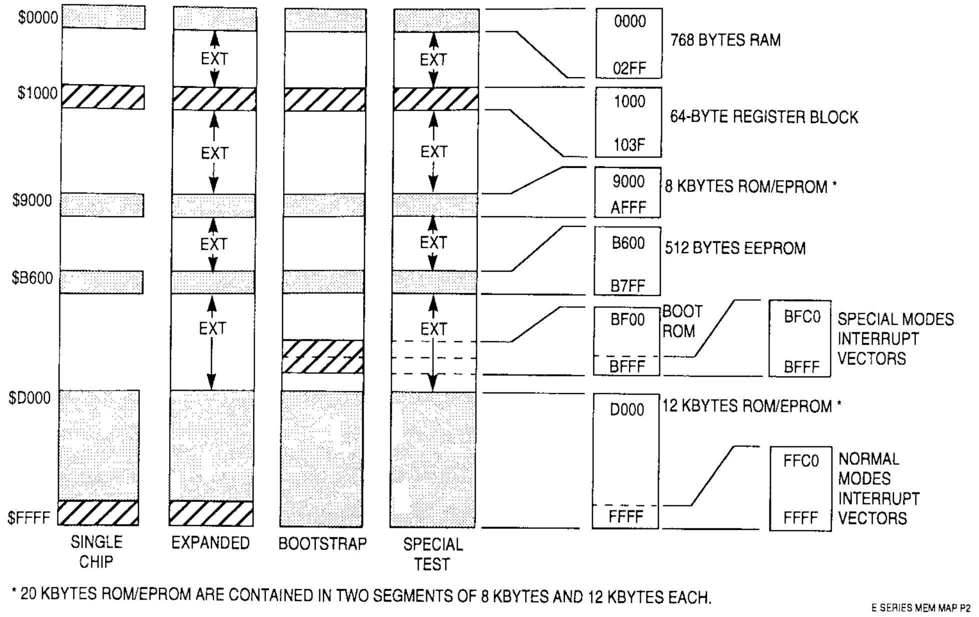


Figure 4–3 Memory Map for MC68HC(7)11E20

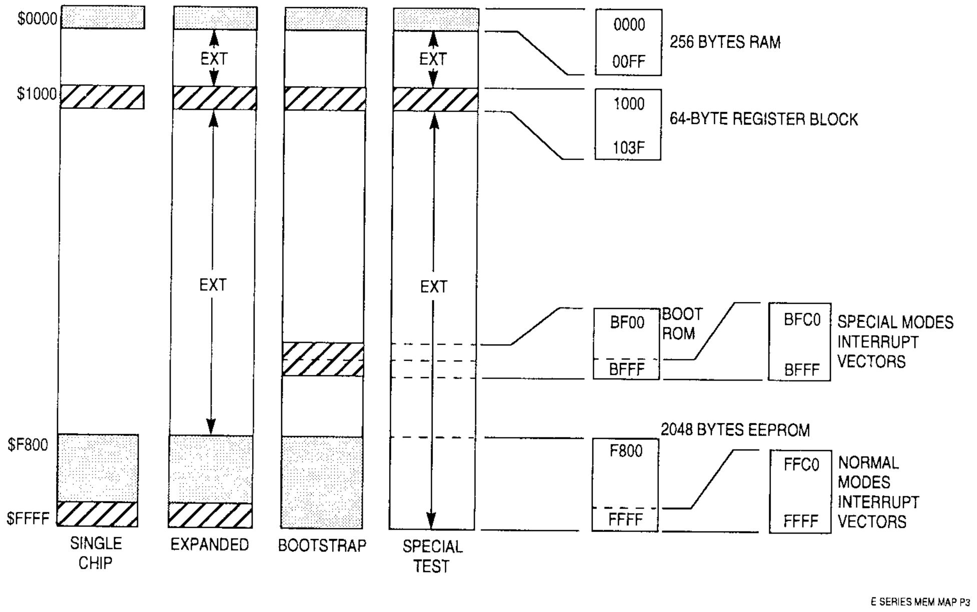


Figure 4–4 Memory Map for MC68HC811E2

Table 4–1 Register and Control Bit Assignments

The register block can be remapped to any 4K boundary

	Bit 7	6	5	4	3	2	1	Bit 0	
$1000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA
$1001									Reserved
$1002	STAF	STAI	CWOM	HNDS	OIN	PLS	EGA	INVB	PIOC
$1003	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC
$1004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
$1005	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0	PORTCL
$1006									Reserved
$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

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

$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	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0	PACTL
$1027	Bit 7	6	5	4	3	2	1	Bit 0	PACNT
$1028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
$1029	SPIF	WCOL	0	MODF	0	0	0	0	SPSR
$102A	Bit 7	6	5	4	3	2	1	Bit 0	SPDR
$102B	TCLR	SCP2[1]	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	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0	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	MBE	0	ELAT	EXCOL	EXROW	T1	T0	PGM	EPROG[2]
$1037									Reserved
$1038									Reserved
$1039	ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR0	OPTION
$103A	Bit 7	6	5	4	3	2	1	Bit 0	COPRST
$103B	ODD	EVEN	ELAT[3]	BYTE	ROW	ERASE	EELAT	EPGM	PPROG
$103C	RBOOT	SMOD	MDA	IRVNE	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO
$103D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT
$103E	TILOP	0	OCCR	CBYP	DISR	FCM	FCOP	TCON	TEST1
$103F	EE3[4]	EE2[4]	EE1[4]	EE0[4]	NOSEC	NOCOP	ROMON	EEON	CONFIG

NOTES:

1. SCP2 adds ÷39 to SCI prescaler and is present only in MC68HC(7)11E20.
2. EPROG is present only in MC68HC711E20.
3. ELAT in PPROG is present only on MC68HC711E9 and MC68S711E9.
4. EEPROM mapping control bits EE[3:0] apply only to MC68HC811E2.

Hardware priority is built into RAM and I/O mapping. Registers have priority over RAM and RAM has priority over ROM. When a lower priority resource is mapped at the same location as a higher priority resource, a read/write of a location results in a read/write of the higher priority resource only. For example, if both the register block and the RAM are mapped to the same location, only the register block will be accessed. If RAM and ROM are located at the same position, RAM has priority.

The fully static RAM can be used to store instructions, variables, and temporary data. The direct addressing mode can access RAM locations using a one-byte address operand, saving program memory space and execution time, depending on the application.

RAM contents can be preserved during periods of processor inactivity by two methods, both of which reduce power consumption.

In the software-based STOP mode, the clocks are stopped while V_{DD} powers the MCU. Because power supply current is directly related to operating frequency in CMOS integrated circuits, only a very small amount of leakage exists when the clocks are stopped.

In the second method, the MODB/V_{STBY} pin can supply RAM power from a battery backup or from a second power supply. **Figure 4–5** shows a typical standby voltage circuit for a standard 5 volt device. Adjustments to the circuit must be made for devices that operate at lower voltages. Using the MODB/V_{STBY} pin may require external hardware, but can be justified when a significant amount of external circuitry is operating from V_{DD}. If V_{STBY} is used to maintain RAM contents, reset must be held low whenever V_{DD} is below normal operating level. Refer to **SECTION 5 RESETS AND INTERRUPTS**.

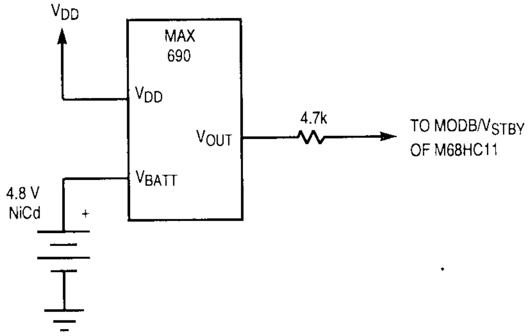


Figure 4–5 RAM Standby MODB/V_{STBY} Connections

The bootloader program is contained in the internal bootstrap ROM. This ROM, which appears as internal memory space at locations $BF00–$BFFF, is enabled only if the MCU is reset in special bootstrap mode.

In expanded modes, the ROM/EPROM/OTPROM (if present) is enabled out of reset and located at the top of the memory map if the ROMON bit in the CONFIG register is set. ROM or EPROM is enabled out of reset in single-chip and bootstrap modes, regardless of the state of ROMON.

For devices with 512 bytes of EEPROM, the EEPROM is located at $B600–$B7FF and has the same read cycle time as the internal ROM. The 512 bytes of EEPROM cannot be remapped to other locations.

For the MC68HC811E2, EEPROM is located at $F800–$FFFF and can be remapped to any 4-Kbyte boundary. EEPROM mapping control bits (EE[3:0] in CONFIG) determine the location of the 2048 bytes of EEPROM and are present only on the MC68HC811E2. Refer to the description of the CONFIG register for MC68HC811E2.

EEPROM can be programmed or erased by software and an on-chip charge pump, allowing EEPROM changes using the single 5 Vdc supply.

4.2.1 Mode Selection

The four mode variations are selected by the logic states of the MODA and MODB pins during reset. The MODA and MODB logic levels determine the logic state of SMOD and the MDA control bits in the highest priority I-bit interrupt and miscellaneous (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 is normally connected to V_{DD} through a pull-up resistor of 4.7 kΩ. The MODA pin also functions as the load instruction register $\overline{LIR}$ pin when the MCU is not in reset. The open-drain active low $\overline{LIR}$ output pin drives low during the first E cycle of each instruction. 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 **Table 4–2**, which is a summary of mode pin operation, the mode control bits, and the four operating modes.

Table 4–2 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	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**.

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

	Bit 7	6	5	4	3	2	1	Bit 0	
	RBOOT[1]	SMOD[1]	MDA[1]	IRV(NE)[1]	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	0	0	1	1	0	Bootstrap
	0	1	1	1	0	1	1	0	Test

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

RBOOT — Read Bootstrap ROM

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

0 = Bootloader ROM disabled and not in map
1 = Bootloader ROM enabled and in map at $BE00–$BFFF

SMOD and MDA — Special Mode Select and Mode Select A

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 anytime in special modes. MDA can only be written once in normal modes. SMOD cannot be set once it has been cleared.

Inputs		**Mode**	**Latched at Reset**	
MODB	**MODA**		**SMOD**	**MDA**
1	0	Single Chip	0	0
1	1	Expanded	0	1
0	0	Bootstrap	1	0
0	1	Special Test	1	1

IRV(NE) — Internal Read Visibility (/Not E)

IRVNE can be written once in any mode. In expanded modes, IRVNE determines whether IRV is on or off. In special test mode, IRVNE is reset to one. In all other modes, IRVNE is reset to zero. For the MC68HC811E2, this bit is IRV and only controls the internal read visibility function.

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

In single-chip modes this bit determines whether the E clock drives out from the chip. For the MC68HC811E2, this bit has no meaning or effect in single-chip and bootstrap modes.

0 = E is driven out from the chip.
1 = E pin is driven low. Refer to the following table.

Mode	IRVNE Out of Reset	E Clock Out of Reset	IRV Out of Reset	IRVNE Affects Only	IRVNE Can Be Written
Single Chip	0	On	Off	E	Once
Expanded	0	On	Off	IRV	Once
Bootstrap	0	On	Off	E	Once
Special Test	1	On	On	IRV	Once

PSEL[3:0] — Priority Select Bits [3:0]
Refer to **SECTION 5 RESETS AND INTERRUPTS**.

4.2.2 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 4–3 Write Access Limited Registers

Operating Modes	Register Address	Register Name	Must be Written in First 64 Cycles	Write Anytime
SMOD = 0	$x024	Timer Interrupt Mask 2 (TMSK2)	Bits [1:0], once only	Bits [7:2]
	$x035	Block Protect Register (BPROT)	Clear bits, once only	Set bits only
	$x039	System Configuration Options (OPTION)	Bits [5:4], Bits [2:0], once only	Bits [7:6], Bit 3
	$x03C	Highest Priority I-Bit Interrupt and Miscellaneous (HPRIO)	See HPRIO description	See HPRIO description
	$x03D	RAM and I/O Map Register (INIT)	Yes, once only	—
SMOD = 1	$x024	Timer Interrupt Mask 2 (TMSK2)	—	All, set or clear
	$x035	Block Protect Register (BPROT)	—	All, set or clear
	$x039	System Configuration Options (OPTION)	—	All, set or clear
	$x03C	Highest Priority I-Bit Interrupt and Miscellaneous (HPRIO)	See HPRIO description	See HPRIO description
	$x03D	RAM and I/O Map Register (INIT)	—	All, set or clear

4.2.2.1 CONFIG Register

The CONFIG register consists of an EEPROM byte and static latches that control the startup 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.

Operation of the CONFIG register in the MC68HC811E2 differs from other devices in the M68HC11 E series. Refer to the two descriptions of the CONFIG register.

CONFIG — System Configuration **$103F**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	—	—	NOSEC	NOCOP	ROMON	EEON
RESETS:								
S. Chip:	0	0	0	0	U	U	1	U
Boot:	0	0	0	0	U	U(L)	U	U
Exp.:	0	0	0	0	1	U	U	U
Test:	0	0	0	0	1	U(L)	U	U

CONFIG — System Configuration (MC68HC811E2 only) **$103F**

Bit 7	6	5	4	3	2	1	Bit 0	
	EE3	EE2	EE1	EE0	NOSEC	NOCOP	—	EEON
RESETS:								
S. chip	1	1	1	1	U	U	1	1
Boot	1	1	1	1	U	U(L)	1	1
Exp	U	U	U	U	1	U	1	U
Test	U	U	U	U	1	U(L)	1	0

U indicates a previously programmed bit. U(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.

EE[3:0] — EEPROM Mapping

EE[3:0] apply only to MC68HC811E2 and allow the 2048 bytes of EEPROM to be remapped to any 4-Kbyte boundary. Refer to **Table 4–4**.

Table 4–4 EEPROM Mapping

EE[3:0]	EEPROM Location
0 0 0 0	$0800 – $OFFF
0 0 0 1	$1800 – $1FFF
0 0 1 0	$2800 – $2FFF
0 0 1 1	$3800 – $3FFF
0 1 0 0	$4800 – $4FFF
0 1 0 1	$5800 – $5FFF
0 1 1 0	$6800 – $6FFF
0 1 1 1	$7800 – $7FFF
1 0 0 0	$8800 – $8FFF
1 0 0 1	$9800 – $8FFF
1 0 1 0	$A800 – $AFFF
1 0 1 1	$B800 – $BFFF
1 1 0 0	$C800 – $CFFF
1 1 0 1	$D800 – $DFFF
1 1 1 0	$E800 – $EFFF
1 1 1 1	$F800 – $FFFF

NOSEC — Security Disable

NOSEC is invalid unless the security mask option is specified before the MCU is manufactured. If the security mask option is omitted NOSEC always reads one. The enhanced security feature is available in the MC68S711E9 MCU. The enhancement to the standard security feature protects the EPROM as well as RAM and EEPROM.

0 = Security enabled
1 = Security disabled

NOCOP — COP System Disable

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

ROMON — ROM/EPROM/OTPROM Enable

When this bit is zero, the ROM or EPROM is disabled and that memory space becomes externally addressed. In single-chip mode, ROMON is forced to one to enable ROM/EPROM regardless of the state of the ROMON bit.

0 = ROM disabled from the memory map
1 = ROM present in the memory map

EEON — EEPROM Enable

When this bit is zero, the EEPROM is disabled and that memory space becomes externally addressed.

0 = EEPROM removed from the memory map
1 = EEPROM present in the memory map

4.2.2.2 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 in normal modes, and then it 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. It is initialized to address $0000 out of reset. Refer to **Table 4–5**.

REG[3:0] — 64-Byte Register Block Position

These four bits specify the upper hexadecimal digit of the address for the 64-byte block of internal registers. The register block, positioned at the beginning of any 4-Kbyte page in the memory map, is initialized to address $1000 out of reset. Refer to **Table 4–6**.

Table 4–5 RAM Mapping

RAM[3:0]	Address
0000	$0000 – $0xFF
0001	$1000 – $1xFF
0010	$2000 – $2xFF
0011	$3000 – $3xFF
0100	$4000 – $4xFF
0101	$5000 – $5xFF
0110	$6000 – $6xFF
0111	$7000 – $7xFF
1000	$8000 – $8xFF
1001	$9000 – $9xFF
1010	$A000 – $AxFF
1011	$B000 – $BxFF
1100	$C000 – $CxFF
1101	$D000 – $DxFF
1110	$E000 – $ExFF
1111	$F000 – $FxFF

Table 4–6 Register Mapping

REG[3:0]	Address
0000	$0000 – $003F
0001	$1000 – $103F
0010	$2000 – $203F
0011	$3000 – $303F
0100	$4000 – $403F
0101	$5000 – $503F
0110	$6000 – $603F
0111	$7000 – $703F
1000	$8000 – $803F
1001	$9000 – $903F
1010	$A000 – $A03F
1011	$B000 – $B03F
1100	$C000 – $C03F
1101	$D000 – $D03F
1110	$E000 – $E03F
1111	$F000 – $F03F

4.2.2.3 OPTION Register

The 8-bit special-purpose OPTION register sets internal system configuration options during initialization. The time protected control bits, IRQE, DLY, and CR[1:0] can be written only once after a reset and then become read-only. This minimizes the possibility of any accidental changes to the system configuration.

OPTION — System Configuration Options **$1039**

	Bit 7	6	5	4	3	2	1	Bit 0
	ADPU	CSEL	IRQE[1]	DLY[1]	CME	—	CR1[1]	CR0[1]
RESET:	0	0	0	1	0	0	0	0

NOTES:

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

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

CSEL — Clock Select
Selects alternate clock source for on-chip EEPROM charge pump. Refer to **4.4.1 EEPROM Programming** for more information on EEPROM use. CSEL also selects the clock source for the A/D converter, a function discussed in **SECTION 10 ANALOG-TO-DIGITAL CONVERTER**.

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

DLY — Enable Oscillator Startup Delay

0 = The oscillator startup 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. This delay allows the crystal oscillator to stabilize.

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

Bit 2 — Not implemented
Always reads zero

CR[1:0] — COP Timer Rate Select Bits
The internal E clock is divided by 2^{15} before it enters the COP watchdog system. These control bits determine a scaling factor for the watchdog timer. Refer to **SECTION 5 RESETS AND INTERRUPTS**.

4.3 EPROM/OTPROM

Certain devices in the M68HC11 E series include on-chip EPROM/OTPROM. The MC68HC711E9 devices contain 12 Kbytes of on-chip EPROM (OTPROM in non-windowed package). The MC68HC711E20 has 20 Kbytes of EPROM (OTPROM in non-windowed package).

Using the on-chip EPROM/OTPROM programming feature requires an external 12 volt nominal power supply (V_{PPE}). Normal programming is accomplished using the EPROM/OTPROM programming register (PPROG). PPROG is the combined EPROM/OTPROM and EEPROM programming register on all devices with EPROM/OTPROM except the MC68HC711E20. For the MC68HC711E20, there is a separate register for EPROM/OTPROM programming called the EPROG register. There are two possible methods of programming and verifying EPROM.

4.3.1 Programming an Individual EPROM Address

In this method, the MCU programs its own EPROM by controlling the PPROG register (EPROG in MC68HC711E20). Use the following procedure to program the EPROM through the MCU with the ROMON bit set in the CONFIG register. The 12 volt nominal programming voltage must be present on the $\overline{\text{XIRQ}}$/V_{PPE} pin. Any operating mode can be used. The first example applies to all devices with EPROM/OTPROM except MC68HC711E20. The second example applies to the MC68HC711E20 only.

```
EPROG     LDAB   #$20
          STAB   $103B    Set ELAT bit in (EPGM=0) to enable EPROM
                          latches.
          STAA   $0,X     Store data to EPROM address
          LDAB   #$21
          STAB   $103B    Set EPGM bit with ELAT=1 to enable EPROM
                          programming voltage
          JSR    DLYEP    Delay 2-4 ms
          CLR    $103B    Turn off programming voltage and set to READ mode
```

The following example applies only to MC68HC711E20.

```
EPROG     LDAB   #$20
          STAB   $1036    Set ELAT bit (EPGM=0) to enable EPROM latches.
          STAA   $0,X     Store data to EPROM address
          LDAB   #$21
          STAB   $1036    Set EPGM bit with ELAT=1 to enable EPROM
                          programming voltage
          JSR    DLYEP    Delay 2-4 ms
          CLR    $1036    Turn off programming voltage and set to READ mode
```

If the MCU is operating with programming voltage present on the $\overline{\text{XIRQ}}$/V_{PPE} pin, the $\overline{\text{IRQ}}/\text{CE}$ pin must be pulled to a high level before the address and data are changed to program the next location.

4.3.2 Programming EPROM with Downloaded Data

When using this method, the EPROM is programmed by software while in the special test or bootstrap modes. User-developed software can be uploaded through the SCI, or a ROM-resident EPROM programming utility can be used.The 12 volt nominal programming voltage must be present on the $\overline{\text{XIRQ}}$/V_{PPE} pin. To use the resident utility, bootload a three-byte program consisting of a single jump instruction to $BF00. $BF00 is the starting address of a resident EPROM programming utility. The utility program sets the X and Y index registers to default values, then receives programming data from an external host and puts it in EPROM. The value in IX determines programming delay time. The value in IY is a pointer to the first address in EPROM to be programmed (default = $D000).

When the utility program is ready to receive programming data, it sends the host the $FF character. Then it waits. When the host sees the $FF character, the EPROM programming data is sent, starting with the first location in the EPROM array. After the last byte to be programmed is sent and the corresponding verification data is returned, the programming operation is terminated by resetting the MCU.

4.3.3 EPROM Programming Control Register

The PPROG register enables the EPROM programming voltage and controls the latching of data to be programmed. For MC68HC711E9, PPROG is also the EEPROM programming control register. In the MC68HC711E20 EPROM programming is controlled by the EPROG register and EEPROM programming is controlled by the PPROG register.

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

	Bit 7	6	5	4	3	2	1	Bit 0
	ODD	EVEN	ELAT[1]	BYTE	ROW	ERASE	EELAT	EPGM
RESET:	0	0	0	0	0	0	0	0

NOTES:
1. MC68HC711E9 only.

ODD — Program Odd Rows in Half of EEPROM (Test)
Refer to **4.4 EEPROM**.

EVEN — Program Even Rows in Half of EEPROM (Test)
Refer to **4.4 EEPROM**.

ELAT — EPROM/OTPROM Latch Control
When ELAT = 1, writes to EPROM cause address and data to be latched and the EPROM/OTPROM cannot be read. ELAT can be read any time. ELAT can be written any time except when EPGM = 1; then the write to ELAT is disabled. For the MC68HC711E9, EPGM enables the high voltage necessary for both EEPROM and EPROM/OTPROM programming. For the MC68HC711E9, ELAT and EELAT are mutually exclusive and cannot both equal one.
0 = EPROM address and data bus configured for normal reads
1 = EPROM address and data bus configured for programming

Byte — Byte/Other EEPROM Erase Mode
Refer to **4.4 EEPROM**.

ROW — Row/All EEPROM Erase Mode
Refer to **4.4 EEPROM**.

ERASE — Erase Mode Select
Refer to **4.4 EEPROM**.

EELAT — EEPROM Latch Control
Refer to **4.4 EEPROM**.

EPGM — EPROM/OTPROM/EEPROM Programming Voltage Enable
0 = Programming voltage to EPROM/OTPROM/EEPROM array disconnected
1 = Programming voltage to EPROM/OTPROM/EEPROM array connected
EPGM can be read any time and can only be written when ELAT = 1 (for EPROM/OTPROM programming) or when EELAT = 1 (for EEPROM programming).

EPROG — EPROM Programming Control (MC68HC711E20 only) **$1036**

	Bit 7	6	5	4	3	2	1	Bit 0
	MBE	—	ELAT	EXCOL	EXROW	T1	T0	PGM
RESET:	0	0	0	0	0	0	0	0

MBE — Multiple-Byte Programming Enable
When multiple-byte programming is enabled, address bit 5 is considered a don't care so that bytes with address bit 5 = 0 and address bit 5 = 1 both get programmed. MBE can be read in any mode and always reads zero in normal modes. MBE can only be written in special modes.
0 = EPROM array configured for normal programming
1 = Program two bytes with the same data

Bit 6 — Not Implemented
Always reads zero

ELAT — EPROM/OTPROM Latch Control
When ELAT = 1, writes to EPROM cause address and data to be latched and the EPROM/OTPROM cannot be read. ELAT can be read any time. ELAT can be written any time except when PGM = 1; then the write to ELAT is disabled.
0 = EPROM/OTPROM address and data bus configured for normal reads
1 = EPROM/OTPROM address and data bus configured for programming

EXCOL — Select Extra Columns

0 = User array selected

1 = User array is disabled and extra columns are accessed at bits [7:0]. Addresses use bits [13:5] and bits [4:0] are don't care. EXCOL can only be read in special modes and always returns zero in normal modes. EXCOL can be written in special modes only.

EXROW — Select Extra Rows

0 = User array selected

1 = User array is disabled and two extra rows are available. Addresses use bits [7:0] and bits [13:8] are don't care. EXROW can only be read in special modes and always returns zero in normal modes. EXROW can be written in special modes only.

T[1:0] — EPROM Test Mode Select Bits

T1	T0	Function Selected
0	0	Normal Mode
0	1	Reserved
1	0	Gate Stress
1	1	Drain Stress

These bits allow selection of either gate stress or drain stress test modes. They can be read only in special modes and always read zero in normal modes. T[1:0] can only be written in special modes.

PGM — EPROM Programming Voltage Enable

0 = Programming voltage to EPROM array disconnected

1 = Programming voltage to EPROM array connected

PGM can be read any time and can only be written when ELAT = 1.

4.4 EEPROM

Some E-series devices contain 512 bytes of on-chip EEPROM. The MC68HC811E2 contains 2048 bytes of EEPROM with selectable base address.

4.4.1 EEPROM Programming

The erased state of an EEPROM bit is one. During a read operation, bit lines are precharged to one. The floating gate devices of programmed bits conduct and pull the bit lines to zero. Unprogrammed bits remain at the precharged level and are read as ones. Programming a bit to one causes no change. Programming a bit to zero changes the bit so that subsequent reads return zero.

When appropriate bits in the BPROT register are cleared, the PPROG register controls programming and erasing the EEPROM. The PPROG register can be read or written at any time, but logic enforces defined programming and erasing sequences to prevent unintentional changes to EEPROM data. When the EELAT bit in the PPROG register is cleared, the EEPROM can be read as if it were a ROM.

The on-chip charge pump that generates the EEPROM programming voltage from V_{DD} uses MOS capacitors, which are relatively small in value. The efficiency of this charge pump and its drive capability are affected by the level of V_{DD} and the frequency of the driving clock. The load depends on the number of bits being programmed or erased and capacitances in the EEPROM array.

The clock source driving the charge pump is software selectable. When the clock select (CSEL) bit in the OPTION register is zero, the E clock is used; when CSEL is one, an on-chip resistor-capacitor (RC) oscillator is used.

The EEPROM programming voltage power supply voltage to the EEPROM array is not enabled until there has been a write to PPROG with EELAT set and PGM cleared. This must be followed by a write to a valid EEPROM location or to the CONFIG address, and then a write to PPROG with both the EELAT and EPGM bits set. Any attempt to set both EELAT and EPGM during the same write operation results in neither bit being set.

4.4.1.1 BPROT Register

This register prevents inadvertent writes to both the CONFIG register and EEPROM. The active bits in this register are initialized to one out of reset and can only be cleared during the first 64 E-clock cycles after reset in the normal modes. When these bits are cleared, the associated EEPROM section and the CONFIG register can be programmed or erased. EEPROM is only visible if the EEON bit in the CONFIG register is set. The bits in the BPROT register can be written to one at any time to protect EEPROM and the CONFIG register. In test or bootstrap modes, write protection is inhibited and BPROT can be written repeatedly. Address ranges for protected areas of EEPROM differ significantly for the MC68HC811E2. Refer to the following description.

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 CONFIG Register
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
When set, these bits protect a block of EEPROM from being programmed or electronically erased. Ultraviolet light, however, can erase the entire EEPROM contents regardless of BPRT[3:0] (windowed packages only). Refer to **Table 4–7** and **Table 4–8**. When cleared, BPRT[3:0] allow programming and erasure of the associated block.

Table 4–7 EEPROM Block Protect

Bit Name	Block Protected	Block Size
BPRT0	$B600 – $B61F	32 Bytes
BPRT1	$B620 – $B65F	64 Bytes
BPRT2	$B660 – $B6DF	128 Bytes
BPRT3	$B6E0 – $B7FF	288 Bytes

Table 4–8 EEPROM Block Protect (MC68HC811E2)

Bit Name	Block Protected	Block Size
BPRT0	$x800 – $x9FF*	512 Bytes
BPRT1	$xA00 – $xBFF*	512 Bytes
BPRT2	$xC00 – $xDFF*	512 Bytes
BPRT3	$xE00 – $xFFF*	512 Bytes

*x is determined by the value of EE[3:0] in CONFIG. Refer to the CONFIG register description for MC68HC811E2.

4.4.1.2 PPROG Register

The PPROG register selects and controls the EEPROM programming function. Bits in PPROG enable the programming voltage, control the latching of data to be programmed, and select the method of erasure (byte, row, etc.).

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

	Bit 7	6	5	4	3	2	1	Bit 0
	ODD	EVEN	ELAT[1]	BYTE	ROW	ERASE	EELAT	EPGM
RESET:	0	0	0	0	0	0	0	0

NOTES:
1. MC68HC711E9 only.

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

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

ELAT — EPROM/OTPROM Latch Control

For the MC68HC711E9, EPGM enables the high voltage necessary for both EPROM/OTPROM and EEPROM programming. For MC68HC711E9, ELAT and EELAT are mutually exclusive and cannot both equal one.

0 = EPROM address and data bus configured for normal reads
1 = EPROM address and data bus configured for programming

Byte — Byte/Other EEPROM Erase Mode

This bit overrides the ROW bit.

0 = Row or bulk erase
1 = Erase only one byte

ROW — Row/All EEPROM Erase Mode

If BYTE is one, ROW has no meaning.

0 = Bulk erase

1 = Row erase

Table 4–9 EEPROM Erase

BYTE	ROW	Action
0	0	Bulk Erase (Entire array)
0	1	Row Erase (16 Bytes)
1	0	Byte Erase
1	1	Byte Erase

ERASE — Erase Mode Select

0 = Normal read or program mode

1 = Erase mode

EELAT — EEPROM Latch Control

0 = EEPROM address and data bus configured for normal reads and cannot be programmed

1 = EEPROM address and data bus configured for programming or erasing and cannot be read

EPGM — EPROM/OTPROM/EEPROM Programming Voltage Enable

0 = Programming voltage to EEPROM array switched off

1 = Programming voltage to EEPROM array switched on

During EEPROM programming, the ROW and BYTE bits of PPROG are not used. If the frequency of the E clock is 1 MHz or less, set the CSEL bit in the OPTION register. Recall that zeros must be erased by a separate erase operation before programming. The following examples of how to program an EEPROM byte assume that the appropriate bits in BPROT are cleared.

```
PROG     LDAB  #$02     EELAT = 1
         STAB  $103B    Set EELAT bit
         STAA  $XXXX    Store data to EEPROM address (for valid EEPROM
                        address see memory map for each device)
         LDAB  #$03     EELAT = 1, EPGM = 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 Bulk Erase

The following is an example of how to bulk erase the entire EEPROM. The CONFIG register is not affected in this example.

```
BULKE    LDAB  #$06     EELAT = 1
         STAB  $103B    Set EELAT bit
         STAA  $XXXX    Store data to any EEPROM address (for valid EEPROM
                        address see memory map for each device)
         LDAB  #$07     EELAT = 1, EPGM = 1
         STAB  $103B    Turn on programming voltage
         JSR   DLY10    Delay 10 ms
         CLR   $103B    Turn off high voltage and set to READ mode
```

4.4.1.4 EEPROM Row Erase

The following example shows how to perform a fast erase of large sections of EEPROM.

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

4.4.1.5 EEPROM Byte Erase

The following is an example of how to erase a single byte of EEPROM.

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

4.4.1.6 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.

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

4.4.2 EEPROM Security

The optional security feature, available only on ROM-based MCUs, protects the EEPROM and RAM contents from unauthorized access. A program, or a key portion of a program, can be protected against unauthorized duplication. To accomplish this, the protection mechanism restricts operation of protected devices to the single-chip modes. This prevents the memory locations from being monitored externally because single-chip modes do not allow visibility of the internal address and data buses. Resident programs, however, have unlimited access to the internal EEPROM and RAM and can read, write, or transfer the contents of these memories.

An enhanced security feature which protects EPROM contents, RAM, and EEPROM from unauthorized accesses is available in MC68S711E9. Refer to **APPENDIX B MECHANICAL DATA AND ORDERING INFORMATION** for the exact part number.

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 immediately stops execution of the current instruction and forces the program counter to a known starting address. Internal registers and control bits are initialized so the MCU can resume executing instructions. 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. Most 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. Do not 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 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 timeout period. The system E clock is divided by 2^{15} and then further scaled by a factor shown in **Table 5–1**. After reset, these bits are zero, which selects the fastest timeout period. In normal operating modes, these bits can only be written once within 64 bus cycles after reset.

Table 5–1 COP Timer Rate Select

CR[1:0]	Divide E/2^{15} By	XTAL = 4.0 MHz Timeout – 0 ms, + 32.8 ms	XTAL = 8.0 MHz Timeout – 0 ms, + 16.4 ms	XTAL = 12.0 MHz Timeout – 0 ms, + 10.9 ms
0 0	1	32.768 ms	16.384 ms	10.923 ms
0 1	4	131.072 ms	65.536 ms	43.691 ms
1 0	16	524.28 ms	262.14 ms	174.76 ms
1 1	64	2.098 s	1.049 s	699.05 ms
	E =	1.0 MHz	2.0 MHz	3.0 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 control bit in the OPTION register. The presence of a timeout 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 timeout 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 the CME bit in the OPTION register to zero to disable the clock monitor. After recovery from STOP, set the CME bit to logic one to enable the clock monitor. Alternatively, executing a STOP instruction with the CME bit set to logic one can be used as a software initiated reset.

5.1.5 Option Register

OPTION — System Configuration Options **$1039**

	Bit 7	6	5	4	3	2	1	Bit 0
	ADPU	CSEL	IRQE[1]	DLY[1]	CME	—	CR1[1]	CR0[1]
RESET:	0	0	0	1	0	0	0	0

NOTES:
1. 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 = $\overline{\text{IRQ}}$ is configured for level-sensitive operation
1 = $\overline{\text{IRQ}}$ is configured for edge-sensitive only operation

DLY — Enable Oscillator Startup Delay
Refer to **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY** and **SECTION 10 ANALOG-TO-DIGITAL CONVERTER**.

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.

0 = Clock monitor circuit disabled
1 = Slow or stopped clocks cause reset

Bit 2 — Not implemented

Always reads zero

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.

5.1.6 CONFIG Register

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

	Bit 7	6	5	4	3	2	1	Bit 0
	EE3*	EE2*	EE1*	EE0*	NOSEC	NOCOP	ROMON	EEON
RESET:	0	0	0	0	1	1	1	1

EE[3:0] — EEPROM Mapping*

EE[3:0] apply only to MC68HC811E2. Refer to **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY**.

NOSEC — Security Mode Disable

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

NOCOP — COP System Disable

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

ROMON — ROM (EPROM) Enable

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

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, Reset Vector, and Operating Mode

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 Timeout	$FFFA, FFFB	$BFFA, $BFFB

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

5.2.1 Central Processing Unit

After reset, the CPU fetches the restart 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, mapping the RAM at $00, and the control registers at $1000. For the MC68HC811E2, the CONFIG register resets to $FF. EEPROM mapping bits (EE[3:0]) place the EEPROM at $F800. Refer to the memory map diagram for MC68HC811E2 in **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY**.

5.2.3 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.

5.2.4 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.

5.2.5 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.

5.2.6 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 timeout.

5.2.7 Serial Communications Interface (SCI)

The reset condition of the SCI system is independent of the operating mode. At reset, the SCI baud rate control register (BAUD) is initialized to $04. 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 (SCSR) are both ones, 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 in the SCI control register 2 (SCCR2) are cleared.

5.2.8 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.

5.2.9 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 indeterminate.

5.2.10 System

The EEPROM programming controls are disabled, so the memory system is configured for normal read operation. PSEL[3:0] are initialized with the value %0110, 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. MODA and MODB inputs select one of the four operating modes. After reset, writing SMOD and MDA in special modes causes the MCU to change operating modes. Refer to the description of HPRIO register in **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY** for a detailed description of SMOD and MDA. 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 is 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 5–3**)

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, HPRIO 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[1]	SMOD[1]	MDA[1]	IRVNE	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	0	0	1	1	0	Bootstrap
	0	1	1	1	0	1	1	0	Special Test

NOTES:

1. The values of the RBOOT, SMOD, and MDA reset bits depend on the mode during power-up. Refer to **Table 4–2**.

RBOOT — Read Bootstrap ROM

Has meaning only when the SMOD bit is a one (bootstrap mode or special test mode). At all other times this bit is clear and cannot be written. Refer to **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY** for more information.

SMOD — Special Mode Select

This bit reflects the inverse of the MODB input pin at the rising edge of reset. Refer to **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY** for more information.

MDA — Mode Select A

The mode select A bit reflects the status of the MODA input pin at the rising edge of reset. Refer to **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY** for more information.

IRVNE — Internal Read Visibility/Not E

The IRVNE control bit allows internal read accesses to be available on the external data bus during operation in expanded modes. In single-chip and bootstrap modes, IRVNE determines whether the E clock is driven out an external pin. For the MC68HC811E2, this bit is IRV and only controls internal read visibility. Refer to **SECTION 4 OPERATING MODES AND ON-CHIP MEMORY** for more information.

PSEL[3:0] — Priority Select Bits

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

Table 5–3 Highest Priority Interrupt Selection

PSEL[3:0]	Interrupt Source Promoted
0 0 0 0	Timer Overflow
0 0 0 1	Pulse Accumulator Overflow
0 0 1 0	Pulse Accumulator Input Edge
0 0 1 1	SPI Serial Transfer Complete
0 1 0 0	SCI Serial System
0 1 0 1	Reserved (Default to $\overline{\text{IRQ}}$)
0 1 1 0	$\overline{\text{IRQ}}$ (External Pin or Parallel I/O)
0 1 1 1	Real-Time Interrupt
1 0 0 0	Timer Input Capture 1
1 0 0 1	Timer Input Capture 2
1 0 1 0	Timer Input Capture 3
1 0 1 1	Timer Output Compare 1
1 1 0 0	Timer Output Compare 2
1 1 0 1	Timer Output Compare 3
1 1 1 0	Timer Output Compare 4
1 1 1 1	Timer Input Capture 4/Output Compare 5

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 nonmaskable 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 • SCI Receive Data Register Full • SCI Receiver Overrun • SCI Transmit Data Register Empty • SCI Transmit Complete • SCI Idle Line Detect	I	 RIE RIE TIE TCIE ILIE
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	$\overline{\text{IRQ}}$ (External Pin)	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	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 Nonmaskable Interrupt Request ($\overline{\text{XIRQ}}$)

Nonmaskable 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 $\overline{\text{NMI}}$ (nonmaskable interrupt) 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 unmasked. 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 5–2** 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 5–2** is an expansion of a block in **Figure 5–1** and illustrates interrupt priorities. **Figure 5–3** shows the resolution of interrupt sources within the SCI subsystem.

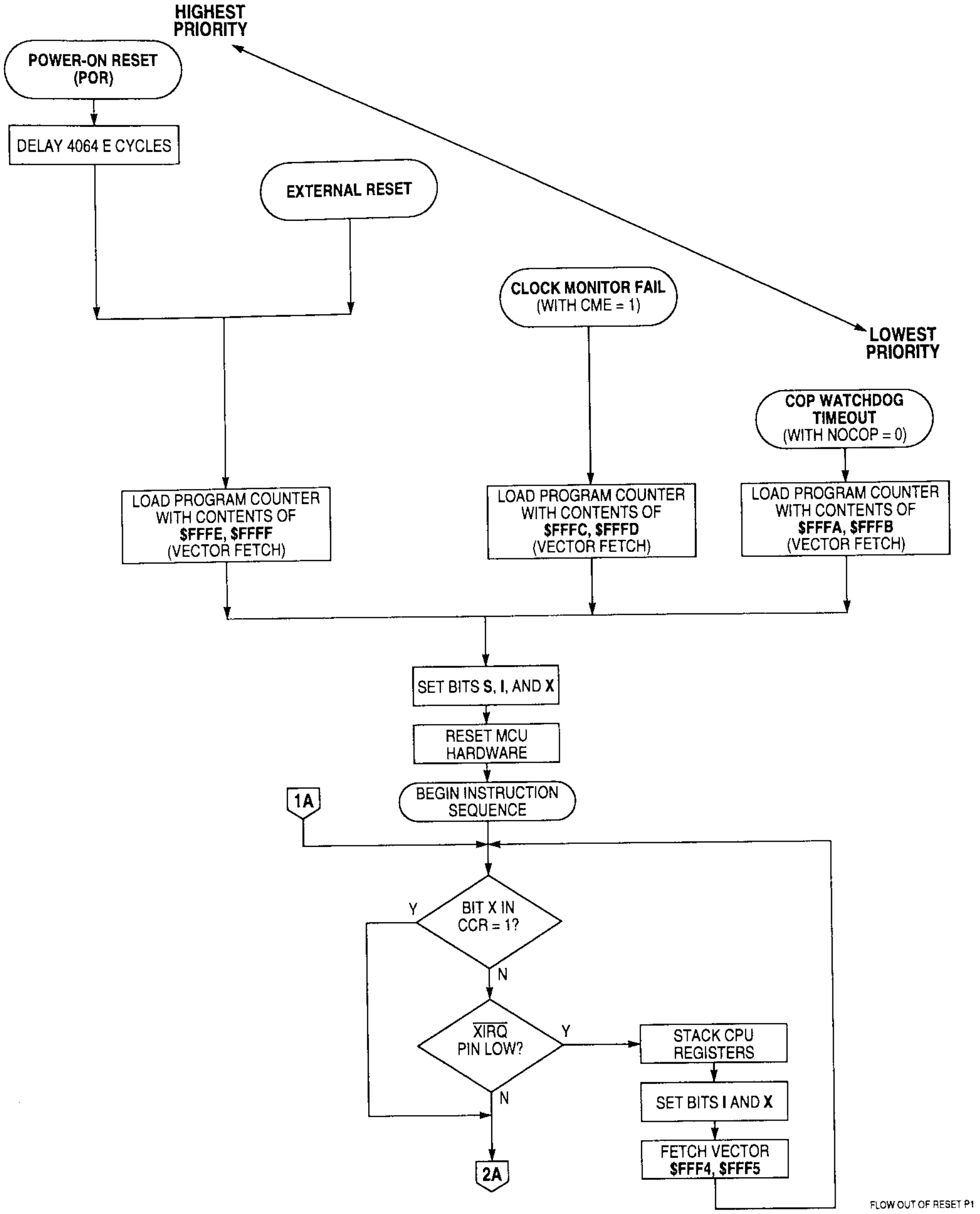

HIGHEST PRIORITY
POWER-ON RESET (POR)
DELAY 4064 E CYCLES
EXTERNAL RESET
CLOCK MONITOR FAIL (WITH CME = 1)
LOWEST PRIORITY
COP WATCHDOG TIMEOUT (WITH NOCOP = 0)
LOAD PROGRAM COUNTER WITH CONTENTS OF $FFFE, $FFFF (VECTOR FETCH)
LOAD PROGRAM COUNTER WITH CONTENTS OF $FFFC, $FFFD (VECTOR FETCH)
LOAD PROGRAM COUNTER WITH CONTENTS OF $FFFA, $FFFB (VECTOR FETCH)
SET BITS S, I, AND X
RESET MCU HARDWARE
1A
BEGIN INSTRUCTION SEQUENCE
BIT X IN CCR = 1?
Y
N
XIRQ PIN LOW?
Y
N
STACK CPU REGISTERS
SET BITS I AND X
FETCH VECTOR $FFF4, $FFF5
2A
FLOW OUT OF RESET P1

Figure 5–1 Processing Flow out of Reset (1 of 2)

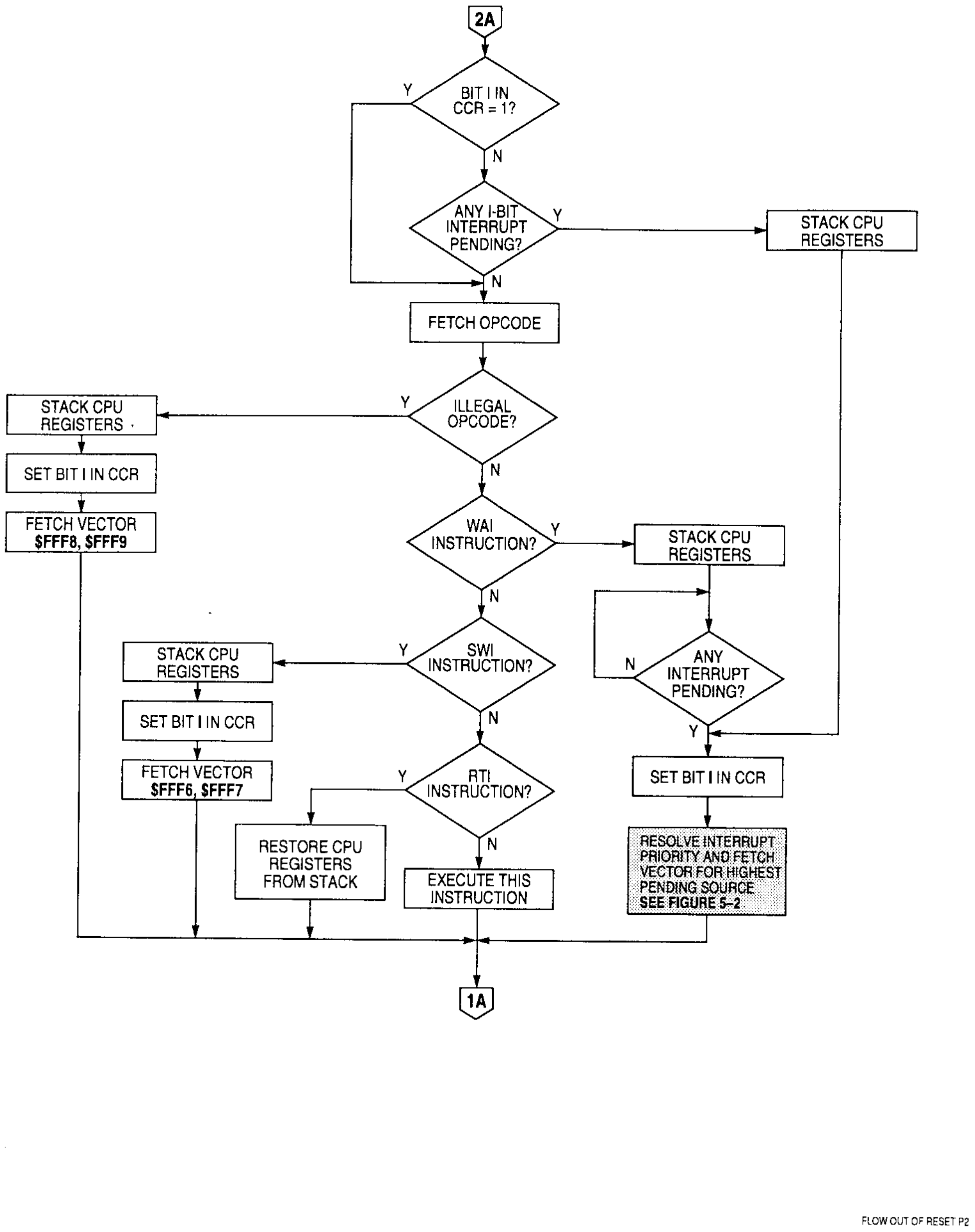


Figure 5–1 Processing Flow out of Reset (2 of 2)

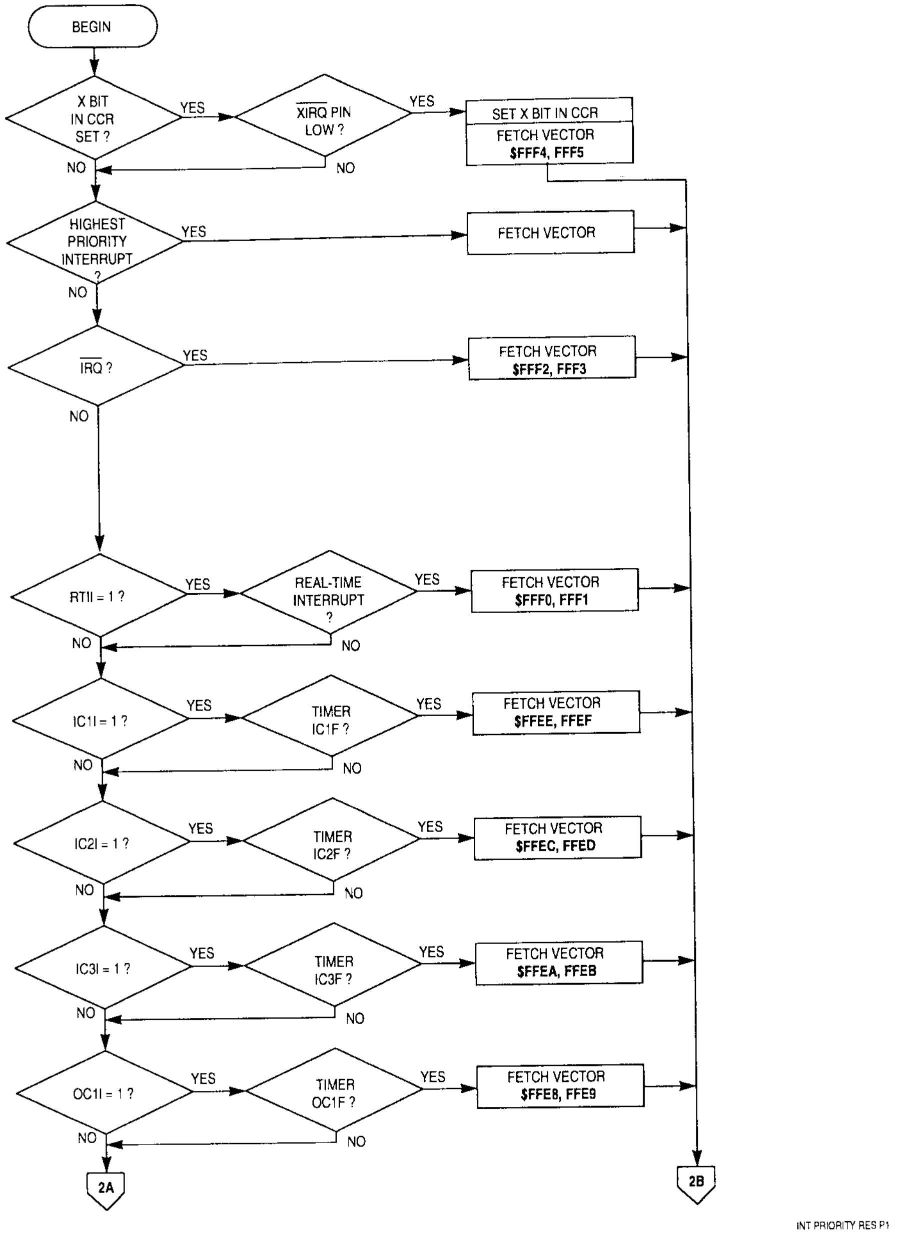


Figure 5–2 Interrupt Priority Resolution (1 of 2)

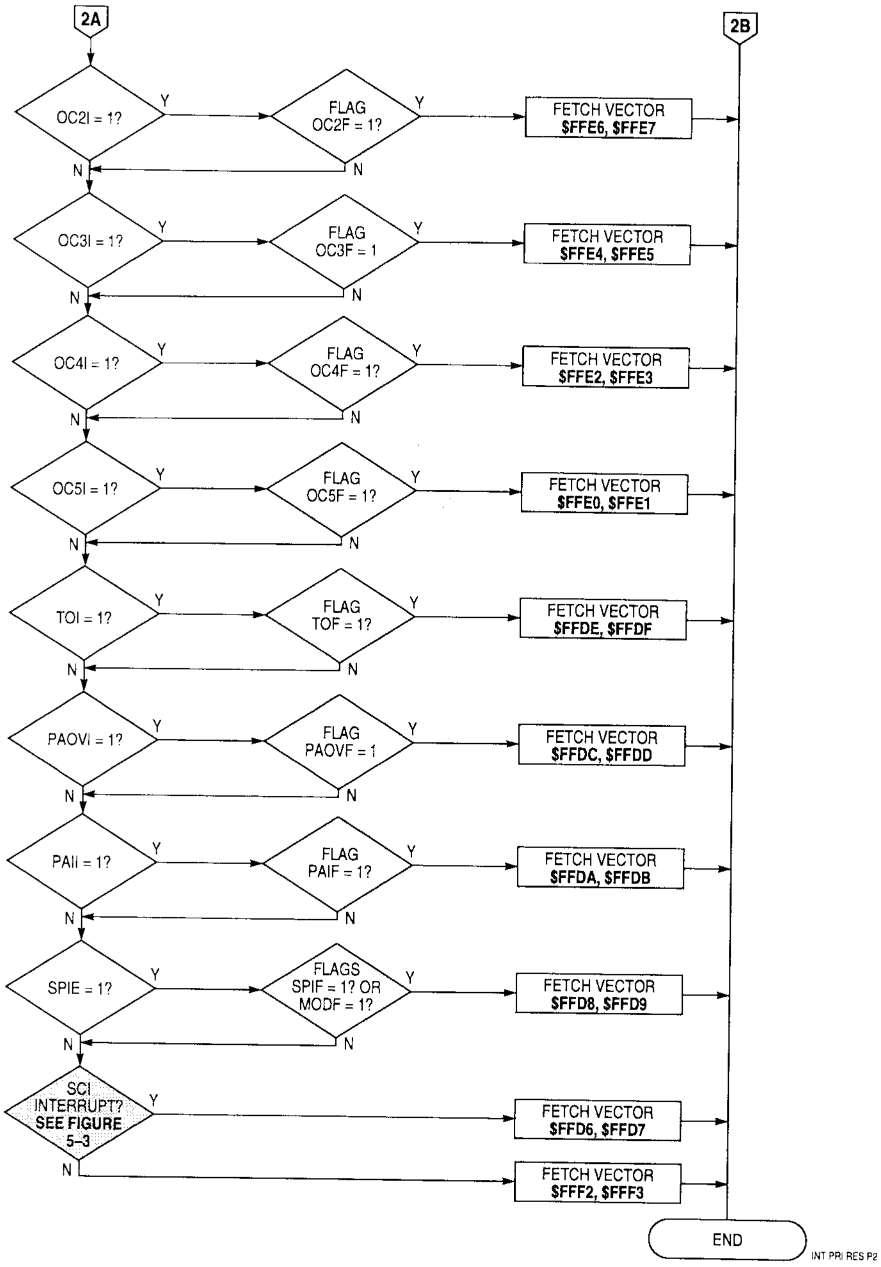

2A
2B
OC2I = 1?
FLAG OC2F = 1?
FETCH VECTOR $FFE6, $FFE7
OC3I = 1?
FLAG OC3F = 1
FETCH VECTOR $FFE4, $FFE5
OC4I = 1?
FLAG OC4F = 1?
FETCH VECTOR $FFE2, $FFE3
OC5I = 1?
FLAG OC5F = 1?
FETCH VECTOR $FFE0, $FFE1
TOI = 1?
FLAG TOF = 1?
FETCH VECTOR $FFDE, $FFDF
PAOVI = 1?
FLAG PAOVF = 1
FETCH VECTOR $FFDC, $FFDD
PAII = 1?
FLAG PAIF = 1?
FETCH VECTOR $FFDA, $FFDB
SPIE = 1?
FLAGS SPIF = 1? OR MODF = 1?
FETCH VECTOR $FFD8, $FFD9
SCI INTERRUPT? SEE FIGURE 5–3
FETCH VECTOR $FFD6, $FFD7
FETCH VECTOR $FFF2, $FFF3
Y
N
END
INT PRI RES P2

Figure 5–2 Interrupt Priority Resolution (2 of 2)

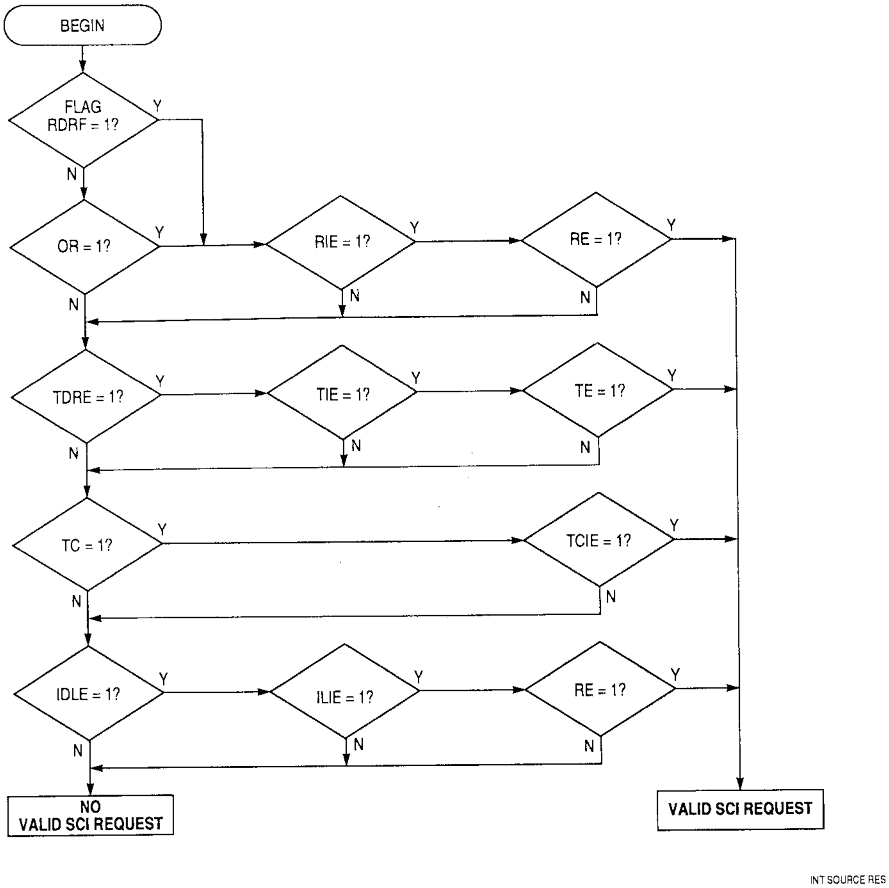


Figure 5–3 Interrupt Source Resolution Within SCI

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 the entire RAM array.

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 $\overline{\text{IRQ}}$, an $\overline{\text{XIRQ}}$, 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. 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 ($\overline{\text{IRQ}}$ or $\overline{\text{XIRQ}}$) or to the $\overline{\text{RESET}}$ pin. A pending edge-triggered $\overline{\text{IRQ}}$ 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 comes to restart 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 in which all I/O pins and functions are also restored to their initial states.

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

All M68HC11 E-series MCUs have five I/O ports and up to 38 input/output lines, depending on the operating mode. Refer to **Table 6–1**, which is a summary of the ports and their shared functions.

Table 6–1 Input/Output Ports

Port	Input Pins	Output Pins	Bidirectional Pins	Shared Functions
Port A	3	3	2	Timer
Port B	—	8	—	High Order Address
Port C	—	—	8	Low Order Address and Data Bus
Port D	—	—	6	SCI and SPI
Port E	8	—	—	A/D Converter

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, which has three input-only pins, three output-only pins, and two bidirectional I/O pins, 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	0	0	0	I	I	I	I
Alt. Func:	PAI	OC2	OC3	OC4	IC4/OC5	IC1	IC2	IC3
And/or:	OC1	OC1	OC1	OC1	OC1	—	—	—

PACTL — Pulse Accumulator Control **$1026**

	Bit 7	6	5	4	3	2	1	Bit 0
	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0
RESET:	0	0	0	0	0	0	0	0

DDRA7 — Data Direction for Port A Bit 7

Overridden if an output compare function is configured to control the PA7 pin

0 = Input

1 = Output

The pulse accumulator uses port A bit 7 as the PAI input, but the pin can also be used as general-purpose I/O or as an output compare. Note that even when port A bit 7 is configured as an output, the pin still drives the input to the pulse accumulator.

PAEN — Pulse Accumulator System Enable

Refer to **SECTION 9 TIMING SYSTEM**.

PAMOD — Pulse Accumulator Mode

Refer to **SECTION 9 TIMING SYSTEM**.

PEDGE — Pulse Accumulator Edge Control

Refer to **SECTION 9 TIMING SYSTEM**.

DDRA3 — Data Direction for Port A Bit 3

Overridden if an output compare function is configured to control the PA3 pin.

0 = Input

1 = Output

I4/O5 — Input Capture 4/Output Compare 5

Refer to **SECTION 9 TIMING SYSTEM**.

RTR[1:0] — RTI Interrupt Rate Select

Refer to **SECTION 9 TIMING SYSTEM**.

6.2 Port B

In single-chip or bootstrap modes, port B pins are general-purpose outputs. In expanded or special test modes, port B pins are high order address outputs.

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

In single-chip and bootstrap modes, port C pins reset to high impedance inputs (DDRC bits are set to zero). In expanded and special test modes, port C pins are multiplexed address/data bus and the port C register address is treated as an external memory location.

PORTC — Port C Data **$1003**

	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:	ADDR7 DATA7	ADDR6 DATA6	ADDR5 DATA5	ADDR4 DATA4	ADDR3 DATA3	ADDR2 DATA2	ADDR1 DATA1	ADDR0 DATA0

PORTCL — Port C Latched **$1005**

	Bit 7	6	5	4	3	2	1	Bit 0
	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0
RESET:	I	I	I	I	I	I	I	I

PORTCL is used in the handshake clearing mechanism. When an active edge occurs on the STRA pin, port C data is latched into the PORTCL register. Reads of this register return the last value latched into PORTCL and clear STAF flag (following a read of PIOC with STAF set).

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

In handshake output mode, DDRC bits select the three-stated output option (DDCx = 1).

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 PD[5:0] are configured as high impedance inputs (DDRD bits cleared).

PORTD — Port D Data **$1008**

	Bit 7	6	5	4	3	2	1	Bit 0
	0	0	PD5	PD4	PD3	PD2	PD1	PD0
RESET:	—	—	I	I	I	I	I	I
Alt. Func.:	—	—	PD5 $\overline{\text{SS}}$	PD4 SCK	PD3 MOSI	PD2 MISO	PD1 Tx	PD0 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
When DDRD bit 5 is one and MSTR = 1 in SPCR, PD5/$\overline{\text{SS}}$ is a general-purpose output and mode fault logic is disabled.
0 = Input
1 = Output

6.5 Port E

Port E is used for general-purpose static inputs or pins that share 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. Func.:	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0

6.6 Handshake Protocol

Simple and full handshake input and output functions are available on ports B and C pins in single-chip mode. In simple strobed mode, port B is a strobed output port and port C is a latching input port. The two activities are available simultaneously.

The STRB output is pulsed for two E-clock periods each time there is a write to the PORTB register. The INVB bit in the PIOC register controls the polarity of STRB pulses. Port C levels are latched into the alternate port C latch (PORTCL) register on each assertion of the STRA input. STRA edge select, flag, and interrupt enable bits are located in the PIOC register. Any or all of the port C lines can still be used as general-purpose I/O while in strobed input mode.

Full handshake modes use port C pins and the STRA and STRB lines. Input and output handshake modes are supported, and output handshake mode has a three-stated variation. STRA is an edge-detecting input and STRB is a handshake output. Control and enable bits are located in the PIOC register.

In full input handshake mode, the MCU asserts STRB to signal an external system that it is ready to latch data. Port C logic levels are latched into PORTCL when the STRA line is asserted by the external system. The MCU then negates STRB. The MCU reasserts STRB after the PORTCL register is read. In this mode, a mix of latched inputs, static inputs, and static outputs is allowed on port C, differentiated by the data direction bits and use of the PORTC and PORTCL registers.

In full output handshake mode, the MCU writes data to PORTCL which, in turn, asserts the STRB output to indicate that data is ready. The external system reads port C data and asserts the STRA input to acknowledge that data has been received.

In the three-state variation of output handshake mode, lines intended as three-state handshake outputs are configured as inputs by clearing the corresponding DDRC bits. The MCU writes data to PORTCL and asserts STRB. The external system responds by activating the STRA input, which forces the MCU to drive the data in PORTC out on all of the port C lines. After the trailing edge of the active signal on STRA, the MCU negates the STRB signal. The three-state mode variation does not allow part of port C to be used for static inputs while other port C pins are being used for handshake outputs. Refer to the PIOC register description for further information.

6.7 Parallel I/O Control Register

The parallel handshake functions are available only in the single-chip operating mode. PIOC is a read/write register except for bit 7, which is read only. **Table 6–2** shows a summary of handshake operations.

PIOC — Parallel I/O Control **$1002**

	Bit 7	6	5	4	3	2	1	Bit 0
	STAF	STAI	CWOM	HNDS	OIN	PLS	EGA	INVB
RESET:	0	0	0	0	0	U	1	1

STAF — Strobe A Interrupt Status Flag

0 = No edge on strobe A

1 = Selected edge on strobe A

STAF is set when the selected edge occurs on strobe A. This bit can be cleared by a read of PIOC with STAF set followed by a read of PORTCL (simple strobed or full input handshake mode) or a write to PORTCL (output handshake mode).

STAI — Strobe A Interrupt Enable Mask

0 = STAF does not request interrupt
1 = STAF requests interrupt

CWOM — Port C Wired-OR Mode (affects all eight port C pins)

It is customary to have an external pull-up resistor on lines that are driven by open-drain devices.

0 = Port C outputs are normal CMOS outputs
1 = Port C outputs are open-drain outputs

HNDS — Handshake Mode

0 = Simple strobe mode
1 = Full input or output handshake mode

OIN — Output or Input Handshake Select

HNDS must be set to one for this bit to have meaning.

0 = Input handshake
1 = Output handshake

PLS — Pulsed/Interlocked Handshake Operation

HNDS must be set to one for this bit to have meaning. When interlocked handshake is selected, strobe B is active until the selected edge of strobe A is detected.

0 = Interlocked handshake
1 = Pulsed handshake (strobe B pulses high for two E-clock cycles.)

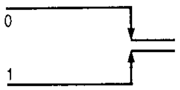
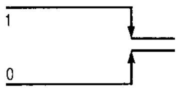
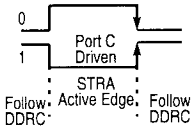
EGA — Active Edge for Strobe A

0 = STRA falling edge selected, high level activates port C outputs (output handshake)
1 = STRA rising edge selected, low level activates port C outputs (output handshake)

INVB — Invert Strobe B

0 = Active level is logic zero
1 = Active level is logic one

Table 6–2 Parallel I/O Control

	STAF Clearing Sequence	HNDS	OIN	PLS	EGA	Port B	Port C
Simple Strobed Mode	Read PIOC with STAF = 1 then read PORTCL	0	X	X		Inputs latched into PORTCL on any active edge on STRA	STRB pulses on writes to PORTB
Full input Handshake mode	Read PIOC with STAF = 1 then read PORTCL	1	0	0 = STRB active level 1 = STRB active pulse		Inputs latched into PORTCL on any active edge on STRA	Normal output port, unaffected in handshake modes
Full output handshake mode	read PIOC with STAF = 1 then write PORTCL	1	1	0 = STRB active level 1 = STRB active pulse	 0 1 Port C Driven STRA Active Edge Follow DDRC Follow DDRC	Driven as outputs if STRA at active level; follows DDRC if STRA not at active level	Normal output port, unaffected in handshake modes

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 M68HC11 E series of microcontrollers. It has a standard nonreturn to zero (NRZ) format (one start, eight or nine data, 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.

All members of the E series contain the same SCI, with one exception. The SCI system in the MC68HC11E20 and MC68HC711E20 MCUs has an enhanced SCI baud rate generator. A divide-by-39 stage has been added that is enabled by an extra bit in the BAUD register. This increases the available SCI baud rate selections. Refer to the baud rate generator diagrams and the BAUD register description.

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.

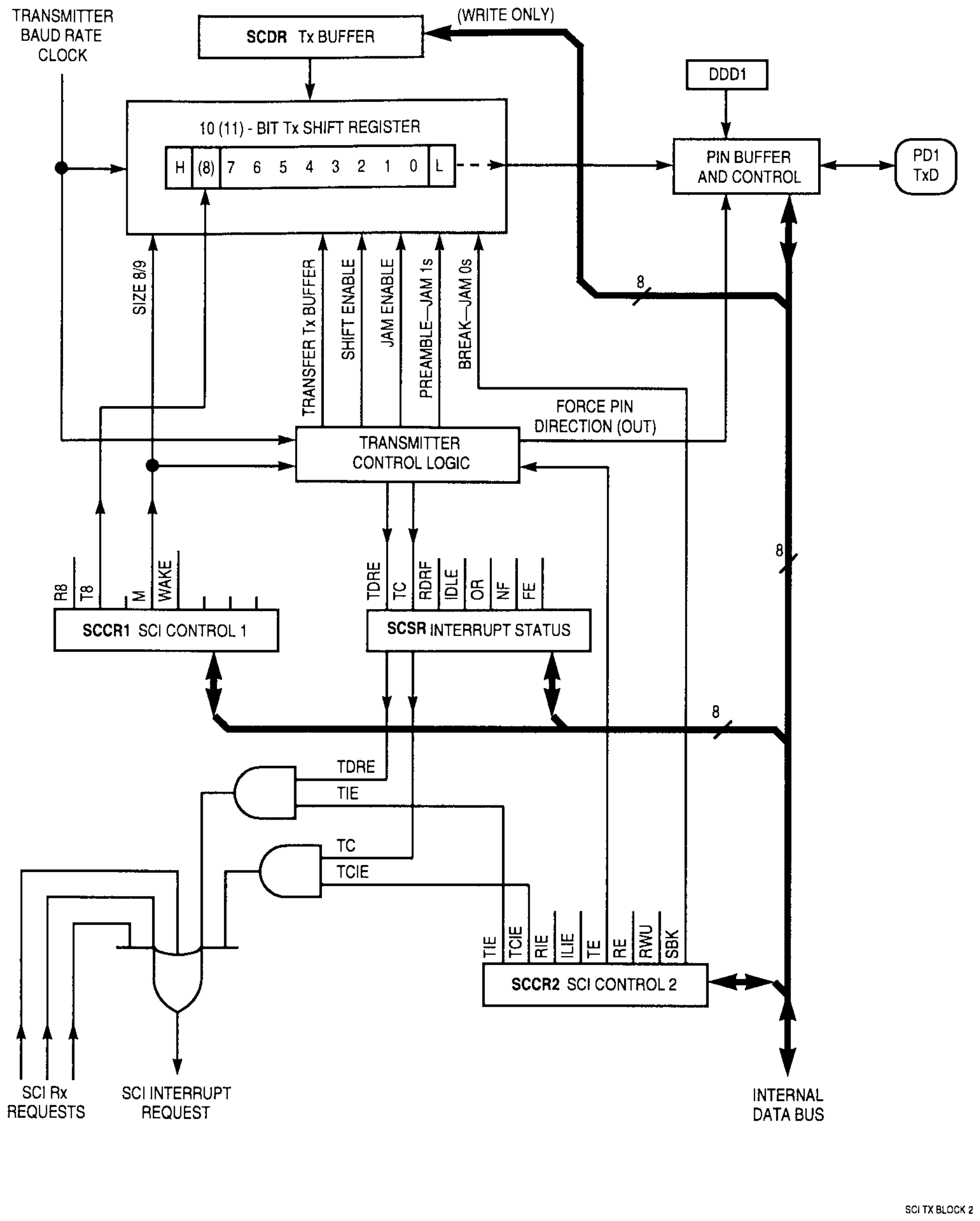


Figure 7–1 SCI Transmitter Block Diagram

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.

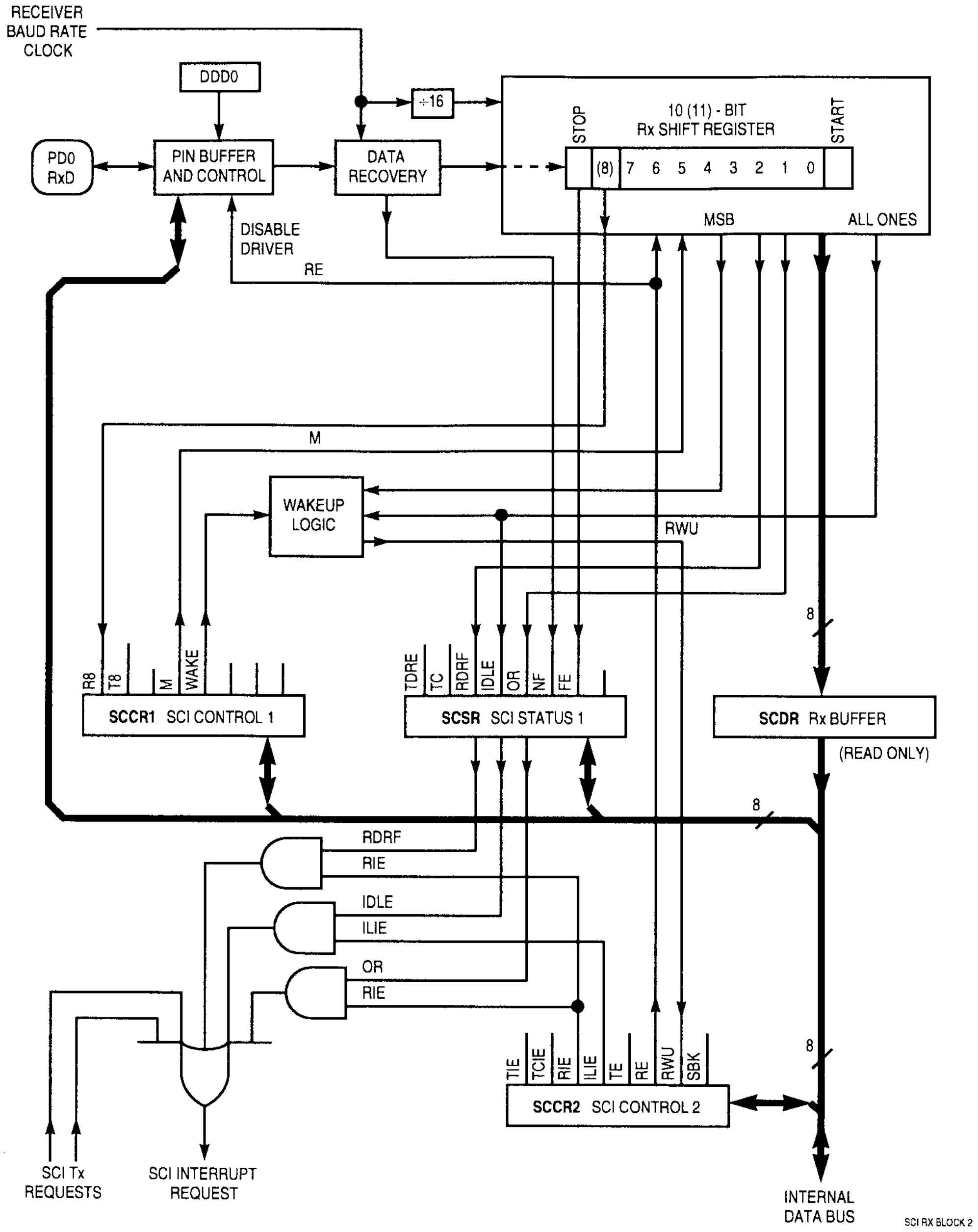


Figure 7–2 SCI Receiver Block Diagram

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. The SCI registers are the same for all M68HC11 E-series devices with one exception. The SCI system for MC68HC(7)11E20 contains an extra bit in the BAUD register that provides a greater selection of baud prescaler rates. Refer to the BAUD register description as well as the block diagrams for the baud rate prescalers.

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.

Bit 5 — Not implemented
Always reads zero

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)

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

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.

	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 with TDRE set 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 with TC set 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 with RDRF set 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 with IDLE set 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 with OR set 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 with NF set 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 with FE set 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] (SCP[2:0] in MC68HC(7)11E20) bits function as a prescaler 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	SCP2	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0
RESET:	0	0	0	0	0	U	U	U

TCLR — Clear Baud Rate Counters (Test)

SCP[2:0] — SCI Baud Rate Prescaler Selects

Note that SCP2 applies to MC68HC(7)11E20 only. When SCP2 = 1, SCP[1:0] must equal zeros. Any other values for SCP[1:0] are not decoded in the prescaler and the results are unpredictable. Refer to the SCI baud rate generator block diagrams.

Table 7–1 Baud Rate Prescaler Selects

Prescaler Selects			Divide Internal Clock By	Crystal Frequency (MHz)				
SCP2	**SCP1**	**SCP0**		**4.0**	**4.9152**	**8.0**	**8.3886**	**12.0**
0	0	0	1	62500	76800	125000	131072	187500
0	0	1	3	20833	25600	41667	43691	62500
0	1	0	4	15625	19200	31250	32768	46875
0	1	1	13	4800	5907	9600	10082	14423
1	0	0	39	1602	1969	3205	3361	4808
1	0	1	Do Not Use					
1	1	0	Do Not Use					
1	1	1	Do Not Use					

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 diagrams.

Table 7–2 Baud Rate Selects

SCR[2:0]	Divide Prescaler By	Highest Baud Rate (Prescaler Output from Previous Table)				
		131072	**76800**	**32768**	**19200**	**4800**
0 0 0	1	131072	76800	32768	19200	4800
0 0 1	2	65536	38400	16384	9600	2400
0 1 0	4	32768	19200	8192	4800	1200
0 1 1	8	16384	9600	4096	2400	600
1 0 0	16	8192	4800	2048	1200	300
1 0 1	32	4096	2400	1024	600	150
1 1 0	64	2048	1200	512	300	75
1 1 1	128	1024	600	256	150	—

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.

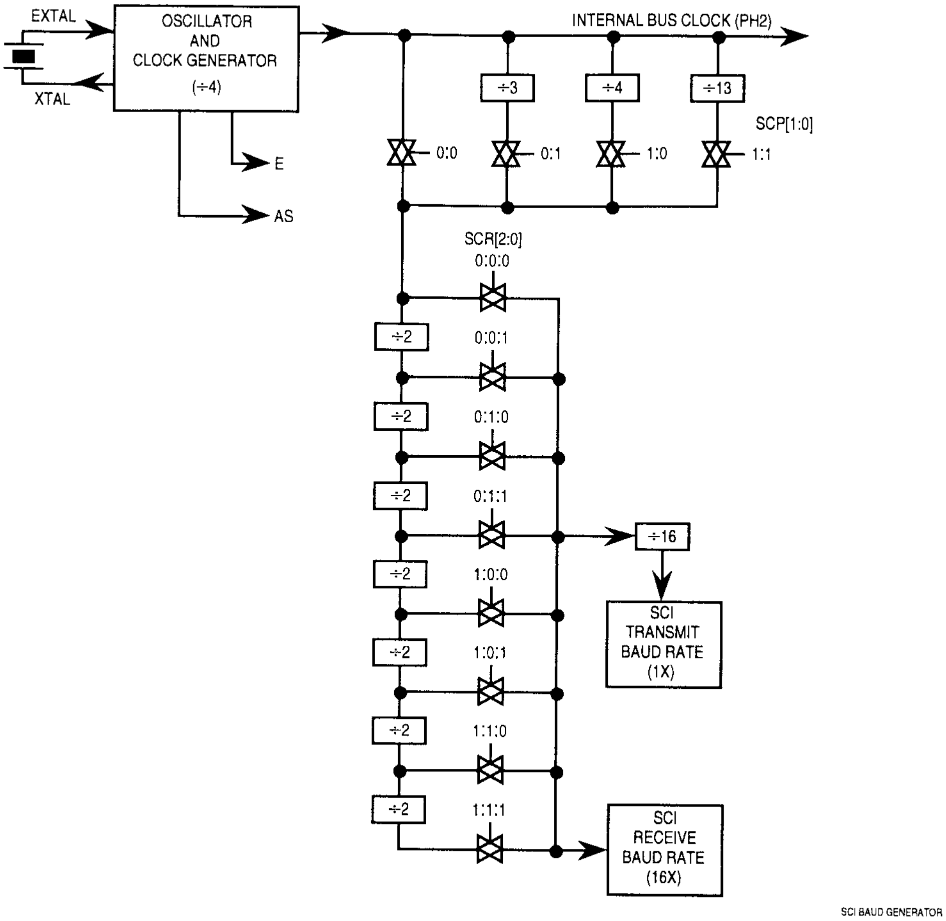


Figure 7–3 SCI Baud Rate Generator Block Diagram

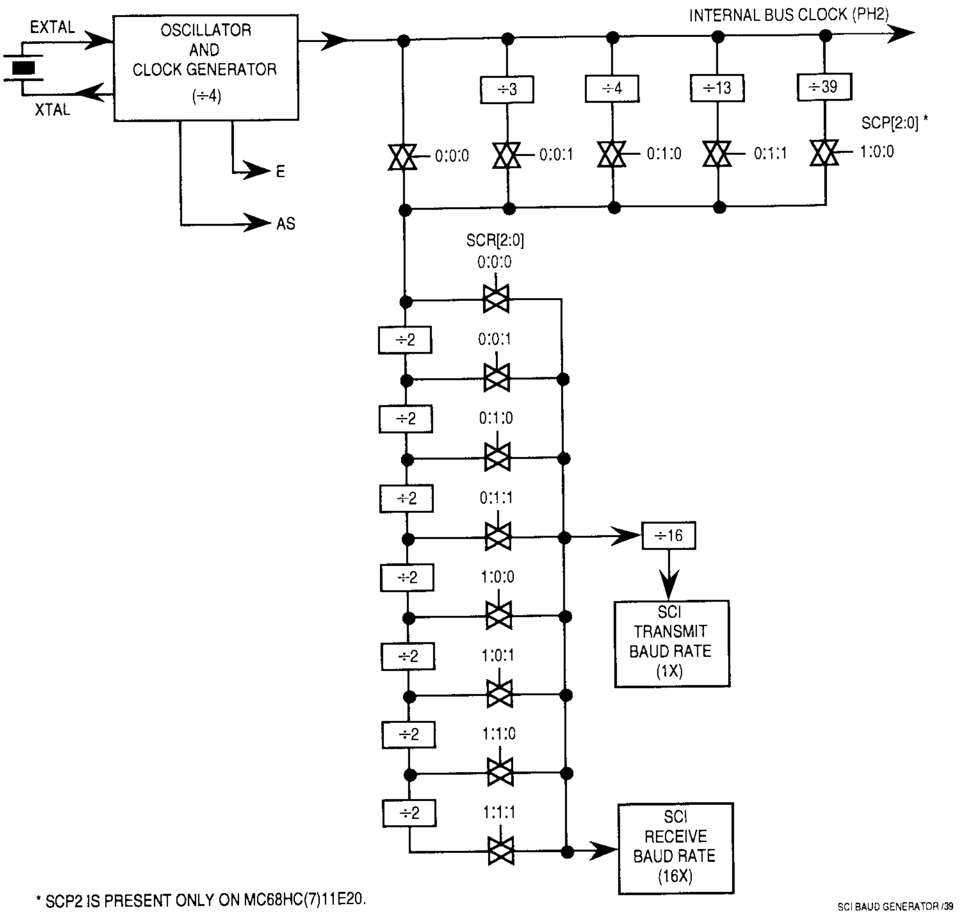


Figure 7–4 MC68HC(7)11E20 SCI Baud Rate Generator Block Diagram

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 7–5**, 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.

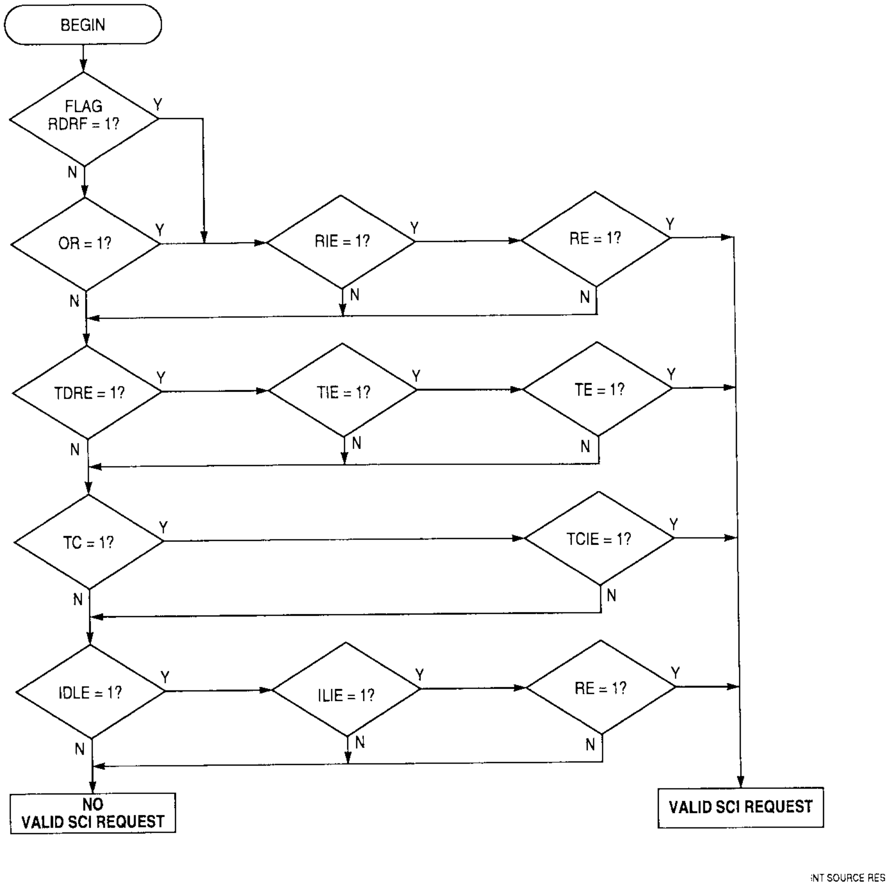


Figure 7–5 Interrupt Source Resolution Within SCI

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 (LCD) display 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 (1.5 Mbits per second for a 3-MHz bus frequency). When configured as a slave, data transfers can be as fast as the E clock rate (3 Mbits per second for a 3-MHz bus frequency).

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 functions (transfer complete, write collision, and mode fault) performed by the serial peripheral 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.

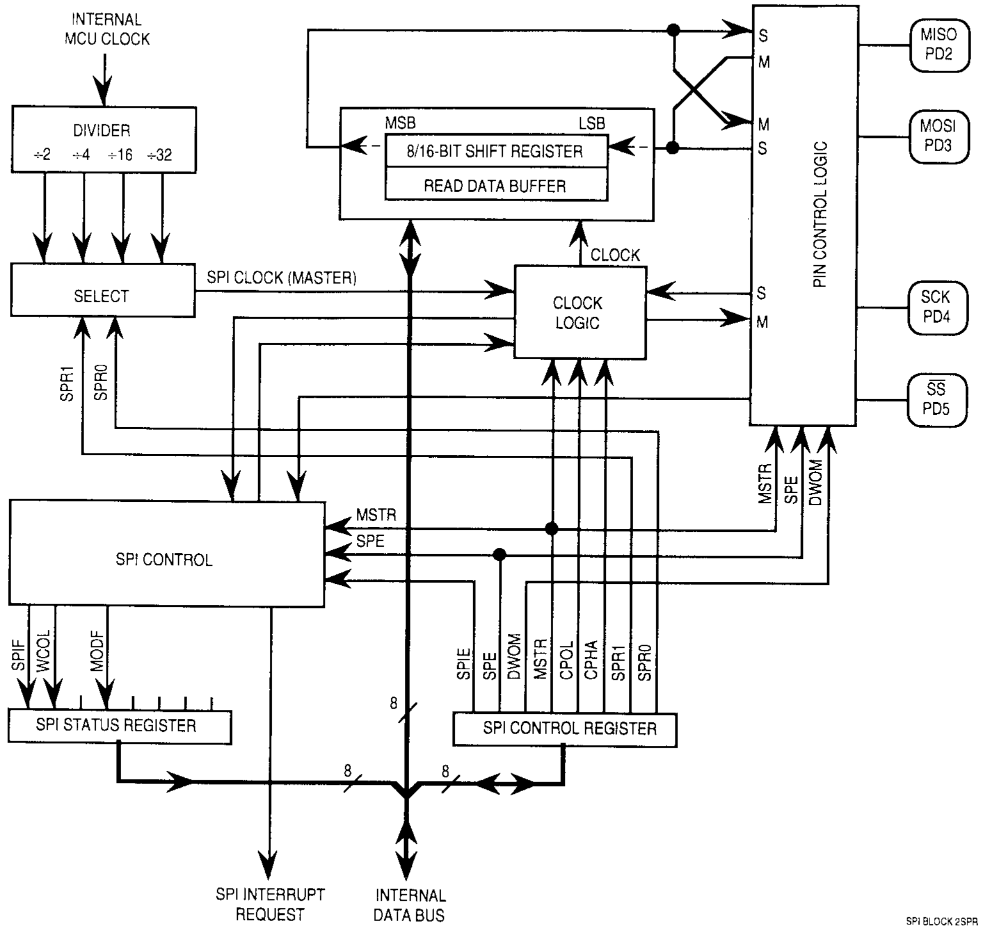
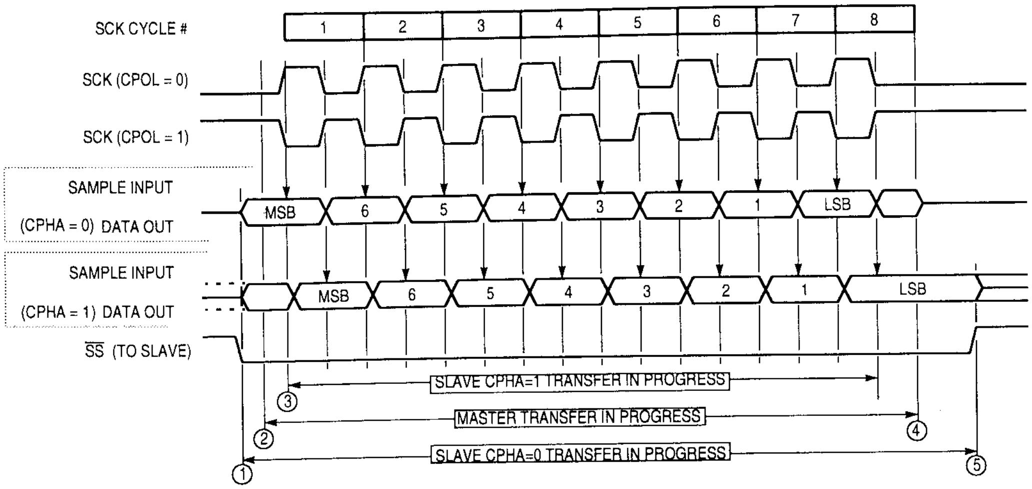


Figure 8–1 SPI Block Diagram

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**.



1. $\overline{SS}$ ASSERTED
2. MASTER WRITES TO SPDR
3. FIRST SCK EDGE
4. SPIF SET
5. $\overline{SS}$ NEGATED

SPI TRANSFER FORMAT 1

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 $\overline{SS}$ 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

It is customary to have an external pull-up resistor on lines that are driven by open-drain devices.

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]	Divide E-Clock By	Frequency at E = 1 MHz (Baud)	Frequency at E = 2 MHz (Baud)	Frequency at E = 3 MHz (Baud)
0 0	2	500 kHz	1.0 MHz	1.5 MHz
0 1	4	250 kHz	500 kHz	750 kHz
1 0	16	62.5 kHz	125 kHz	187.5 kHz
1 1	32	31.3 kHz	62.5 kHz	93.8 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 with SPIF set, 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 (with WCOL set) 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 (with MODF set), 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 I/O 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.

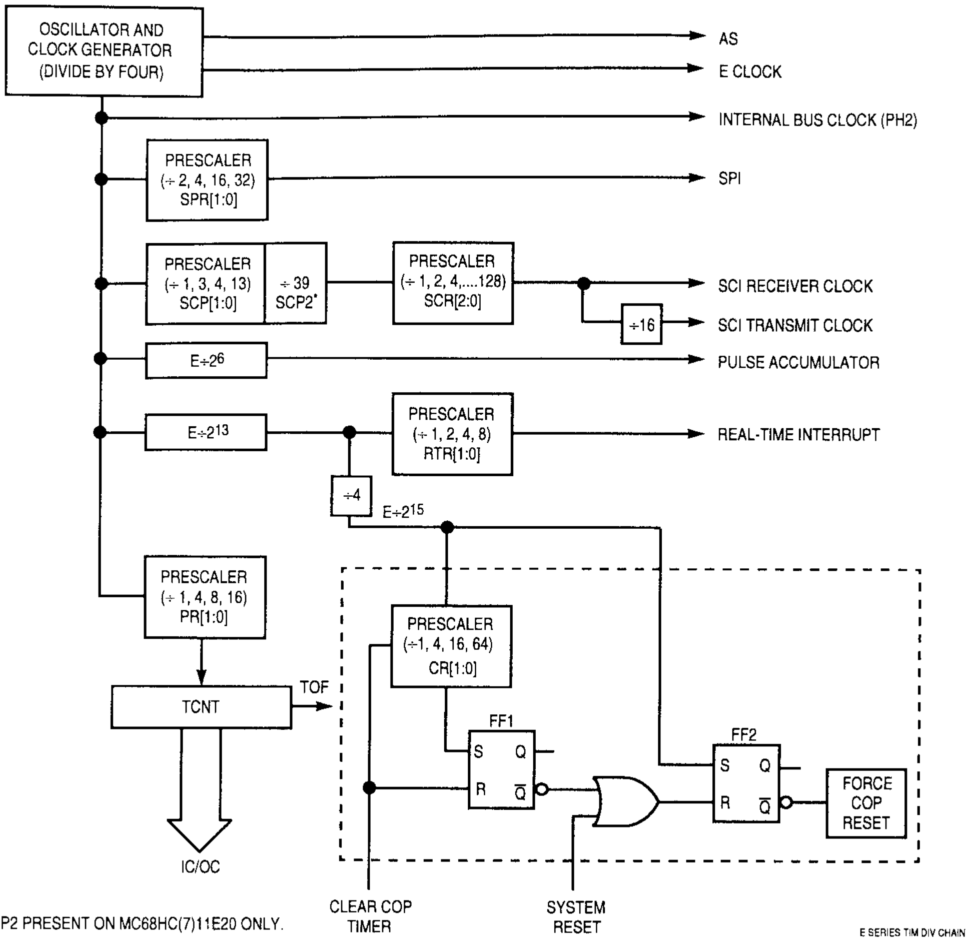
The prescaler output divides the system clock by 1, 4, 8, or 16. Taps off of this main clocking chain drive circuitry that generates the slower clocks used by the pulse accumulator, the real-time interrupt (RTI), and the computer operating properly (COP) watchdog subsystems, also described in this section.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


OSCILLATOR AND CLOCK GENERATOR (DIVIDE BY FOUR)
AS
E CLOCK
INTERNAL BUS CLOCK (PH2)
PRESCALER (÷ 2, 4, 16, 32) SPR[1:0]
SPI
PRESCALER (÷ 1, 3, 4, 13) SCP[1:0]
÷ 39 SCP2*
PRESCALER (÷ 1, 2, 4,....128) SCR[2:0]
SCI RECEIVER CLOCK
÷16
SCI TRANSMIT CLOCK
E÷2^6
PULSE ACCUMULATOR
E÷2^13
PRESCALER (÷ 1, 2, 4, 8) RTR[1:0]
REAL-TIME INTERRUPT
÷4
E÷2^15
PRESCALER (÷ 1, 4, 8, 16) PR[1:0]
TCNT
TOF
IC/OC
PRESCALER (÷1, 4, 16, 64) CR[1:0]
FF1
S
Q
R
Q̅
FF2
S
Q
R
Q̅
FORCE COP RESET
CLEAR COP TIMER
SYSTEM RESET
E SERIES TIM DIV CHAIN

* SCP2 PRESENT ON MC68HC(7)11E20 ONLY.

Figure 9–1 Timer Clock Divider Chains

Table 9–1 Timer Summary

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

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.

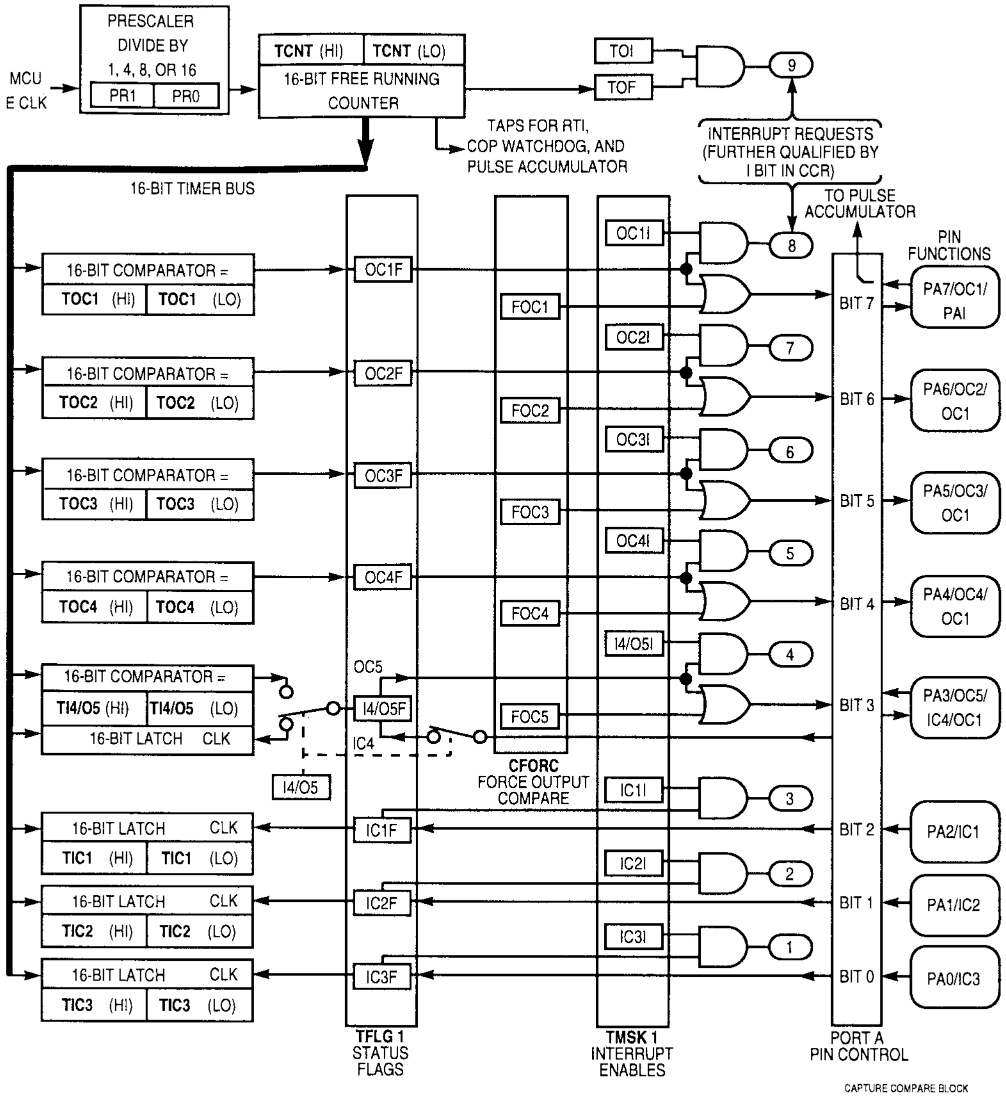


Figure 9–2 Capture/Compare Block Diagram

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 DDRA3 bit of the PACTL 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 DDRA3 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.

Table 9–2 Timer Control Configuration

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

9.2.2 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 Register

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.

$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 read-only.

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 9–3 Timer Output Compare Actions

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

Bits [3:2] — Not implemented
Always read zero

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 9–4 Timer Prescale

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 9–5**, which shows the periodic real-time interrupt rates.

Table 9–5 RTI Rates

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

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 timeouts 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 timeout, not from when RTIF is cleared.

Every timeout 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**.

Bits [3:2] — Not implemented
Always read zero

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
	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0
RESET:	0	0	0	0	0	0	0	0

DDRA7 — Data Direction for Port A Bit 7
Refer to **SECTION 6 PARALLEL INPUT/OUTPUT**.

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**.

DDRA3 — Data Direction for Port A Bit 3
Refer to **SECTION 6 PARALLEL INPUT/OUTPUT**.

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 M68HC11 family of 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 clocked to increasing values by an external pin. 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.

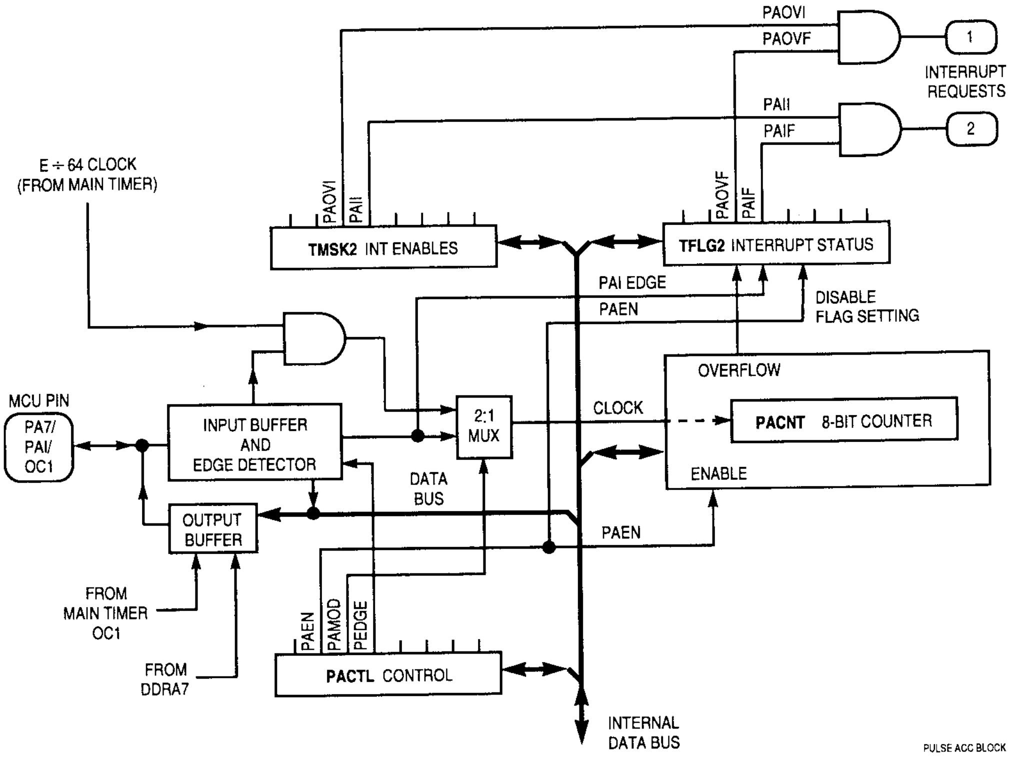


Figure 9–3 Pulse Accumulator

Table 9–6 Pulse Accumulator Timing

Crystal Frequency	E Clock	Cycle Time	E ÷ 64	PACNT Overflow
4.0 MHz	1 MHz	1000 ns	64 µs	16.384 ms
8.0 MHz	2 MHz	500 ns	32 µs	8.192 ms
12.0 MHz	3 MHz	333 ns	21.33 µs	5.461 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
	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0
RESET:	0	0	0	0	0	0	0	0

DDRA7 — Data Direction for Port A Bit 7
Refer to **SECTION 6 PARALLEL INPUT/OUTPUT**.

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 9–7**.

Table 9–7 Pulse Accumulator Edge 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.

DDRA3 — Data Direction for Port A Bit 3
Refer to **SECTION 6 PARALLEL INPUT/OUTPUT**.

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 and 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 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 additional internal analog signal lines are routed to it.

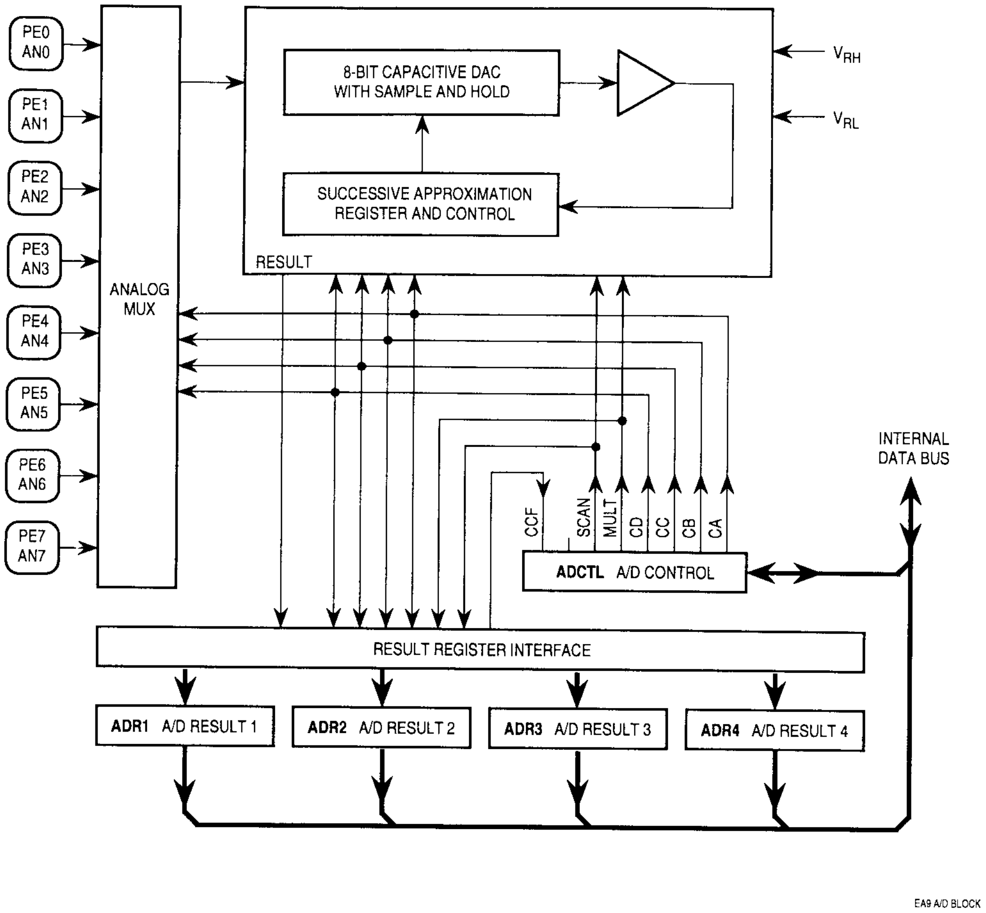
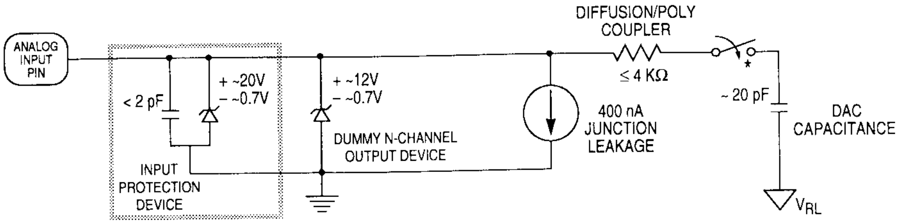


Figure 10–1 A/D Converter Block Diagram

Port E pins can also be used as digital inputs. Digital 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 within up to 100 µs before the converter can be used. The charge pump is enabled by the ADPU bit in the OPTION register.

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 ADR[4:1] 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 E-clock frequency is below 750 kHz, charge leakage in the capacitor array can cause errors, and the internal oscillator should be used. 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.

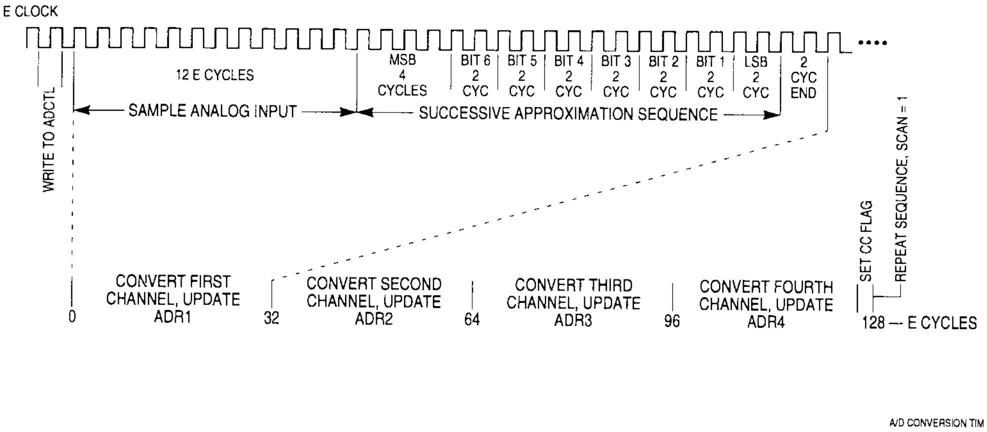


Figure 10–3 A/D Conversion Sequence

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 inherently synchronized to the main MCU clocks. This allows the comparator output to be sampled at relatively quiet times during MCU clock cycles. Since the internal RC oscillator is asynchronous to the MCU clock there is more error attributable to internal system clock noise. A/D converter accuracy is reduced slightly while the internal RC oscillator is being used (CSEL = 1).

OPTION — System Configuration Options **$1039**

	Bit 7	6	5	4	3	2	1	Bit 0
	ADPU	CSEL	IRQE[1]	DLY[1]	CME	—	CR1[1]	CR0[1]
RESET:	0	0	0	1	0	0	0	0

NOTES:

1. 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 Startup Delay

0 = The oscillator startup 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. This delay allows the crystal oscillator to stabilize.

CME — Clock Monitor Enable

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

Bit 2 — Not implemented

Always reads zero.

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 to the MCU, four of the channels are internal reference points or test functions, and four channels are reserved. Refer to **Table 10–1**.

Table 10–1 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 – 12	Reserved	—
13	V_{RH}[1]	ADR1
14	V_{RL}[1]	ADR2
15	$(V_{RH})/2$[1]	ADR3
16	Reserved[1]	ADR4

NOTES:

1. 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 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:	I	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 are performed continuously 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
10XX	Reserved	—
1100	V_{RH}[1]	ADR1
1101	V_{RL}[1]	ADR2
1110	$(V_{RH})/2$[1]	ADR3
1111	Reserved[1]	ADR4

NOTES:

1. 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.

ADR[1:4] — 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

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 MC68HC(7)11Ex MC68HC(7)11ExC MC68HC(7)11ExV MC68HC(7)11ExM MC68HC811E2 MC68HC811E2C MC68HC811E2V MC68HC811E2M MC68L11Ex	T_A	T_L to T_H 0 to + 70 –40 to + 85 –40 to + 105 –40 to + 125 0 to + 70 –40 to + 85 –40 to + 105 –40 to + 125 –20 to + 70	°C
Storage Temperature Range	T_{stg}	–55 to + 150	°C
Current Drain per Pin[1] Excluding V_{DD}, V_{SS}, AV_{DD}, V_{RH}, and V_{RL}	I_D	25	mA

NOTES:

1. 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	$T_A + (P_D \times \Theta_{JA})$	°C
Ambient Temperature	T_A	User-determined	°C
Package Thermal Resistance (Junction-to-Ambient) 48-Pin Plastic DIP (MC68HC811E2 only) 56-Pin Plastic SDIP 52-Pin Plastic Leaded Chip Carrier 52-Pin Plastic Thin Quad Flat Pack (TQFP) 64-Pin Quad Flat Pack	Θ_{JA}	 50 50 50 85 85	°C/W
Total Power Dissipation (Note 1)	P_D	$P_{INT} + P_{I/O}$ $K / (T_J + 273°C)$ (Note 1)	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	$P_D \times (T_A + 273°C) +$ $\Theta_{JA} \times P_D^2$	W·°C

NOTES:

1. This is an approximate value, neglecting $P_{I/O}$.
2. For most applications 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 ±10%, V_{SS} = 0 Vdc, T_A = T_L to T_H, unless otherwise noted

Characteristics		Symbol	Min	Max	Unit
Output Voltage (Note 1)	All Outputs except XTAL	V_{OL}	—	0.1	V
	All Outputs Except XTAL, $\overline{\text{RESET}}$, and MODA	V_{OH}	V_{DD} –0.1	—	
I_{Load} = ±10.0 µA					
Output High Voltage (Note 1)	All Outputs Except XTAL, $\overline{\text{RESET}}$, and MODA	V_{OH}	V_{DD} –0.8	—	V
I_{Load} = –0.8 mA, V_{DD} = 4.5 V					
Output Low Voltage	All Outputs Except XTAL	V_{OL}	—	0.4	V
I_{Load} = 1.6 mA					
Input High Voltage	All Inputs Except $\overline{\text{RESET}}$	V_{IH}	0.7 × V_{DD}	V_{DD} + 0.3	V
	$\overline{\text{RESET}}$		0.8 × VDD	V_{DD} + 0.3	
Input Low Voltage	All Inputs	V_{IL}	V_{SS} –0.3	0.2 × V_{DD}	V
I/O Ports, Three-State Leakage	PA7, PA3, PC[7:0], PD[5:0], AS/STRA,	I_{OZ}	—	±10	µA
V_{in} = V_{IH} or V_{IL}	MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$				
Input Leakage Current (Note 2)		I_{in}			
V_{in} = V_{DD} or V_{SS}	PA[2:0], $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$		—	±1	µA
V_{in} = V_{DD} or V_{SS}	MODB/V_{STBY} ($\overline{\text{XIRQ}}$ on EPROM-based devices)		—	±10	µA
RAM Standby Voltagee	Power down	V_{SB}	4.0	V_{DD}	V
RAM Standby Current	Power down	I_{SB}	—	10	µA
Input Capacitance	PA[2:0], PE[7:0], $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$, EXTAL	C_{in}	—	8	pF
	PA7, PA3, PC[7:0], PD[5:0], AS/STRA, MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$		—	12	pF
Output Load Capacitance	All Outputs Except PD[4:1]	C_L	—	90	pF
	PD[4:1]		—	100	pF
Maximum Total Supply Current (Note 3)					
RUN:		I_{DD}			
Single-Chip Mode	2 MHz		—	15	mA
	3 MHz		—	27	mA
Expanded Multiplexed Mode	2 MHz		—	27	mA
	3 MHz		—	35	mA
WAIT: (All Peripheral Functions Shut Down)		W_{IDD}			
Single-Chip Mode	2 MHz		—	6	mA
	3 MHz		—	15	mA
Expanded Multiplexed Mode	2 MHz		—	10	mA
	3 MHz		—	20	mA
STOP:		S_{IDD}			
Single-Chip Mode, No Clocks	–40 to + 85		—	25	µA
	> + 85 to + 105		—	50	
	> +105 to + 125		—	100	
Maximum Power Dissipation		P_D			
Single-Chip Mode	2 MHz		—	85	mW
	3 MHz		—	150	mW
Expanded Multiplexed Mode	2 MHz		—	150	mW
	3 MHz		—	195	mW

NOTES:

1. V_{OH} specification for $\overline{\text{RESET}}$ 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; V_{IL} ≤0.2 V;
   V_{IH} ≥V_{DD} – 0.2 V; No dc loads

Table A–3a DC Electrical Characteristics (MC68L11E9)

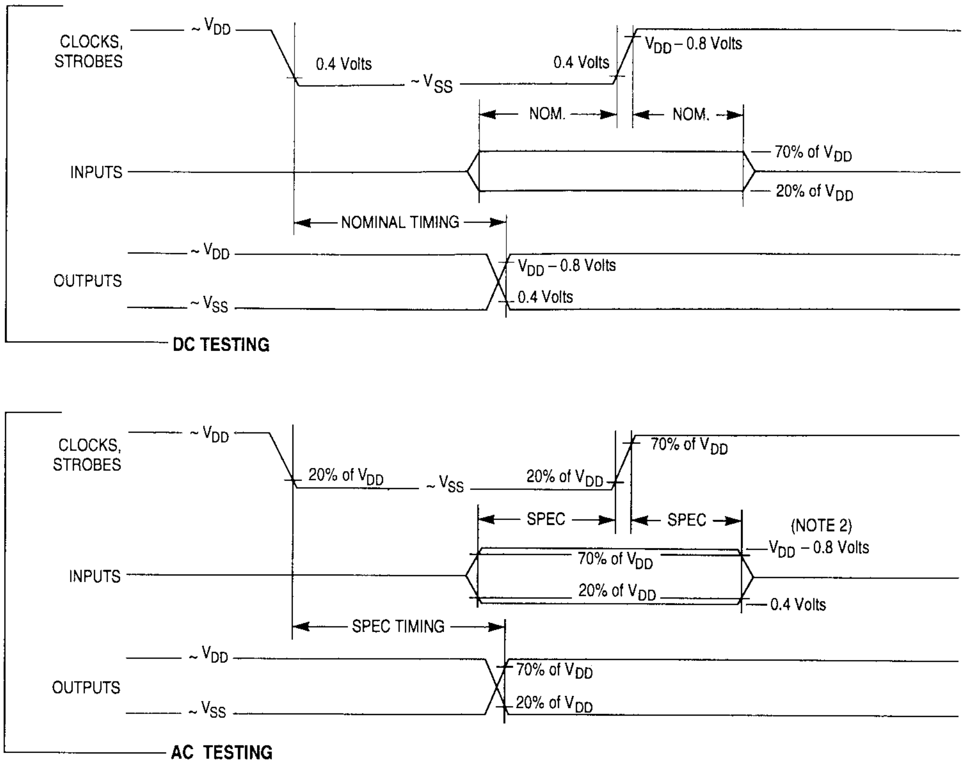
V_{DD} = 3.0 Vdc to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted

Characteristic	Symbol	Min	Max	Unit
Output Voltage (Note 1) All Outputs except XTAL All Outputs Except XTAL, $\overline{\text{RESET}}$, 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, $\overline{\text{RESET}}$, and MODA I_{Load} = −0.5 mA, V_{DD} = 3.0 V 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_{DD} = 5.0 V I_{Load} = 1.0 mA, V_{DD} = 3.0 V	V_{OL}	—	0.4	V
Input High Voltage All Inputs Except $\overline{\text{RESET}}$ $\overline{\text{RESET}}$	V_{IH}	0.7 × V_{DD} 0.8 × VDD	V_{DD} + 0.3 V_{DD} + 0.3	V V
Input Low Voltage All Inputs	V_{IL}	V_{SS} −0.3	0.2 × V_{DD}	V
I/O Ports, Three-State Leakage PA7, PA3, PC[7:0], PD[5:0], AS/STRA, MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$ $V_{in} = V_{IH}$ or V_{IL}	I_{OZ}	—	±10	µA
Input Leakage Current (Note 2) $V_{in} = V_{DD}$ or V_{SS} PA[2:0], $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$ $V_{in} = V_{DD}$ or V_{SS} MODB/V_{STBY} ($\overline{\text{XIRQ}}$ on EPROM-based devices)	I_{in}	 — —	 ±1 ±10	 µA µA
RAM Standby Voltage Power down	V_{SB}	2.0	V_{DD}	V
RAM Standby Current Power down	I_{SB}	—	10	µA
Input Capacitance PA[2:0], PE[7:0], $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$, EXTAL PA7, PA3, PC[7:0], PD[5:0], AS/STRA, MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$	C_{in}	— —	8 12	pF pF
Output Load Capacitance All Outputs Except PD[4:1] PD[4:1]	C_L	— —	90 100	pF pF

Characteristic		Symbol	1 MHz	2 MHz	Unit
Maximum Total Supply Current (Note 3)					
RUN:		I_{DD}			
Single-Chip Mode	V_{DD} = 5.5 V		8	15	mA
	V_{DD} = 3.0 V		4	8	mA
Expanded Multiplexed Mode	V_{DD} = 5.5 V		14	27	mA
	V_{DD} = 3.0 V		7	14	mA
WAIT:(All Peripheral Functions Shut Down)		WI_{DD}			
Single-Chip Mode	V_{DD} = 5.5 V		3	6	mA
	V_{DD} = 3.0 V		1.5	3	mA
Expanded Multiplexed Mode	V_{DD} = 5.5 V		5	10	mA
	V_{DD} = 3.0 V		2.5	5	mA
STOP:		SI_{DD}			
Single-Chip Mode, No Clocks	V_{DD} = 5.5 V		50	50	µA
	V_{DD} = 3.0 V		25	25	µA
Maximum Power Dissipation		P_D			
Single-Chip Mode	V_{DD} = 5.5 V		44	85	mW
	V_{DD} = 3.0 V		12	24	mW
Expanded Multiplexed Mode	V_{DD} = 5.5 V		77	150	mW
	V_{DD} = 3.0 V		21	42	mW

NOTES:

1. V_{OH} specification for $\overline{\text{RESET}}$ 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} = 1000 ns for 1 MHz rating;
   t_{cyc} = 500 ns for 2 MHz rating; V_{IL} ≤0.2 V;
   V_{IH} ≥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 VDD – 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 ±10%, V_{SS} = 0 Vdc, T_A = T_L to T_H

Characteristic	Symbol	1.0 MHz		2.0 MHz		3.0 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_o	dc	1.0	dc	2.0	dc	3.0	MHz
E-Clock Period	t_{cyc}	1000	—	500	—	333	—	ns
Crystal Frequency	f_{XTAL}	—	4.0	—	8.0	—	12.0	MHz
External Oscillator Frequency	4 f_o	dc	4.0	dc	8.0	dc	12.0	MHz
Processor Control Setup Time t_{PCSU} = 1/4 t_{cyc} + 50 ns	t_{PCSU}	300	—	175	—	133	—	ns
Reset Input Pulse Width	PW_{RSTL}							t_{cyc}
To Guarantee External Reset Vector		8	—	8	—	8	—	
Minimum Input Time (Can Be Preempted by Internal Reset)		1	—	1	—	1	—	
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}	1020	—	520	—	353	—	ns
Wait Recovery Start-up Time	t_{WRS}	—	4	—	4	—	4	t_{cyc}
Timer Pulse Width Input Capture Pulse Accumulator Input PW_{TIM} = t_{cyc} + 20 ns	PW_{TIM}	1020	—	520	—	353	—	ns

NOTES:

1. $\overline{RESET}$ 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.
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.

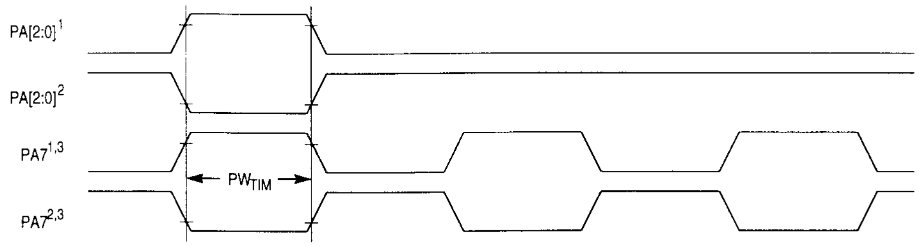
Table A–4a Control Timing (MC68L11E9)

V_{DD} = 3.0 Vdc to 5.5 Vdc, V_{SS} = 0 Vdc, T_A = T_L to T_H

Characteristic	Symbol	1.0 MHz		2.0 MHz		Unit
		Min	Max	Min	Max	
Frequency of Operation	f_o	dc	1.0	dc	2.0	MHz
E-Clock Period	t_{cyc}	1000	—	500	—	ns
Crystal Frequency	f_{XTAL}	—	4.0	—	8.0	MHz
External Oscillator Frequency	4 f_o	dc	4.0	dc	8.0	MHz
Processor Control Setup Time t_{PCSU} = 1/4 t_{cyc} + 75 ns	t_{PCSU}	325	—	200	—	ns
Reset Input Pulse Width	PW_{RSTL}					
To Guarantee External Reset Vector		8	—	8	—	t_{cyc}
Minimum Input Time (Can Be Preempted by Internal Reset)		1	—	1	—	t_{cyc}
Mode Programming Setup Time	t_{MPS}	2	—	2	—	t_{cyc}
Mode Programming Hold Time	t_{MPH}	10	—	10	—	ns
Interrupt Pulse Width, $\overline{IRQ}$ Edge-Sensitive Mode PW_{IRQ} = t_{cyc} + 20 ns	PW_{IRQ}	1020	—	520	—	ns
Wait Recovery Start-up Time	t_{WRS}	—	4	—	4	t_{cyc}
Timer Pulse Width, Input Capture Pulse Accumulator Input PW_{TIM} = t_{cyc} + 20 ns	PW_{TIM}	1020	—	520	—	ns

NOTES:

1. $\overline{RESET}$ 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.
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.



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

TIMER INPUTS TIM

Figure A–2 Timer Inputs

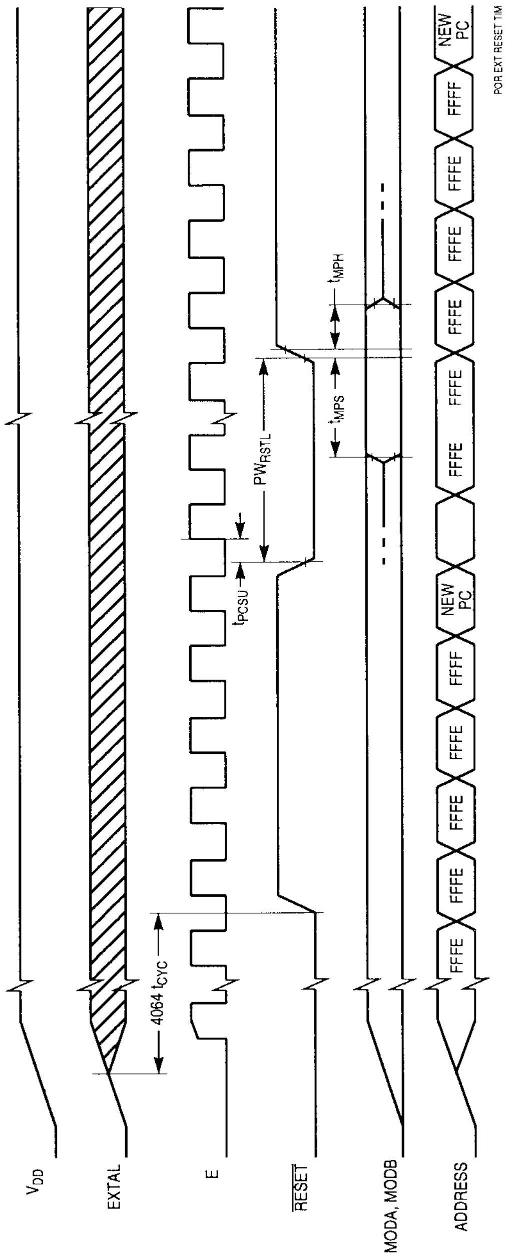
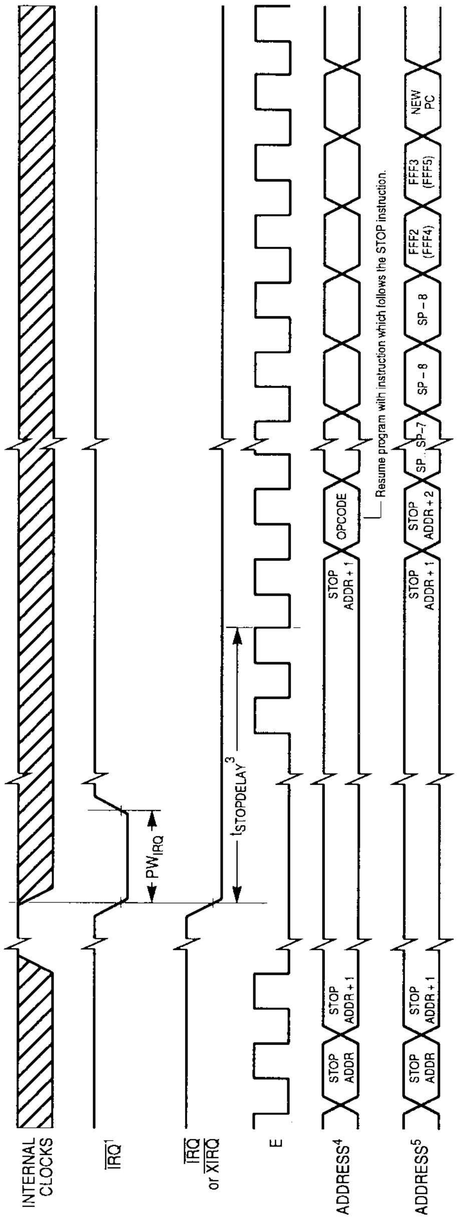


Figure A–3 POR External Reset Timing Diagram



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).

STOP RECOVERY TIM

Figure A–4 STOP Recovery Timing Diagram

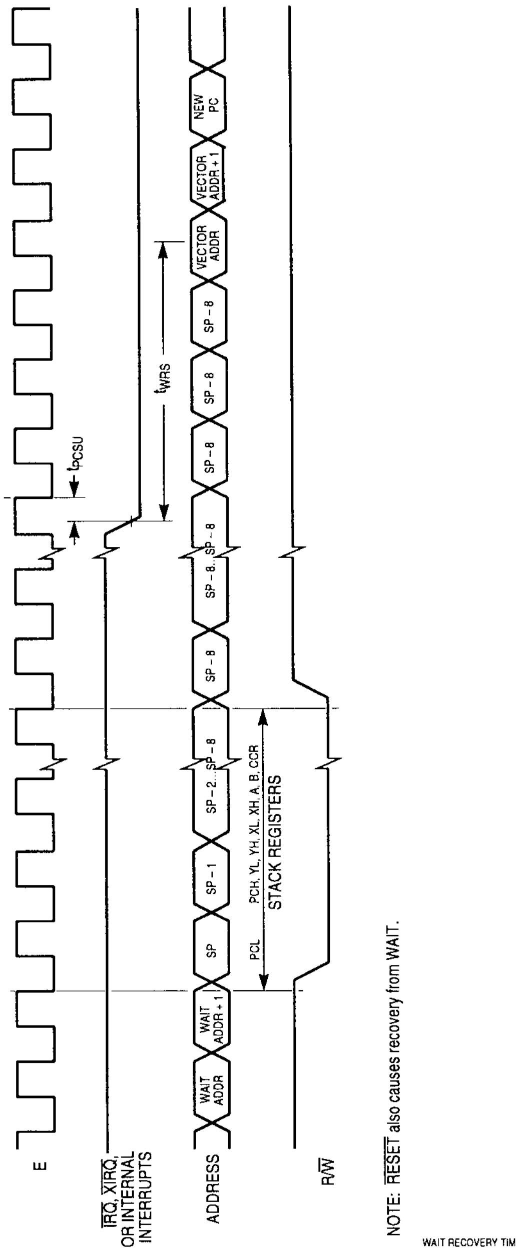
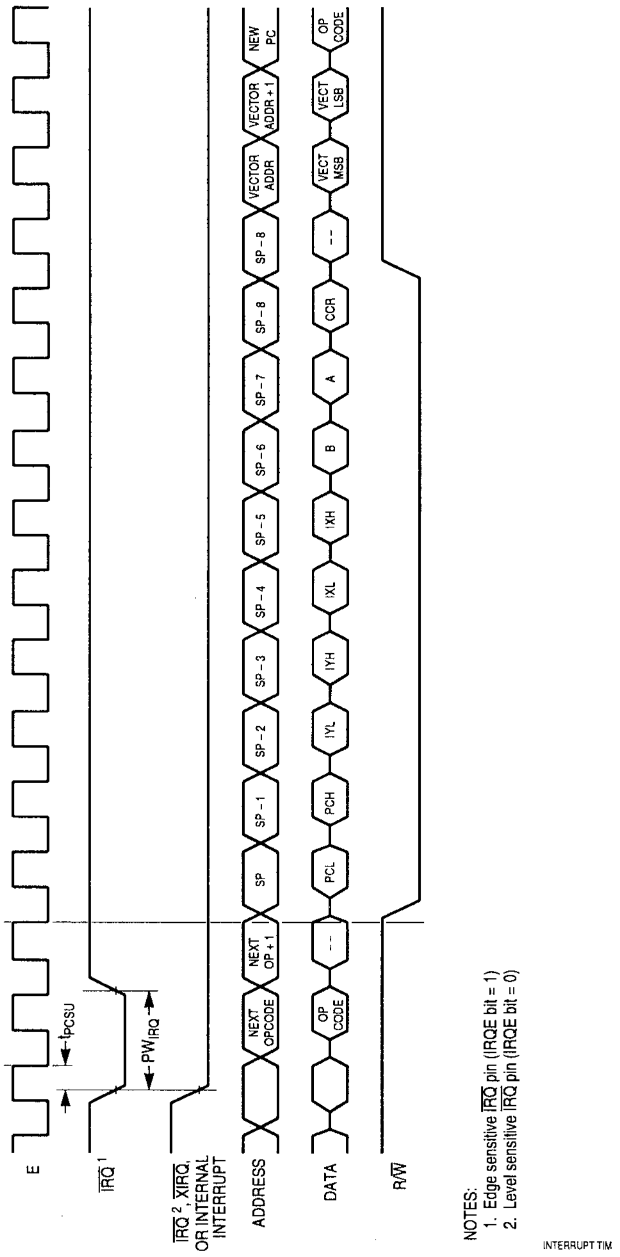


Figure A–5 WAIT Recovery from Interrupt Timing Diagram



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

Characteristic	Symbol	1.0 MHz		2.0 MHz		3.0 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation (E-Clock Frequency)	f_o	dc	1.0	dc	2.0	dc	3.0	MHz
E-Clock Period	t_{cyc}	1000	—	500	—	333	—	ns
Peripheral Data Setup Time MCU Read of Ports A, C, D, and E	t_{PDSU}	100	—	100	—	100	—	ns
Peripheral Data Hold Time MCU Read of Ports A, C, D, and E	t_{PDH}	50	—	50	—	50	—	ns
Delay Time, Peripheral Data Write	t_{PWD}							
MCU Write to Port A		—	200	—	200	—	200	ns
MCU Writes to Ports B, C, and D		—	350	—	225	—	183	ns
t_{PWD} = 1/4 t_{cyc} + 100 ns								
Input Data Setup Time (Port C)	t_{IS}	60	—	60	—	60	—	ns
Input Data Hold Time (Port C)	t_{IH}	100	—	100	—	100	—	ns
Delay Time, E Fall to STRB t_{DEB} = 1/4 t_{cyc} + 100 ns	t_{DEB}	—	350	—	225	—	183	ns
Setup Time, STRA Asserted to E Fall (Note 1)	t_{AES}	0	—	0	—	0	—	ns
Delay Time, STRA Asserted to Port C Data Output Valid	t_{PCD}	—	100	—	100	—	100	ns
Hold Time, STRA Negated to Port C Data	t_{PCH}	10	—	10	—	10	—	ns
Three-State Hold Time	t_{PCZ}	—	150	—	150	—	150	ns

NOTES:

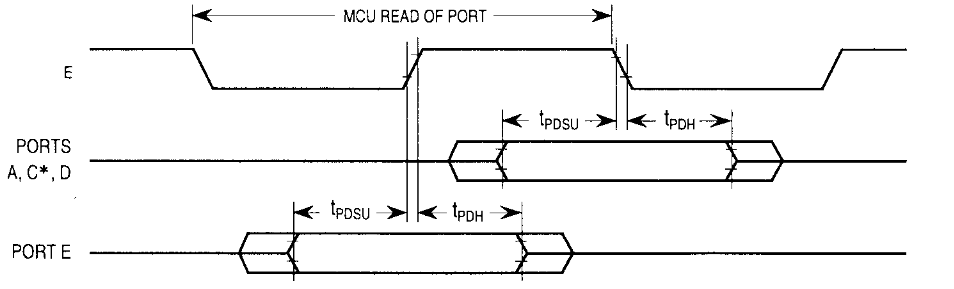
1. If this setup time is met, STRB acknowledges in the next cycle. If it is not met, the response may be delayed one more cycle.
2. Port C and D timing is valid for active drive (CWOM and DWOM bits not set in PIOC and SPCR registers respectively).
3. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.

Table A–5a Peripheral Port Timing (MC68L11E9)

V_{DD} = 3.0 Vdc to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H

Characteristic	Symbol	1.0 MHz		2.0 MHz		Unit
		Min	Max	Min	Max	
Frequency of Operation (E-Clock Frequency)	f_o	dc	1.0	dc	2.0	MHz
E-Clock Period	t_{cyc}	1000	—	500	—	ns
Peripheral Data Setup Time MCU Read of Ports A, C, D, and E	t_{PDSU}	100	—	100	—	ns
Peripheral Data Hold Time MCU Read of Ports A, C, D, and E	t_{PDH}	50	—	50	—	ns
Delay Time, Peripheral Data Write MCU Write to Port A MCU Writes to Ports B, C, and D t_{PWD} = 1/4 t_{cyc} + 150 ns	t_{PWD}	— —	250 400	— —	250 275	ns ns
Input Data Setup Time (Port C)	t_{IS}	60	—	60	—	ns
Input Data Hold Time (Port C)	t_{IH}	100	—	100	—	ns
Delay Time, E Fall to STRB t_{DEB} = 1/4 t_{cyc} + 150 ns	t_{DEB}	—	400	—	275	ns
Setup Time, STRA Asserted to E Fall (Note 1)	t_{AES}	0	—	0	—	ns
Delay Time, STRA Asserted to Port C Data Output Valid	t_{PCD}	—	100	—	100	ns
Hold Time, STRA Negated to Port C Data	t_{PCH}	10	—	10	—	ns
Three-State Hold Time	t_{PCZ}	—	150	—	150	ns

NOTES:
1. If this setup time is met, STRB acknowledges in the next cycle. If it is not met, the response may be delayed one more cycle.
2. Port C and D timing is valid for active drive (CWOM and DWOM bits not set in PIOC and SPCR registers respectively).
3. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.



* FOR NON-LATCHED OPERATION OF PORT C

Figure A–7 Port Read Timing Diagram

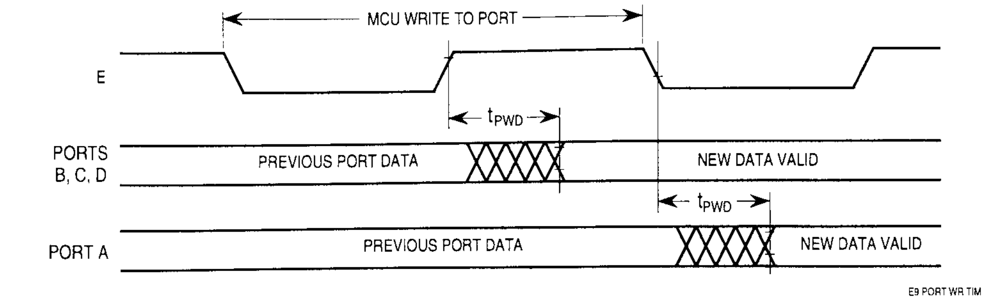


Figure A–8 Port Write Timing Diagram

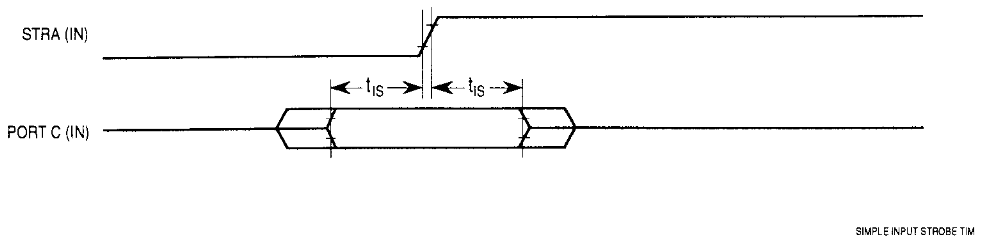


Figure A–9 Simple Input Strobe Timing Diagram

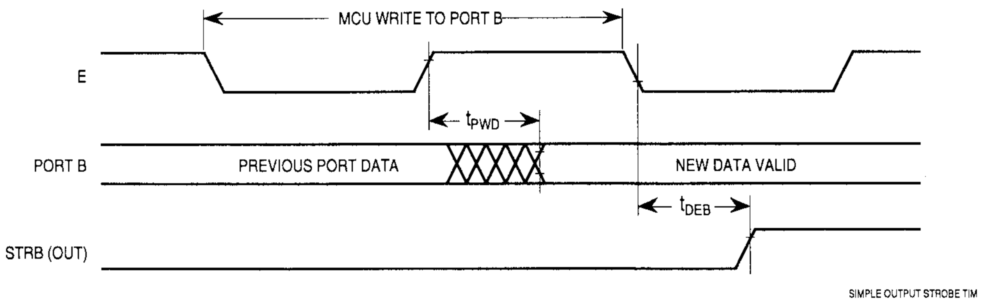
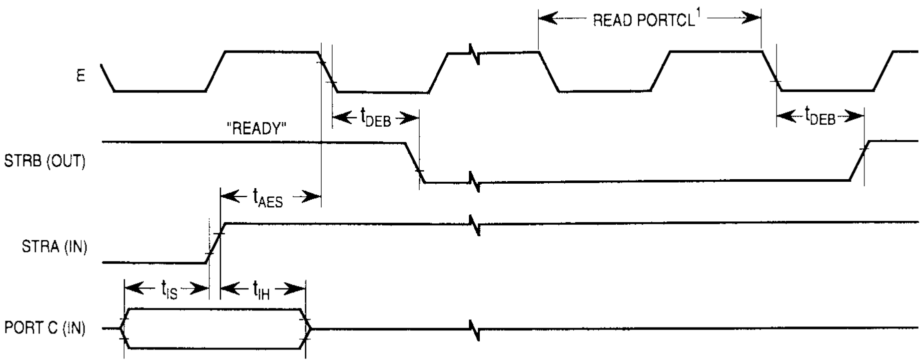


Figure A–10 Simple Output Strobe Timing Diagram

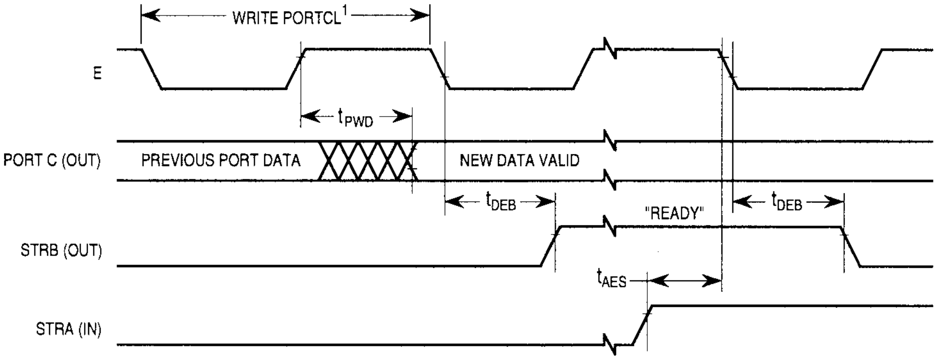


NOTES:

1. After reading PIOC with STAF set
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

PORT C INPUT HNDSHK TIM

Figure A–11 Port C Input Handshake Timing Diagram

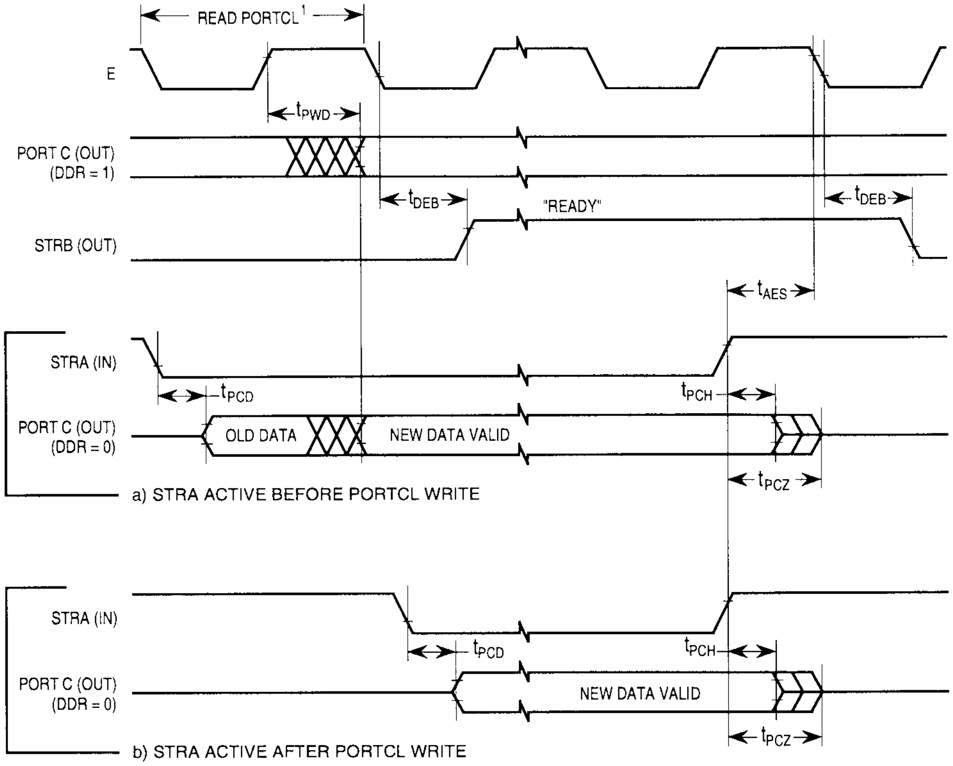


NOTES:

1. After reading PIOC with STAF set
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

PORT C OUTPUT HNDSHK TIM

Figure A–12 Port C Output Handshake Timing Diagram



NOTES:
1. After reading PIOC with STAF set
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

3-STATE VAR OUTPUT HNDSHK TIM

Figure A–13 Three-State Variation of Output Handshake Timing Diagram (STRA Enables Output Buffer)

Table A–6 Analog-To-Digital Converter Characteristics

V_{DD} = 5.0 Vdc ±10%, 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	Unit
				Max	Max	
Resolution	Number of Bits Resolved by A/D Converter	—	8	—	—	Bits
Non-Linearity	Maximum Deviation from the Ideal A/D Transfer Characteristics	—	—	±1/2	±1	LSB
Zero Error	Difference Between the Output of an Ideal and an Actual for Zero Input Voltage	—	—	±1/2	±1	LSB
Full Scale Error	Difference Between the Output of an Ideal and an Actual A/D for Full-Scale Input Voltage	—	—	±1/2	±1	LSB
Total Unadjusted Error	Maximum Sum of Non-Linearity, Zero Error, and Full-Scale Error	—	—	±1/2	±1 1/2	LSB
Quantization Error	Uncertainty Because of Converter Resolution	—	—	±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	LSB
Conversion Range	Analog Input Voltage Range	V_{RL}	—	V_{RH}	V_{RH}	V
V_{RH}	Maximum Analog Reference Voltage (Note 2)	V_{RL}	—	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
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	—	32	—	—	t_{cyc}
	Internal RC Oscillator	—	—	t_{cyc} + 32	t_{cyc} + 32	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	Hex
Sample Acquisition Time	Analog Input Acquisition Sampling Time:					
	E Clock	—	12	—	—	t_{cyc}
	Internal RC Oscillator	—	—	12	12	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]	—	—	400	400	nA
	V_{RL}, V_{RH}	—	—	1.0	1.0	µA

NOTES:
1. 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–6a Analog-To-Digital Converter Characteristics (MC68L11E9)

V_{DD} = 3.0 Vdc to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to $T_{H,}$ 750 kHz ≤E ≤2.0 MHz, unless otherwise noted

Characteristic	Parameter	Min	Absolute	Max	Unit
Resolution	Number of Bits Resolved by A/D Converter	—	8	—	Bits
Non-Linearity	Maximum Deviation from the Ideal A/D Transfer Characteristics	—	—	±1	LSB
Zero Error	Difference Between the Output of an Ideal and an Actual for Zero Input Voltage	—	—	±1	LSB
Full Scale Error	Difference Between the Output of an Ideal and an Actual A/D for Full-Scale Input Voltage	—	—	±1	LSB
Total Unadjusted Error	Maximum Sum of Non-Linearity, Zero Error, and Full-Scale Error	—	—	±1 1/2	LSB
Quantization Error	Uncertainty Because of Converter Resolution	—	—	±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	—	—	±2	LSB
Conversion Range	Analog Input Voltage Range	V_{RL}	—	V_{RH}	V
V_{RH}	Maximum Analog Reference Voltage	V_{RL}	—	V_{DD} + 0.1	V
V_{RL}	Minimum Analog Reference Voltage	V_{SS} –0.1	—	V_{RH}	V
ΔV_R	Minimum Difference between V_{RH} and V_{RL}	3.0	—	—	V
Conversion Time	Total Time to Perform a Single Analog-to-Digital Conversion:				
	E Clock	—	32	—	t_{cyc}
	Internal RC Oscillator	—	—	t_{cyc} + 32	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	Hex
Sample Acquisition Time	Analog Input Acquisition Sampling Time:				
	E Clock	—	12	—	t_{cyc}
	Internal RC Oscillator	—	—	12	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]	—	—	400	nA
	V_{RL}, V_{RH}	—	—	1.0	µA

NOTES:
1. Source impedances greater than 10 kΩ affect accuracy adversely because of input leakage.

Table A–7 Expansion Bus Timing

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

Num	Characteristic	Symbol	1.0 MHz		2.0 MHz		3.0 MHz		Unit
			Min	Max	Min	Max	Min	Max	
	Frequency of Operation (E-Clock Frequency)	f_o	dc	1.0	dc	2.0	dc	3.0	MHz
1	Cycle Time	t_{cyc}	1000	—	500	—	333	—	ns
2	Pulse Width, E Low $PW_{EL} = 1/2\ t_{cyc} - 23$ ns (Note 1)	PW_{EL}	477	—	227	—	146	—	ns
3	Pulse Width, E High $PW_{EH} = 1/2\ t_{cyc} - 28$ ns (Note 1)	PW_{EH}	472	—	222	—	141	—	ns
4a 4b	E and AS Rise Time E and AS Fall Time	t_r t_f	— —	20 20	— —	20 20	— —	20 15	ns
9	Address Hold Time $t_{AH} = 1/8\ t_{cyc} - 29.5$ ns (Note 1, 2a)	t_{AH}	95.5	—	33	—	26	—	ns
12	Nonmultiplexed Address Valid Time to E Rise $t_{AV} = PW_{EL} - (t_{ASD} + 80$ ns) (Note 1, 2a)	t_{AV}	281.5	—	94	—	54	—	ns
17	Read Data Setup Time	t_{DSR}	30	—	30	—	30	—	ns
18	Read Data Hold Time (Max = t_{MAD})	t_{DHR}	0	145.5	0	83	0	51	ns
19	Write Data Delay Time $t_{DDW} = 1/8\ t_{cyc} + 65.5$ ns (Note 1, 2a)	t_{DDW}	—	190.5	—	128		71	ns
21	Write Data Hold Time $t_{DHW} = 1/8\ t_{cyc} - 29.5$ ns (Note 1, 2a)	t_{DHW}	95.5	—	33	—	26	—	ns
22	Multiplexed Address Valid Time to E Rise $t_{AVM} = PW_{EL} - (t_{ASD} + 90$ ns) (Note 1, 2a)	t_{AVM}	271.5	—	84	—	54	—	ns
24	Multiplexed Address Valid Time to AS Fall $t_{ASL} = PW_{ASH} - 70$ ns (Note 1)	t_{ASL}	151	—	26	—	13	—	ns
25	Multiplexed Address Hold Time $t_{AHL} = 1/8\ t_{cyc} - 29.5$ ns (Note 1, 2b)	t_{AHL}	95.5	—	33	—	31	—	ns
26	Delay Time, E to AS Rise $t_{ASD} = 1/8\ t_{cyc} - 9.5$ ns (Note 1, 2a)	t_{ASD}	115.5	—	53	—	31	—	ns
27	Pulse Width, AS High $PW_{ASH} = 1/4\ t_{cyc} - 29$ ns (Note 1)	PW_{ASH}	221	—	96	—	63	—	ns
28	Delay Time, AS to E Rise $t_{ASED} = 1/8\ t_{cyc} - 9.5$ ns (Note 1, 2b)	t_{ASED}	115.5	—	53	—	31	—	ns
29	MPU Address Access Time (Note 2a) $t_{ACCA} = t_{cyc} - (PW_{EL} - t_{AVM}) - t_{DSR} - t_f$	t_{ACCA}	744.5	—	307	—	196	—	ns
35	MPU Access Time $t_{ACCE} = PW_{EH} - t_{DSR}$	t_{ACCE}	—	442	—	192		111	ns
36	Multiplexed Address Delay (Previous Cycle MPU Read) $t_{MAD} = t_{ASD} + 30$ ns (Note 1, 2a)	t_{MAD}	145.5	—	83	—	51	—	ns

1. Formula only for dc to 2 MHz.
2. Input clocks with duty cycles other than 50% affect bus performance. Timing parameters affected by input clock duty cycle are identified by (a) and (b). To recalculate the approximate bus timing values, substitute the following expressions in place of 1/8 t_{cyc} in the above formulas, where applicable:
   (a) (1–DC) × 1/4 t_{cyc}
   (b) DC × 1/4 t_{cyc}
   Where:
   DC is the decimal value of duty cycle percentage (high time).
3. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.

Table A–7a Expansion Bus Timing (MC68L11E9)

V_{DD} = 3.0 Vdc to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H

Num	Characteristic	Symbol	1.0 MHz		2.0 MHz		Unit
			Min	Max	Min	Max	
	Frequency of Operation (E-Clock Frequency)	f_o	dc	1.0	dc	2.0	MHz
1	Cycle Time	t_{cyc}	1000	—	500	—	ns
2	Pulse Width, E Low PW_{EL} = 1/2 t_{cyc} –25 ns	PW_{EL}	475	—	225	—	ns
3	Pulse Width, E High PW_{EH} = 1/2 t_{cyc} –30 ns	PW_{EH}	470	—	220	—	ns
4A 4B	E and AS Rise Time E and AS Fall Time	t_r t_f	— —	25 25	— —	25 25	ns ns
9	Address Hold Time t_{AH} = 1/8 t_{cyc} –30 ns (Note 1a)	t_{AH}	95	—	33	—	ns
12	Nonmultiplexed Address Valid Time to E Rise t_{AV} = PW_{EL} –(t_{ASD} + 80 ns) (Note 1a)	t_{AV}	275	—	88	—	ns
17	Read Data Setup Time	t_{DSR}	30	—	30	—	ns
18	Read Data Hold Time (Max = t_{MAD})	t_{DHR}	0	150	0	88	ns
19	Write Data Delay Time t_{DDW} = 1/8 t_{cyc} + 70 ns (Note 1a)	t_{DDW}	—	195	—	133	ns
21	Write Data Hold Time t_{DHW} = 1/8 t_{cyc} –30 ns (Note 1a)	t_{DHW}	95	—	33	—	ns
22	Multiplexed Address Valid Time to E Rise t_{AVM} = PW_{EL} –(t_{ASD} + 90 ns) (Note 1a)	t_{AVM}	265	—	78	—	ns
24	Multiplexed Address Valid Time to AS Fall t_{ASL} = PW_{ASH} –70 ns	t_{ASL}	150	—	25	—	ns
25	Multiplexed Address Hold Time t_{AHL} = 1/8 t_{cyc} –30 ns (Note 1b)	t_{AHL}	95	—	33	—	ns
26	Delay Time, E to AS Rise t_{ASD} = 1/8 t_{cyc} –5 ns (Note 1a)	t_{ASD}	120	—	58	—	ns
27	Pulse Width, AS High PW_{ASH} = 1/4 t_{cyc} –30 ns	PW_{ASH}	220	—	95	—	ns
28	Delay Time, AS to E Rise t_{ASED} = 1/8 t_{cyc} –5 ns (Note 1b)	t_{ASED}	120	—	58	—	ns
29	MPU Address Access Time (Note 1a) $t_{ACCA} = t_{cyc} -(PW_{EL}-t_{AVM}) -t_{DSR}-t_f$	t_{ACCA}	735	—	298	—	ns
35	MPU Access Time $t_{ACCE} = PW_{EH} - t_{DSR}$	t_{ACCE}	—	440	—	190	ns
36	Multiplexed Address Delay (Previous Cycle MPU Read) $t_{MAD} = t_{ASD}$ + 30 ns (Note 1a)	t_{MAD}	150	—	88	—	ns

NOTES:

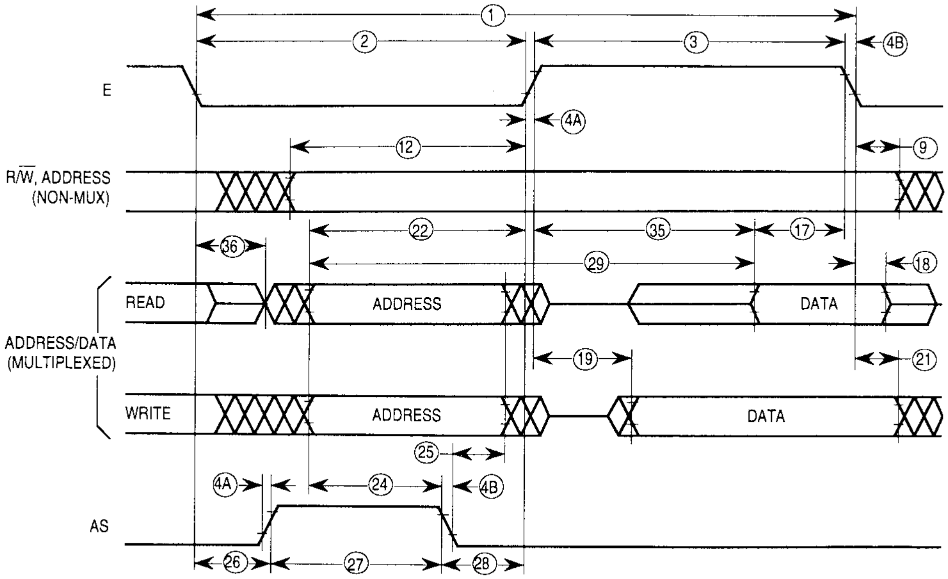
1. Input clocks with duty cycles other than 50% affect bus performance. Timing parameters affected by input clock duty cycle are identified by (a) and (b). To recalculate the approximate bus timing values, substitute the following expressions in place of 1/8 t_{cyc} in the above formulas, where applicable:

   (a) (1–DC) × 1/4 t_{cyc}

   (b) DC × 1/4 t_{cyc}

   Where:

   DC is the decimal value of duty cycle percentage (high time).
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.



NOTE: Measurement points shown are 20% and 70% of V_{DD}.

MUX BUS TIM

Figure A–14 Multiplexed Expansion Bus Timing Diagram

Table A–8 Serial Peripheral Interface Timing

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

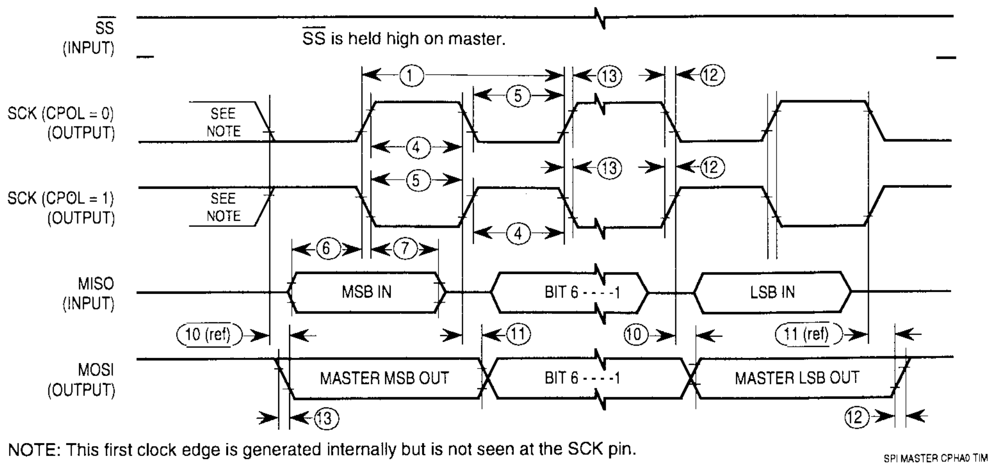
Num	Characteristic	Symbol	2.0 MHz		3.0 MHz		Unit
			Min	Max	Min	Max	
	Operating Frequency						
	Master	$f_{op(m)}$	dc	0.5	dc	0.5	f_{op}
	Slave	$f_{op(s)}$	dc	2.0	dc	3.0	MHz
1	Cycle Time						
	Master	$t_{cyc(m)}$	2.0	—	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	500	—	333	—	ns
2	Enable Lead Time						
	Master (Note 2)	$t_{lead(m)}$	—	—	—	—	ns
	Slave	$t_{lead(s)}$	250	—	240	—	ns
3	Enable Lag Time						
	Master (Note 2)	$t_{lag(m)}$	—	—	—	—	ns
	Slave	$t_{lag(s)}$	250	—	240	—	ns
4	Clock (SCK) High Time						
	Master	$t_{w(SCKH)m}$	340	—	227	—	ns
	Slave	$t_{w(SCKH)s}$	190	—	127	—	ns
5	Clock (SCK) Low Time						
	Master	$t_{w(SCKL)m}$	340	—	227	—	ns
	Slave	$t_{w(SCKL)s}$	190	—	127	—	ns
6	Data Setup Time (Inputs)						
	Master	$t_{su(m)}$	100	—	100	—	ns
	Slave	$t_{su(s)}$	100	—	100	—	ns
7	Data Hold Time (Inputs)						
	Master	$t_{h(m)}$	100	—	100	—	ns
	Slave	$t_{h(s)}$	100	—	100	—	ns
8	Access Time (Time to Data Active from High-Impedance State)						
	Slave	t_a	0	120	0	120	ns
9	Disable Time (Hold Time to High-Impedance State)						
	Slave	t_{dis}	—	240	—	167	ns
10	Data Valid (After Enable Edge) (Note 3)	$t_{v(s)}$	—	240	—	167	ns
11	Data Hold Time (Outputs) (After Enable Edge)	t_{ho}	0	—	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)						
	SPI Outputs (SCK, MOSI, and MISO)	t_{rm}	—	100	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, and $\overline{SS}$)	t_{rs}	—	2.0	—	2.0	µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF)						
	SPI Outputs (SCK, MOSI, and MISO)	t_{fm}	—	100	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, and $\overline{SS}$)	t_{fs}	—	2.0	—	2.0	µs

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 SCK, MOSI, and MISO pins.

Table A–8a Serial Peripheral Interface Timing (MC68L11E9)

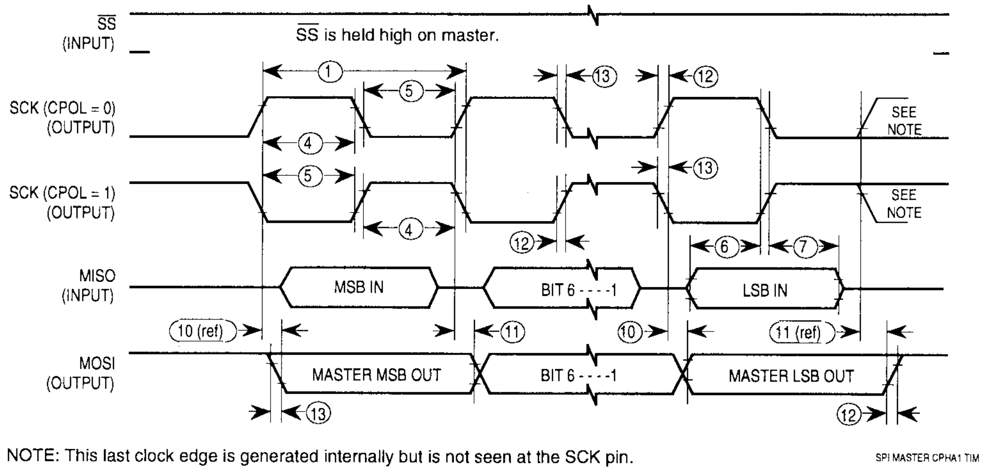
Num	Characteristic	Symbol	1.0 MHz		2.0 MHz		Unit
			Min	Max	Min	Max	
	Operating Frequency						
	Master	$f_{op(m)}$	dc	0.5	dc	0.5	f_{op}
	Slave	$f_{op(s)}$	dc	1.0	dc	2.0	MHz
1	Cycle Time						
	Master	$t_{cyc(m)}$	2.0	—	2.0	—	t_{cyc}
	Slave	$t_{cyc(s)}$	1000	—	500	—	ns
2	Enable Lead Time						
	Master (Note 2)	$t_{lead(m)}$	—	—	—	—	ns
	Slave	$t_{lead(s)}$	500	—	250	—	ns
3	Enable Lag Time						
	Master (Note 2)	$t_{lag(m)}$	—	—	—	—	ns
	Slave	$t_{lag(s)}$	500	—	250	—	ns
4	Clock (SCK) High Time						
	Master	$t_{w(SCKH)m}$	680	—	340	—	ns
	Slave	$t_{w(SCKH)s}$	380	—	190	—	ns
5	Clock (SCK) Low Time						
	Master	$t_{w(SCKL)m}$	680	—	340	—	ns
	Slave	$t_{w(SCKL)s}$	380	—	190	—	ns
6	Data Setup Time (Inputs)						
	Master	$t_{su(m)}$	100	—	100	—	ns
	Slave	$t_{su(s)}$	100	—	100	—	ns
7	Data Hold Time (Inputs)						
	Master	$t_{h(m)}$	100	—	100	—	ns
	Slave	$t_{h(s)}$	100	—	100	—	ns
8	Access Time (Time to Data Active from High-Impedance State) Slave	t_a	0	120	0	120	ns
9	Disable Time (Hold Time to High-Impedance State) Slave	t_{dis}	—	240	—	240	ns
10	Data Valid (After Enable Edge) (Note 3)	$t_{v(s)}$	—	240	—	240	ns
11	Data Hold Time (Outputs) (After Enable Edge)	t_{ho}	0	—	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF)						
	SPI Outputs (SCK, MOSI, and MISO)	t_{rm}	—	100	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, and $\overline{SS}$)	t_{rs}	—	2.0	—	2.0	µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF)						
	SPI Outputs (SCK, MOSI, and MISO)	t_{fm}	—	100	—	100	ns
	SPI Inputs (SCK, MOSI, MISO, and $\overline{SS}$)	t_{fs}	—	2.0	—	2.0	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 100 pF load on all SPI pins.



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

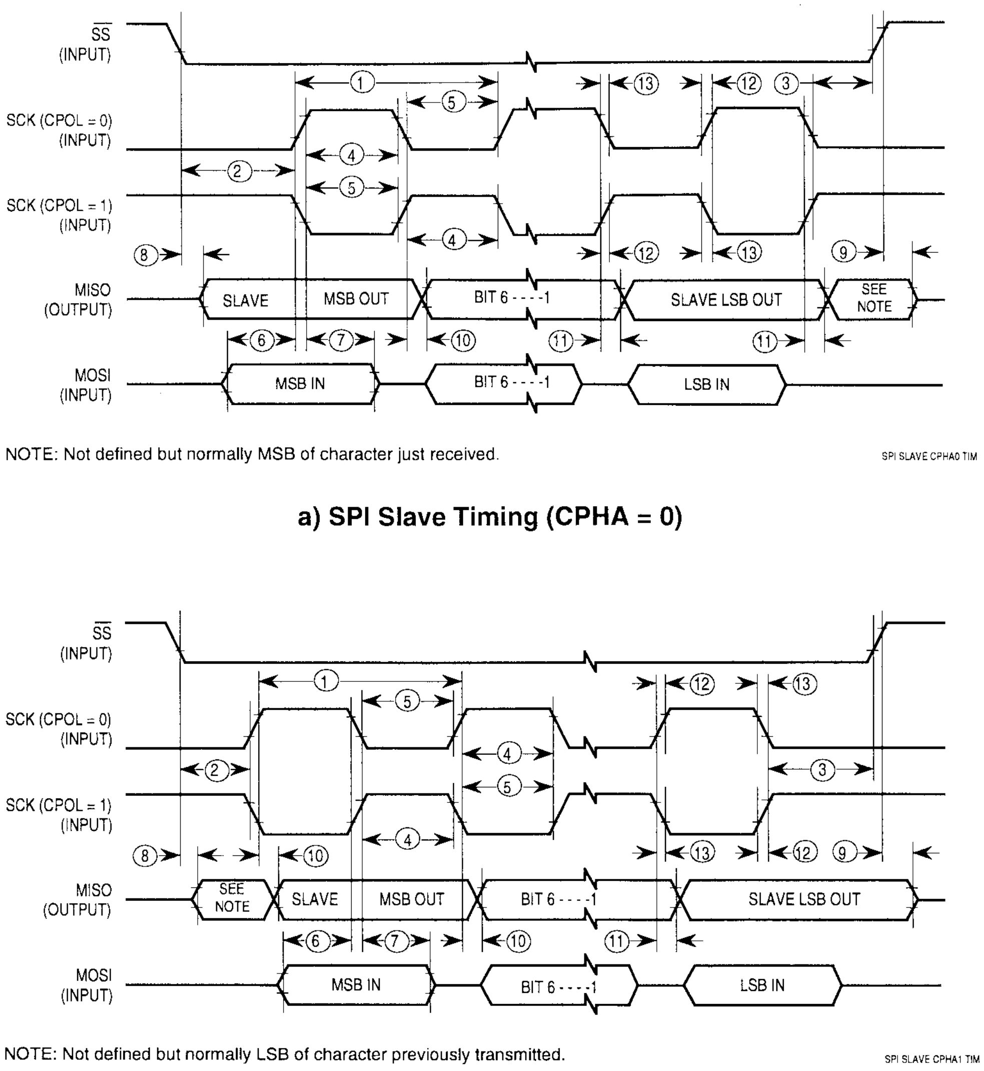
a) SPI Master Timing (CPHA = 0)



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

b) SPI Master Timing (CPHA = 1)

Figure A–15 SPI Timing Diagram (1 of 2)



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

a) SPI Slave Timing (CPHA = 0)

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

b) SPI Slave Timing (CPHA = 1)

Figure A–15 SPI Timing Diagram (2 of 2)

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
	–40 to 85°C	–40 to 105°C	–40 to 125°C	
Programming Time <1.0 MHz, RCO Enabled	10	15	20	ms
(Note 1) 1.0 to 2.0 MHz, RCO Disabled	20	Must use RCO	Must use RCO	
≥2.0 MHz (or Anytime RCO Enabled)	10	15	20	
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.

Table A–9a EEPROM Characteristics (MC68L11E9)

V_{DD} = 3.0 Vdc to 5.5 Vdc, V_{SS} = 0 Vdc, T_A = T_L to T_H

Characteristic		Temperature Range	Unit
		–20 to 70°C	
Programming Time	3 V, E ≤ 2.0 MHz, RCO Enabled	25	ms
(Note 1)	5 V, E ≤ 2.0 MHz, RCO Enabled	10	ms
Erase Time (Byte, Row and Bulk)	3 V, E ≤ 2.0 MHz, RCO Enabled	25	ms
(Note 1)	5 V, E ≤ 2.0 MHz, RCO Enabled	10	ms
Write/Erase Endurance (Note 2)		10,000	Cycles
Data Retention (Note 2)		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.
2. Refer to Reliability Monitor Report (current quarterly issue) for current failure rate information.

APPENDIX B
MECHANICAL DATA AND ORDERING INFORMATION

M68HC11 E-series microcontrollers are available in 52-pin plastic leaded chip carrier (PLCC), 52-pin windowed ceramic leaded chip carrier (CLCC), 64-pin quad flat pack (QFP), 52-pin thin quad flat pack (TQFP), and 56-pin dual in-line package with .070" lead spacing (SDIP). In addition to these packages, the MC68HC811E2 is available in a 48-pin plastic DIP (.100" lead spacing). Refer to **Table B–1**, **Table B–2**, and **Table B–3** for ordering information.

B.1 Ordering Information

Use the information in **Table B–1**, **Table B–2**, and **Table B–3** to specify the appropriate device when placing an order.

Table B–1 Standard Device Ordering Information

Package	Description	CONFIG	Temperature	Frequency	MC Order Number
52-Pin PLCC	BUFFALO ROM	$0F	–40° to +85° C	2 MHz	MC68HC11E9BCFN2
				3 MHz	MC68HC11E9BCFN3
	No ROM	$0D	–40° to +85° C	2 MHz	MC68HC11E1CFN2
				3 MHz	MC68HC11E1CFN3
			–40° to +105° C	2 MHz	MC68HC11E1VFN2
			–40° to +125° C	2 MHz	MC68HC11E1MFN2
	No ROM, No EEPROM	$0C	–40° to +85° C	2 MHz	MC68HC11E0CFN2
				3 MHz	MC68HC11E0CFN3
			–40° to +105° C	2 MHz	MC68HC11E0VFN2
			–40° to +125° C	2 MHz	MC68HC11E0MFN2
	OTPROM	$0F	–40° to +85° C	2 MHz	MC68HC711E9CFN2
				3 MHz	MC68HC711E9CFN3
			–40° to +105° C	2 MHz	MC68HC711E9VFN2
			–40° to +125° C	2 MHz	MC68HC711E9MFN2
	OTPROM, Enhanced Security Feature	$0F	–40° to +85° C	2 MHz	MC68S711E9CFN2
	20 Kbytes OTPROM	$0F	0° to +70° C	3 MHz	MC68HC711E20FN3
			–40° to +85° C	2 MHz	MC68HC711E20CFN2
				3 MHz	MC68HC711E20CFN3
			–40° to +105° C	2 MHz	MC68HC711E20VFN2
			–40° to +125° C	2 MHz	MC68HC711E20MFN2
	No ROM, 2 Kbytes EEPROM	$FF	0° to +70° C	2 MHz	MC68HC811E2FN2
			–40° to +85° C	2 MHz	MC68HC811E2CFN2
			–40° to +105° C	2 MHz	MC68HC811E2VFN2
			–40° to +125° C	2 MHz	MC68HC811E2MFN2
64-Pin QFP	BUFFALO ROM	$0F	–40° to +85° C	2 MHz	MC68HC11E9BCFU2
				3 MHz	MC68HC11E9BCFU3
	No ROM	$0D	–40° to +85° C	2 MHz	MC68HC11E1CFU2
				3 MHz	MC68HC11E1CFU3
			–40° to +105° C	2 MHz	MC68HC11E1VFU2
	No ROM, No EEPROM	$0C	–40° to +85° C	2 MHz	MC68HC11E0CFU2
			–40° to +105° C	2 MHz	MC68HC11E0VFU2
	20 Kbytes OTPROM	$0F	0° to +70° C	3 MHz	MC68HC711E20FU3
			–40° to +85° C	2 MHz	MC68HC711E20CFU2
				3 MHz	MC68HC711E20CFU3
			–40° to +105° C	2 MHz	MC68HC711E20VFU2
			–40° to +125° C	2 MHz	MC68HC711E20MFU2
52-Pin TQFP (10 mm X 10 mm)	BUFFALO ROM	$0F	–40° to +85° C	2 MHz	MC68HC11E9BCPB2
				3 MHz	MC68HC11E9BCPB3

Table B–1 Standard Device Ordering Information (Continued)

Package	Description	CONFIG	Temperature	Frequency	MC Order Number
52-Pin CLCC (Windowed)	EPROM	$0F	–40° to +85° C	2 MHz	MC68HC711E9CFS2
				3 MHz	MC68HC711E9CFS3
			–40° to +105° C	2 MHz	MC68HC711E9VFS2
			–40° to +125° C	2 MHz	MC68HC711E9VFS2
	20 Kbytes EPROM	$0F	0° to +70° C	3 MHz	MC68HC711E20FS3
			–40° to +85° C	2 MHz	MC68HC711E20CFS2
				3 MHz	MC68HC711E20CFS3
			–40° to +105° C	2 MHz	MC68HC711E20VFS2
			–40° to +125° C	2 MHz	MC68HC711E20MFS2
48-Pin DIP (MC68HC811 E2 only)	No ROM, 2 Kbytes EEPROM	$FF	0° to +70° C	2 MHz	MC68HC811E2P2
			–40° to +85° C	2 MHz	MC68HC811E2CP2
			–40° to +105° C	2 MHz	MC68HC811E2VP2
			–40° to +125° C	2 MHz	MC68HC811E2MP2
56-Pin SDIP (.070" Spacing)	BUFFALO ROM	$0F	–40° to +85° C	2 MHz	MC68HC11E9BCB2
				3 MHz	MC68HC11E9BCB3
	No ROM	$0D	–40° to +85° C	2 MHz	MC68HC11E1CB2
				3 MHz	MC68HC11E1CB3
			–40° to +105° C	2 MHz	MC68HC11E1VB2
			–40° to +125° C	2 MHz	MC68HC11E1MB2
	No ROM, No EEPROM	$0C	–40° to +85° C	2 MHz	MC68HC11E0CB2
				3 MHz	MC68HC11E0CB3
			–40° to +105° C	2 MHz	MC68HC11E0VB2
			–40° to +125° C	2 MHz	MC68HC11E0MB2

Table B–2 Custom ROM Device Ordering Information

Package	Description	Temperature	Frequency	MC Order Number
52-Pin PLCC	Custom ROM	0° to +70° C	3 MHz	MC68HC11E9FN3
		–40° to +85° C	2 MHz	MC68HC11E9CFN2
			3 MHz	MC68HC11E9CFN3
		–40° to +105° C	2 MHz	MC68HC11E9VFN2
		–40° to +125° C	2 MHz	MC68HC11E9MFN2
	Custom ROM, No EEPROM	0° to +70° C	3 MHz	MC68HC11E8FN3
		–40° to +85° C	2 MHz	MC68HC11E8CFN2
			3 MHz	MC68HC11E8CFN3
		–40° to +105° C	2 MHz	MC68HC11E8VFN2
		–40° to +125° C	2 MHz	MC68HC11E8MFN2
	20 Kbytes Custom ROM	0° to +70° C	3 MHz	MC68HC11E20FN3
		–40° to +85° C	2 MHz	MC68HC11E20CFN2
			3 MHz	MC68HC11E20CFN3
		–40° to +105° C	2 MHz	MC68HC11E20VFN2
		–40° to +125° C	2 MHz	MC68HC11E20MFN2
64-Pin QFP	Custom ROM	0° to +70° C	3 MHz	MC68HC11E9FU3
		–40° to +85° C	2 MHz	MC68HC11E9CFU2
			3 MHz	MC68HC11E9CFU3
		–40° to +105° C	2 MHz	MC68HC11E9VFU2
		–40° to +125° C	2 MHz	MC68HC11E9MFU2
	Custom ROM, No EEPROM	0° to +70° C	3 MHz	MC68HC11E8FU3
		–40° to +85° C	2 MHz	MC68HC11E8CFU2
			3 MHz	MC68HC11E8CFU3
		–40° to +105° C	2 MHz	MC68HC11E8VFU2
		–40° to +125° C	2 MHz	MC68HC11E8MFU2
	20 Kbytes Custom ROM	0° to +70° C	3 MHz	MC68HC11E20FU3
		–40° to +85° C	2 MHz	MC68HC11E20CFU2
			3 MHz	MC68HC11E20CFU3
		–40° to +105° C	2 MHz	MC68HC11E20VFU2
		–40° to +125° C	2 MHz	MC68HC11E20MFU2
52-Pin TQFP (10 mm X 10 mm)	Custom ROM	0° to +70° C	3 MHz	MC68HC11E9PB3
		–40° to +85° C	2 MHz	MC68HC11E9CPB2
			3 MHz	MC68HC11E9CPB3
		–40° to +105° C	2 MHz	MC68HC11E9VPB2
		–40° to +125° C	2 MHz	MC68HC11E9MPB2
	Custom ROM, No EEPROM	0° to +70° C	3 MHz	MC68HC11E8PB3
		–40° to +85° C	2 MHz	MC68HC11E8CPB2
			3 MHz	MC68HC11E8CPB3
		–40° to +105° C	2 MHz	MC68HC11E8VPB2
		–40° to +125° C	2 MHz	MC68HC11E8MPB2

Table B–2 Custom ROM Device Ordering Information (Continued)

Package	Description	Temperature	Frequency	MC Order Number
56-Pin SDIP (.070" Spacing)	Custom ROM	0° to +70° C	3 MHz	MC68HC11E9B3
		–40° to +85° C	2 MHz	MC68HC11E9CB2
			3 MHz	MC68HC11E9CB3
		–40° to +105° C	2 MHz	MC68HC11E9VB2
		–40° to +125° C	2 MHz	MC68HC11E9MB2
	Custom ROM, No EEPROM	0° to +70° C	3 MHz	MC68HC11E8B3
		–40° to +85° C	2 MHz	MC68HC11E8CB2
			3 MHz	MC68HC11E8CB3
		–40° to +105° C	2 MHz	MC68HC11E8VB2
		–40° to +125° C	2 MHz	MC68HC11E8MB2

Table B–3 Extended Voltage Device Ordering Information (3.0 Vdc to 5.5 Vdc)

Package	Description	Temperature	Frequency	MC Order Number
52-Pin PLCC	Custom ROM	–20° to +70° C	2 MHz	MC68L11E9FN2
	Custom ROM, No EEPROM		2 MHz	MC68L11E8FN2
	No ROM		2 MHz	MC68L11E1FN2
	No ROM, No EEPROM		2 MHz	MC68L11E0FN2
64-Pin QFP	Custom ROM	–20° to +70° C	2 MHz	MC68L11E9FU2
	Custom ROM, No EEPROM		2 MHz	MC68L11E8FU2
	No ROM		2 MHz	MC68L11E1FU2
	No ROM, No EEPROM		2 MHz	MC68L11E0FU2
52-Pin TQFP (10 mm X 10 mm)	Custom ROM	–20° to +70° C	2 MHz	MC68L11E9PB2
	Custom ROM, No EEPROM		2 MHz	MC68L11E8PB2
	No ROM		2 MHz	MC68L11E1PB2
	No ROM, No EEPROM		2 MHz	MC68L11E0PB2
56-Pin SDIP (.070" Spacing)	Custom ROM	–20° to +70° C	2 MHz	MC68L11E9B2
	Custom ROM, No EEPROM		2 MHz	MC68L11E8B2
	No ROM		2 MHz	MC68L11E1B2
	No ROM, No EEPROM		2 MHz	MC68L11E0B2

B.2 Obtaining M68HC11 E-Series Mechanical Information

Although all devices manufactured by Motorola conform to current JDEC standards, complete mechanical information regarding M68HC11 E-series microcontrollers is available by facsimile through Motorola's MFAX system.

Users can obtain instructions on the use of this system by calling (602) 244-6609. The automated system will request that a fax number be entered. Once entered, a facsimile message will be sent containing the instructions.

APPENDIX C
DEVELOPMENT SUPPORT

C.1 Motorola M68HC11 E-Series Development Tools

Device	Package	Emulation Module [1, 3]	Flex Cable [1, 3]	MMDS11 Target Head [1, 3]	SPGMR Programming Adapter [2]
MC68HC11E9 MC68HC711E9	52 FN	M68EM11E20	M68CBL11C	M68TC11E20FN52	M68PA11E20FN52
	52 PB	M68EM11E20	M68CBL11C	M68TC11E20PB52	M68PA11E20PB52
	56 B	M68EM11E20	M68CBL11B	M68TC11E20B56	M68PA11E20B56
	64 FU	M68EM11E20	M68CBL11C	M68TC11E20FU64	M68PA11E20FU64
MC68HC11E20 MC68HC711E20	52 FN	M68EM11E20	M68CBL11C	M68TC11E20FN52	M68PA11E20FN52
	64 FU	M68EM11E20	M68CBL11C	M68TC11E20FU64	M68PA11E20FU64
MC68HC811E2	48 P	M68EM11E20	M68CBL11B	M68TB11E20P48	M68PA11A8P48
	52 FN	M68EM11E20	M68CBL11C	M68TC11E20FN52	M68PA11E20FN52

1. Each MMDS11 system consists of a system console (M68MMDS11), an emulation module, a flex cable, and a target head.
2. Each SPGMR system consists of a universal serial programmer (M68SPGMR11) and a programming adapter. It can be used alone or in conjunction with the MMDS11.
3. A complete EVS consists of a platform board (M68HC11PFB), an emulation module, a flex cable, and a target head.

C.2 EVS — Evaluation System

The EVS is an economical tool for designing, debugging, and evaluating target systems based on the M68HC11. EVS features include the following:

- 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
  - — M68HC11 E-series user map that includes 64 Kbytes of emulation RAM
- 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 Motorola Modular Development System (MMDS11)

The M68MMDS11 Motorola Modular Development System (MMDS11) is an emulator system for developing embedded systems based on an M68HC11 microcontroller unit (MCU). The MMDS11 provides a bus state analyzer (BSA) and real-time memory windows. The unit's integrated development environment includes an editor, an assembler, user interface, and source-level debug. These features significantly reduce the time necessary to develop and debug an embedded MCU system. The unit's compact size requires a minimum of desk space.

The MMDS11 is one component of Motorola's modular approach to MCU-based product development. This modular approach allows easy configuration of the MMDS11 to fit a wide range of requirements. It also reduces development system cost by allowing the user to purchase only the modular components necessary to support the particular MCU derivative.

MMDS11 features include:

- Real-time, non-intrusive, in-circuit emulation at the MCU's operating frequency
- Real-time bus state analyzer
  - 8K x 64 real-time trace buffer
  - Display of real-time trace data as raw data, disassembled instructions, raw data and disassembled instructions, or assembly-language source code
  - Four hardware triggers for commencing trace and to provide breakpoints
  - Nine triggering modes
  - As many as 8190 pre- or post-trigger points for trace data
  - 16 general-purpose logic clips, four of which can be used to trigger the bus state analyzer sequencer
  - 16-bit time tag, or an optional 24-bit time tag that reduces the logic clips traced from sixteen to eight
- Four data breakpoints (hardware breakpoints)
- Hardware instruction breakpoints over either the 64 Kbyte M68HC11 memory map, or over a 1 Mbyte bank switched memory map
- Thirty-two real-time variables, nine of which can be displayed in the variables window. These variables may be read or written while the MCU is running
- Thirty-two bytes of real-time memory can be displayed in the memory window. This memory may be read or written while the MCU is running
- 64 Kbytes of fast emulation memory (SRAM)
- Current-limited target input/output connections
- Six software-selectable oscillator clock sources: five internally generated frequencies and an external frequency via a bus analyzer logic clip
- Command and response logging to MS-DOS disk files to save session history
- SCRIPT command for automatic execution of a sequence of MMDS11 commands
- Assembly or C-language source-level debugging with global variable viewing
- Host/emulator communications speeds as high as 57600 baud for quick program loading

- Extensive on-line MCU information via the CHIPINFO command. View memory-map, vectors, register, and pin-out information pertaining to the device being emulated
- Host software supports:
  - — An editor
  - — An assembler and user interface
  - — Source-level debug
  - — Bus state analysis
  - — IBM mouse

C.4 SPGMR11— Serial Programmer for M68HC11 MCUs

The SPGMR11 is a modular EPROM/EEPROM programming tool for all M68HC11 devices. The programmer features interchangeable adapters that allow programming of various M68HC11 package types.

- Programs M68HC11 family devices that contain an EPROM or EEPROM array.
- Can be operated as a stand-alone programmer connected to a host computer, or connected between a host computer and the M68HC11 Modular Development System (MMDS11) station module.
- Uses plug-in programming adapters to accommodate a variety of MCU devices and packages.
- On-board programming voltage circuit eliminates the need for an external 12-volt supply.
- Includes programming software and a user's manual.
- Includes a +5 volt power cable and a DB9 to DB25 connector adapter.

SUMMARY OF CHANGES

This is a complete revision, with complete reprint. All known errors in this publication have been corrected.The following summary lists significant changes. Typographical errors which do not affect content are not noted.

Section 2 Pin Descriptions

Page 2–6 Modified Figure 2–6 External Reset circuit

Page 2–7 Added Figure 2–7 External Reset Circuit with Delay.

Page 2–8 Corrected Figure 2–9 External Oscillator Connections

Section 4 Operating Modes and On-Chip Memory

Page 4–2 Added $\overline{\text{OE}}$ connection in Figure 4–1 Address/Data Demultiplexing

Page 4–10 Changed reset value of IRV(NE) in Test mode from 0 to 1

Page 4–11 Table 4–3 all bits in INIT register can be set or cleared at anytime

Page 4–16 Removed paragraph 4.3.1.1 EPROM Emulation (PROG) Mode (PROG mode not available)

Section 5 Resets and Interrupts

Page 5–3 Added information about using the STOP instruction as a software initiated reset

Appendix B Mechanical Data and Ordering Information

Pages B–6 – B–11 Removed Case Outline Drawings; added information about using Motorola's MFAX system

Appendix C Development Support

Completely revised and updated information on available development tools.

This book is a product of the Motorola Advanced Microcontroller Documentation Group in Oak Hill, Texas. It was written and illustrated by Gene Bates, and edited by Marilou Groves. Camera-ready copy and line art were produced with Framemaker 4 running on Macintosh computers. Pre-press work on the cover as well as the printing was performed by Imperial Lithographics, Phoenix, Arizona, under the auspices of the Motorola Semiconductor Products Sector Marketing Services.

How to reach us:
USA/EUROPE/Locations Not Listed: Motorola Literature Distribution;
P.O. Box 5405, Denver, Colorado 80217. 303–675–2140 or 1–800–441–2447

Mfax™: RMFAX0@email.sps.mot.com – TOUCHTONE 602–244–6609
– US & Canada ONLY 1–800–774–1848
INTERNET: http://motorola.com/sps

JAPAN: Nippon Motorola Ltd.: SPD, Strategic Planning Office, 4–32–1,
Nishi–Gotanda, Shinagawa–ku, Tokyo 141, Japan. 81–3–5487–8488

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park,
51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852–26629298

MOTOROLA

1ATX31388-10 PRINTED IN USA 3/98 IMPERIAL LITHO 34604 5,000 LITCSG2