MOTOROLA
intelligence everywhere™

digital dna™

MC68HC711D3
MC68HC11D3
MC68HC11D0
MC68L11D0

Data Sheet

M68HC11 Microcontrollers

MC68HC711D3/D
Rev. 2
9/2003

MOTOROLA.COM/SEMICONDUCTORS

MC68HC711D3

Data Sheet

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

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

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

Revision History

Date	Revision Level	Description	Page Number(s)
September, 2003	2	Reformatted to curent publications standards	N/A
		Removed references to PROG mode.	Throughout
		Corrected pin assignments for: **Figure 1-2. Pin Assignments for 40-Pin Plastic DIP** **Figure 1-3. Pin Assignments for 44-Pin PLCC** Added **Figure 1-4. Pin Assignments for 44-Pin QFP**	 15 15 16
		1.9 Interrupt Request (IRQ) — Reworked description for clarity.	18
		2.4 Programmable Read-Only Memory (PROM) — Updated with additional data.	31
		Section 10. Ordering Information and Mechanical Specifications — Added mechanical specifications for 44-pin plastic quad flat pack (QFP).	133
		Added the following appendices: **Appendix A. MC68HC11D3 and MC68HC11D0** **Appendix B. MC68L11D0**	 137 143

List of Sections

Table of Contents

Section 9. Electrical Characteristics

Section 10. Ordering Information and Mechanical Specifications

Appendix A. MC68HC11D3 and MC68HC11D0

Appendix B. MC68L11D0

Section 1. General Description

1.1 Introduction

This section depicts the general characteristics and features of the MC68HC711D3 high-density complementary metal-oxide semiconductor (HCMOS) microcontroller unit (MCU).

The MC68HC711D3 contains highly sophisticated on-chip peripheral functions. This high-speed, low-power programmable read-only memory (PROM) MCU has a nominal bus speed of 3 MHz. The fully static design allows operations at frequencies down to dc.

The MC68HC11D3 and MC68HC11D0 are read-only memory (ROM) based high-performance microcontrollers (MCU) based on the MC68HC11E9 design. The MC68L11D0 is an extended-voltage version of the MC68HC11D0 that can operate in applications that require supply voltages as low as 3.0 V. The information in this document pertains to all the devices with the exceptions noted in **Appendix A. MC68HC11D3 and MC68HC11D0** and **Appendix B. MC68L11D0**.

1.2 Features

Features of the MC68HC711D3 include:

- Expanded 16-bit timer system with four-stage programmable prescaler
- Non-return-to-zero (NRZ) serial communications interface (SCI)
- Power-saving stop and wait modes
- 64 Kbytes memory addressability
- Multiplexed address/data bus
- Serial peripheral interface (SPI)
- 4 Kbytes of one-time programmable read-only memory (OTPROM)
- 8-bit pulse accumulator circuit
- 192 bytes of static random-access memory (RAM) (all saved during standby)
- Real-time interrupt (RTI) circuit
- Computer operating properly (COP) watchdog system

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

1.3 Structure

Refer to **Figure 1-1**, which shows the structure of the MC68HC711D3 MCU.

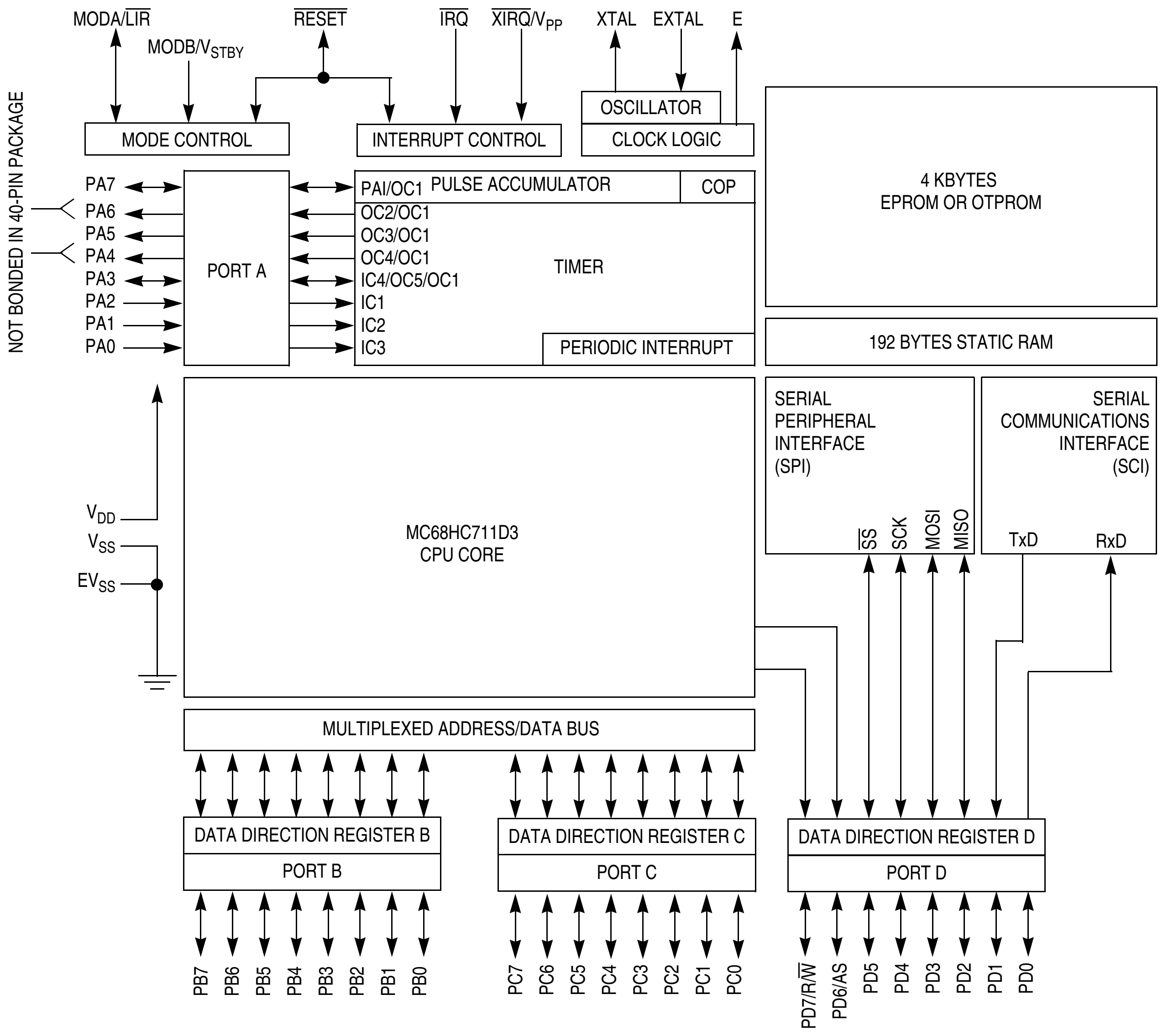


Figure 1-1. MC68HC711D3 Block Diagram

1.4 Pin Descriptions

Refer to **Figure 1-2**, **Figure 1-3**, and **Figure 1-4** for pin assignments.

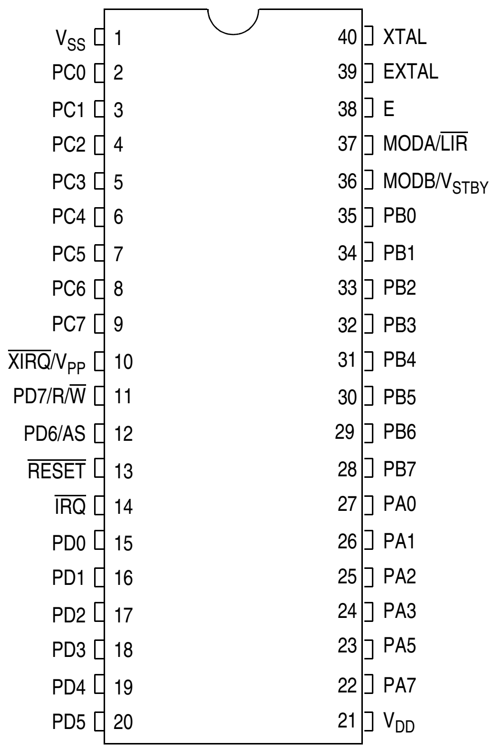


Figure 1-2. Pin Assignments for 40-Pin Plastic DIP

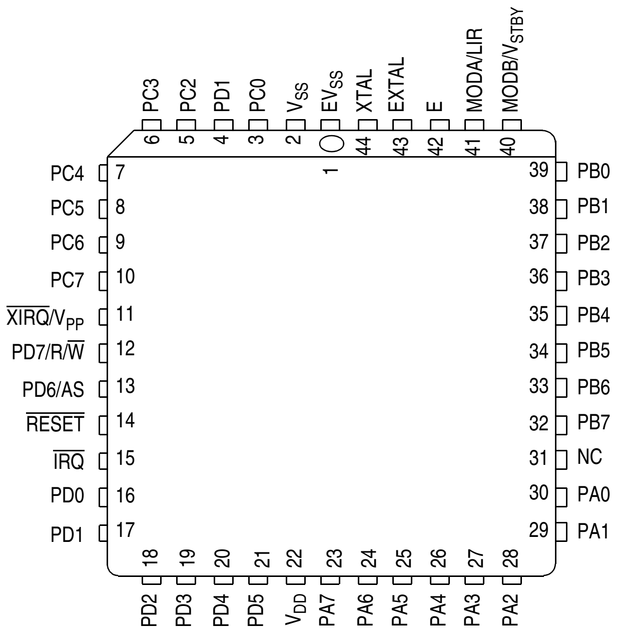


Figure 1-3. Pin Assignments for 44-Pin PLCC

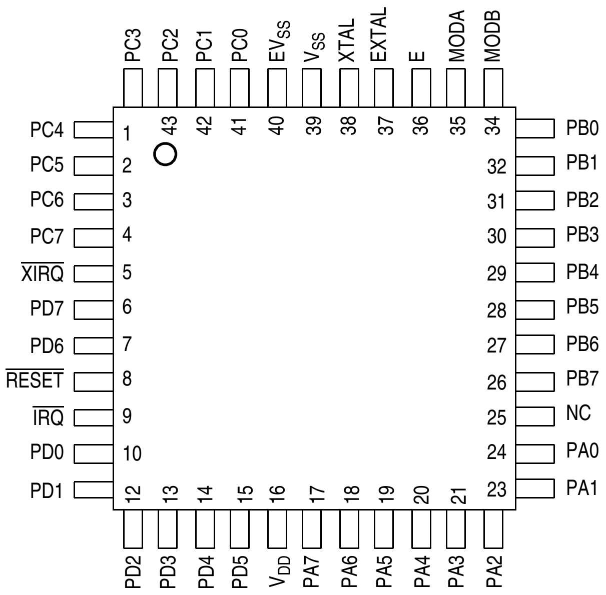


Figure 1-4. Pin Assignments for 44-Pin QFP

1.5 Power Supply (V_{DD}, V_{SS}, and EV_{SS})

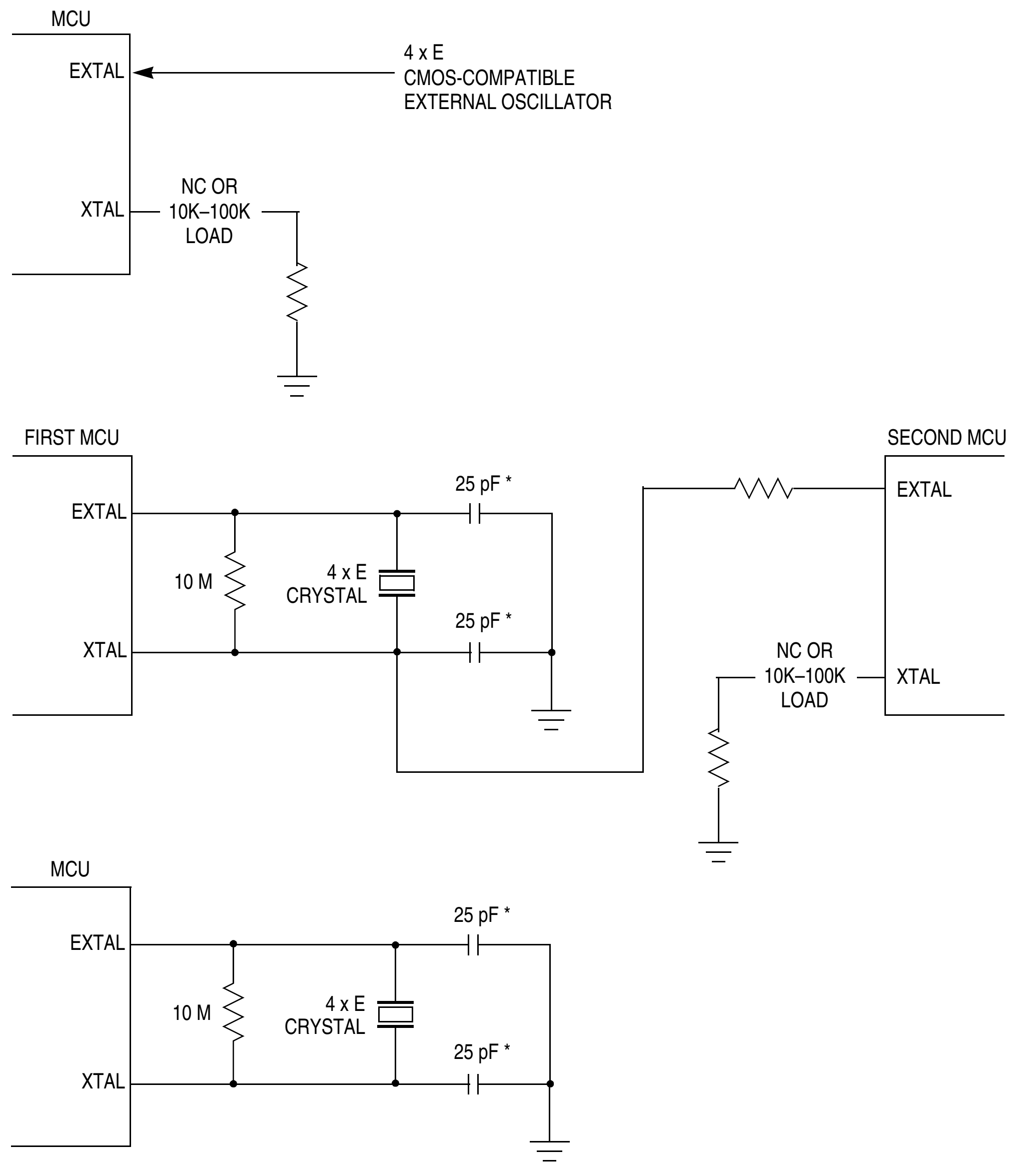
Power is supplied to the MCU through V_{DD} and V_{SS}. V_{DD} is the power supply (+5 V ±10%) and V_{SS} is ground (0 V). EV_{SS}, available on the 44-pin PLCC and QFP, is an additional ground pin.

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

An active low 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 computer operating properly (COP) watchdog circuit. In addition, the state of this pin is one of the factors governing the selection of BOOT mode.

1.7 Crystal Driver and External Clock Input (XTAL and 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. Refer to **Figure 1-5** for crystal and clock connections.



* Values includes all stray capacitances.

Figure 1-5. Oscillator Connections

1.8 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. The E clock can be turned off in single-chip mode for greater noise immunity if desired. See **4.3.6 Highest Priority I Interrupt and Miscellaneous Register (HPRIO)** for details.

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

The $\overline{\text{IRQ}}$ input provides a means of applying asynchronous interrupt requests to the microcontroller unit (MCU). Either negative edge-sensitive triggering or level-sensitive triggering is program selectable by using the IRQE bit of the OPTION register. $\overline{\text{IRQ}}$ is always configured to level-sensitive triggering at reset.

While the programmable read-only memory (PROM) is being programmed, this pin provides the chip enable ($\overline{\text{CE}}$) signal. To prevent accidental programming of the PROM during reset, an external resistor is required on $\overline{\text{IRQ}}$ to pull the pin to V_{DD}.

1.10 Non-Maskable Interrupt/Programming Voltage ($\overline{\text{XIRQ}}$/V_{PP})

The $\overline{\text{XIRQ}}$ input provides the capability for asynchronously applying non-maskable interrupts to the MCU after a power-on reset (POR). During reset, the X bit in the condition code register (CCR) is set masking any interrupt until enabled by software. This level-sensitive input requires an external pullup resistor to V_{DD}.

In the programming configuration of the bootstrap mode, this pin is used to supply one-time programmable read-only memory (OTPROM) programming voltage, V_{PP}, to the MCU. To avoid programming accidents during reset, this pin should be equal to V_{DD} during normal operation unless $\overline{\text{XIRQ}}$ is active.

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

As reset transitions, these pins are used to latch the part into one of the four central processor unit (CPU) controlled modes of operation. The $\overline{\text{LIR}}$ output can be used as an aid to debugging once reset is completed. The open-drain $\overline{\text{LIR}}$ pin goes to an active low during the first E-clock cycle of each instruction and remains low for the duration of that cycle. The V_{STBY} input is used to retain random-access memory (RAM) contents during power down.

1.12 Read/Write (R/$\overline{W}$)

This pin performs either of two separate functions, depending on the operating mode.

- In single-chip and bootstrap modes, R/$\overline{W}$ functions as input/output port D bit 7. Refer to **Section 5. Input/Output (I/O) Ports** for further information.
- In expanded multiplexed and test modes, R/$\overline{W}$ performs a read/write function. R/$\overline{W}$ controls the direction of transfers on the external data bus.

1.13 Port D Bit 6/Address Strobe (PD6/AS)

This pin performs either of two separate functions, depending on the operating mode.

- In single-chip and bootstrap modes, the pin functions as input/output port D bit 6.
- In the expanded multiplexed and test modes, it provides an address strobe (AS) function. AS is used to demultiplex the address and data signals at port C.

Refer to **Section 2. Operating Modes and Memory** for further information.

1.14 Input/Output Lines (PA7–PA0, PB7–PB0, PC7–PC0, and PD7–PD0)

In the 44-pin PLCC package, 32 input/output lines are arranged into four 8-bit ports: A, B, C, and D. The lines of ports B, C, and D are fully bidirectional. Port A has two bidirectional, three input-only, and three output-only lines in the 44-pin PLCC packaging. In the 40-pin DIP, two of the output-only lines are not bonded.

Each of these four ports serves a purpose other than input/output (I/O), depending on the operating mode or peripheral functions selected.

NOTE: *Ports B, C, and two bits of port D are available for I/O functions only in single-chip and bootstrap modes.*

Refer to **Table 1-1** for details about the functions of the 32 port signals within different operating modes.

Table 1-1. Port Signal Functions

Port/Bit	Single-Chip and Bootstrap Mode	Expanded Multiplexed and Special Test Mode
PA0	PA0/IC3	
PA1	PA1/IC2	
PA2	PA2/IC1	
PA3	PA3/OC5/IC4/and-or OC1	
PA4[1]	PA4/OC4/and-or OC1	
PA5	PA5/OC3/and-or OC1	
PA6[1]	PA6/OC2/and-or OC1	
PA7	PA7/PAI/and-or OC1	
PB0	PB0	A8
PB1	PB1	A9
PB2	PB2	A10
PB3	PB3	A11
PB4	PB4	A12
PB5	PB5	A13
PB6	PB6	A14
PB7	PB7	A15
PC0	PC0	A0/D0
PC1	PC1	A1/D1
PC2	PC2	A2/D2
PC3	PC3	A3/D3
PC4	PC4	A4/D4
PC5	PC5	A5/D5
PC6	PC6	A6/D6
PC7	PC7	A7/D7
PD0	PD0/RxD	
PD1	PD1/TxD	
PD2	PD2/MISO	
PD3	PD3/MOSI	
PD4	PD4/SCK	
PD5	PD5/$\overline{\text{SS}}$	
PD6	PD6	AS
PD7	PD7	R/$\overline{\text{W}}$

1. In the 40-pin package, pins PA4 and PA6 are not bonded. Their associated I/O and output compare functions are not available externally. They can still be used as internal software timers, however.

Section 2. Operating Modes and Memory

2.1 Introduction

This section contains information about:

- The modes that define MC68HC711D3 operating conditions
- The on-chip memory that allows the microcontroller unit (MCU) to be configured for various applications
- The 4-Kbytes of programmable read-only memory (PROM)

2.2 Operating Modes

The MC68HC711D3 uses two dedicated pins, MODA and MODB, to select one of two normal operating modes or one of two special operating modes. A value reflecting the microcontroller unit (MCU) status or mode selected is latched on bits SMOD and MDA of the highest priority I-bit interrupt and miscellaneous register (HPRIO) on the rising edge of reset. The normal operating modes are the single-chip and expanded-multiplexed modes. The special operating modes are the bootstrap and test modes. **Table 2-1** shows mode selection according to the values encoded on the MODA and MODB pins, and the value latched in the SMOD and MDA bits.

Table 2-1. Mode Selection

RESET	MODA	MODB	Mode Selected	SMOD	MDA
1	0	1	Normal — single chip	0	0
1	1	1	Normal — expanded multiplexed	0	1
1	0	0	Special — bootstrap (BOOT)	1	0
1	1	0	Special — test	1	1
0	0	0	Reserved	X	X

2.2.1 Single-Chip Mode

In single-chip mode, the MCU functions as a self-contained microcontroller and has no external address or data bus. The 4-Kbyte erasable programmable read-only memory (EPROM) would contain all program code and is located at \$F000–\$FFFF. This mode provides maximum use of the pins for on-chip peripheral functions, and all the address and data activity occurs within the MCU.

2.2.2 Expanded Multiplexed Mode

In the expanded-multiplexed mode, the MCU can address up to 64 Kbytes of address space. High-order address bits are output on the port B pins. Low-order address bits and the bidirectional data bus are multiplexed on port C. The AS pin provides the control output used in demultiplexing the low-order address. The R/$\overline{\text{W}}$ pin is used to control the direction of data transfer on the port C bus.

If this mode is entered out of reset, the EPROM is located at \$7000–\$7FFF and vector accesses are from external memory. To be in expanded-multiplexed mode with EPROM located at \$F000–\$FFFF, it is necessary to start in single-chip mode, executing out of EPROM, and then set the MDA bit of the HPRIO register to switch mode.

NOTE: *R/$\overline{\text{W}}$, AS, and the high-order address bus (port B) are inputs in single-chip mode. These inputs may need to be pulled up so that off-chip accesses cannot occur while the MCU is in single-chip mode.*

2.2.3 Special Bootstrap Mode (BOOT)

This special mode is similar to single-chip mode. The resident bootloader program contains a 256-byte program in a special on-chip read-only memory (ROM). The user downloads a small program into on-board RAM using the SCI port. Program control is passed to RAM when an idle line of at least four characters occurs. In this mode, all interrupt vectors are mapped to RAM (see **Table 2-2**), so that the user can set up a jump table, if desired.

Bootstrap mode (BOOT) is entered out of reset if the voltage level on both MODA and MODB is low. The programming aspect of bootstrap mode, used to program the one-time programmable ROM (OTPROM) through the MCU, is entered automatically if $\overline{\text{IRQ}}$ is low and programming voltage is available on the V_{PP} pin. $\overline{\text{IRQ}}$ should be pulled up while in reset with MODA and MODB configured for bootstrap mode to prevent unintentional programming of the EPROM.

This versatile mode (BOOT) can be used for test and diagnostic functions on completed modules and for programming the on-board PROM. The serial receive logic is initialized by software in the bootloader ROM, which provides program control for the SCI baud rate and word format. Mode switching to other modes can occur under program control by writing to the SMOD and MDA bits of the HPRIO register. Two special bootloader functions allow either an immediate jump-to-RAM at memory address \$0000 or an immediate jump-to-EPROM at \$F000.

Table 2-2. Bootstrap Mode Jump Vectors

Address	Vector
00C4	SCI
00C7	SPI
00CA	Pulse accumulator input edge
00CD	Pulse accumulator overflow
00D0	Timer overflow
00D3	Timer output compare 5/input capture 4
00D6	Timer output compare 4
00D9	Timer output compare 3
00DC	Timer output compare 2
00DF	Timer output compare 1
00E3	Timer input capture 3
00E5	Timer input capture 2
00E8	Timer input capture 1
00EB	Real-time interrupt
00EE	IRQ
00F1	XIRQ
00F4	SWI
00F7	Illegal opcode
00FA	COP fail
00FD	Clock monitor
BF00 (Boot)	Reset

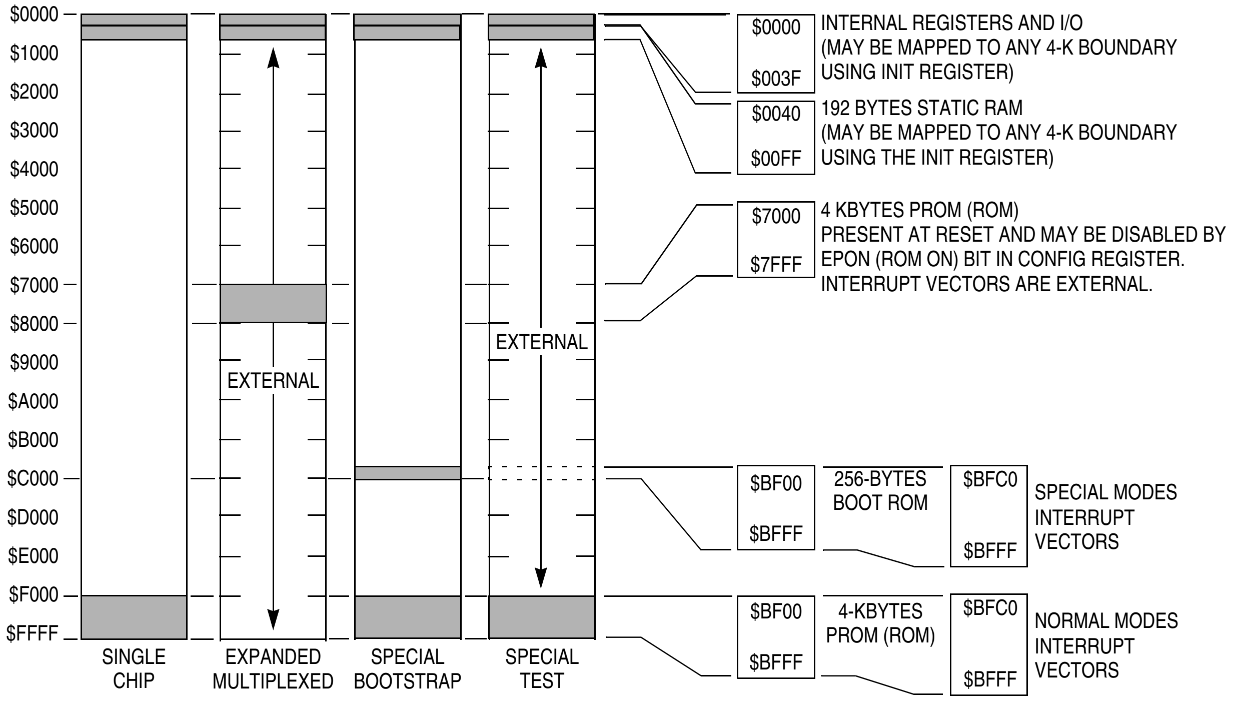
2.2.4 Special Test Mode

This special expanded mode is primarily intended or production testing. The user can access a number of special test control bits in this mode. Reset and interrupt vectors are fetched externally from locations $BFC0–$BFFF. A switch can be made from this mode to other modes under program control.

2.3 Memory Map

Figure 2-1 illustrates the memory map for both normal modes of operation (single-chip and expanded-multiplexed), as well as for both special modes of operation (bootstrap and test).

- In the single-chip mode, the MCU does not generate external addresses. The internal memory locations are shown in the shaded areas, and the contents of these shaded areas are explained on the right side of the diagram.
- In expanded-multiplexed mode, the memory locations are basically the same as in the single-chip mode except that the memory locations between shaded areas are for externally addressed memory and I/O.
- The special bootstrap mode is similar to the single-chip mode, except that the bootstrap program ROM is located at memory locations $BF00–$BFFF, vectors included.
- The special test mode is similar to the expanded-multiplexed mode except the interrupt vectors are at external memory locations.


$0000
$1000
$2000
$3000
$4000
$5000
$6000
$7000
$8000
$9000
$A000
$B000
$C000
$D000
$E000
$F000
$FFFF
EXTERNAL
EXTERNAL
SINGLE CHIP
EXPANDED MULTIPLEXED
SPECIAL BOOTSTRAP
SPECIAL TEST
$0000
$003F
INTERNAL REGISTERS AND I/O (MAY BE MAPPED TO ANY 4-K BOUNDARY USING INIT REGISTER)
$0040
$00FF
192 BYTES STATIC RAM (MAY BE MAPPED TO ANY 4-K BOUNDARY USING THE INIT REGISTER)
$7000
$7FFF
4 KBYTES PROM (ROM) PRESENT AT RESET AND MAY BE DISABLED BY EPON (ROM ON) BIT IN CONFIG REGISTER. INTERRUPT VECTORS ARE EXTERNAL.
$BF00
$BFFF
256-BYTES BOOT ROM
$BFC0
$BFFF
SPECIAL MODES INTERRUPT VECTORS
$BF00
$BFFF
4-KBYTES PROM (ROM)
$BFC0
$BFFF
NORMAL MODES INTERRUPT VECTORS

MODB	MODA	Mode Selected
1	0	Single-chip (mode 0)
1	1	Expanded multiplexed (mode 1)
0	0	Special bootstrap
0	1	Special test

Figure 2-1. MC68HC711D3 Memory Map

2.3.1 Control and Status Registers

Figure 2-2 is a representation of all 64 bytes of control and status registers, I/O and data registers, and reserved locations that make up the internal register block. This block may be mapped to any 4-K boundary in memory, but reset locates it at \$0000–\$003F. This mappability factor and the default starting addresses are indicated by the use of a bold **0** as the starting character of a register's address.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$**0**000	Port A Data Register (PORTA) See page 68.	Read: / Write:	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
		Reset:	Hi-Z	0	0	0	Hi-Z	Hi-Z	Hi-Z	Hi-Z
\$**0**001	Reserved		R	R	R	R	R	R	R	R
\$**0**002	Port C Control Register (PIOC) See page 70.	Read: / Write:	0	0	CWOM	0	0	0	0	0
		Reset:	0	0	0	0	0	0	0	0
\$**0**003	Port C Data Register (PORTC) See page 70.	Read: / Write:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
		Reset:	Reset configures pins as Hi-Z inputs							
\$**0**004	Port B Data Register (PORTB) See page 69.	Read: / Write:	PB7	PB6	PB5	PB4	PB3	BP2	BP1	PB0
		Reset:	Reset configures pins as Hi-Z inputs							
\$**0**005	Reserved		R	R	R	R	R	R	R	R
\$**0**006	Data Direction Register for Port B (DDRB) See page 69.	Read: / Write:	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
		Reset:	0	0	0	0	0	0	0	0
\$**0**007	Data Direction Register for Port C (DDRC) See page 71.	Read: / Write:	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
		Reset:	0	0	0	0	0	0	0	0
\$**0**008	Port D Data Register (PORTD) See page 71.	Read: / Write:	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
		Reset:	0	0	0	0	0	0	0	0
\$**0**009	Data Direction Register for Port D (DDRD) See page 72.	Read: / Write:	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
		Reset:	0	0	0	0	0	0	0	0
\$**0**00A	Reserved		R	R	R	R	R	R	R	R

▨ = Unimplemented R = Reserved U = Unaffected

Figure 2-2. Register and Control Bit Assignments (Sheet 1 of 5)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$000B	Timer Compare Force Register (CFORC) See page 104.	Read: Write:	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0
		Reset:	0	0	0	0	0	0	0	0
$000C	Output Compare 1 Mask Register (OC1M) See page 105.	Read: Write:	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0
		Reset:	0	0	0	0	0	0	0	0
$000D	Output Compare 1 Data Register (OC1D) See page 105.	Read: Write:	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0
		Reset:	0	0	0	0	0	0	0	0
$000E	Timer Counter Register High (TCNT) See page 106.	Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Reset:	0	0	0	0	0	0	0	0
$000F	Timer Counter Register Low (TCNT) See page 106.	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	0	0	0	0	0	0	0	0
$0010	Timer Input Capture Register 1 High (TIC1) See page 100.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	Unaffected by reset							
$0011	Timer Input Capture Register 1 Low (TIC1) See page 100.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	Unaffected by reset							
$0012	Timer Input Capture Register 2 High (TIC2) See page 100.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	Unaffected by reset							
$0013	Timer Input Capture Register 2 Low (TIC2) See page 100.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	Unaffected by reset							
$0014	Timer Input Capture Register 3 High (TIC3) See page 100.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	Unaffected by reset							
$0015	Timer Input Capture Register 3 Low (TIC3) See page 100.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	Unaffected by reset							
$0016	Timer Output Compare Register 1 High (TOC1) See page 103.	Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 15
		Reset:	1	1	1	1	1	1	1	1

▨ = Unimplemented   R = Reserved   U = Unaffected

Figure 2-2. Register and Control Bit Assignments (Sheet 2 of 5)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0017	Timer Output Compare Register 1 Low (TOC1) See page 103.	Read: / Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$0018	Timer Output Compare Register 2 High (TOC2) See page 103.	Read: / Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Reset:	1	1	1	1	1	1	1	1
$0019	Timer Output Compare Register 2 Low (TOC2) See page 103.	Read: / Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$001A	Timer Output Compare Register 3 High (TOC3) See page 103.	Read: / Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Reset:	1	1	1	1	1	1	1	1
$001B	Timer Output Compare Register 3 Low (TOC3) See page 103.	Read: / Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$001C	Timer Output Compare Register 4 High (TOC4) See page 103.	Read: / Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Reset:	1	1	1	1	1	1	1	1
$001D	Timer Output Compare Register 4 Low (TOC4) See page 103.	Read: / Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$001E	Timer Input Capture 4/ Output Compare 5 Register High (TI4/O5) See page 101.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	1	1	1	1	1	1	1	1
$001F	Timer Input Capture 4/ Output Compare 5 Register Low (TI4/O5) See page 101.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:	▨	▨	▨	▨	▨	▨	▨	▨
		Reset:	1	1	1	1	1	1	1	1
$0020	Timer Control 1 Register (TCTL1) See page 106.	Read: / Write:	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5
		Reset:	0	0	0	0	0	0	0	0
$0021	Timer Control Register 2 (TCTL2) See page 99.	Read: / Write:	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
		Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented   R = Reserved   U = Unaffected

Figure 2-2. Register and Control Bit Assignments (Sheet 3 of 5)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0022	Timer Interrupt Mask 1 Register (TMSK1) See page 107.	Read: Write:	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I
		Reset:	0	0	0	0	0	0	0	0
$0023	Timer Interrupt Flag 1 Register (TFLG1) See page 108.	Read: Write:	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F
		Reset:	0	0	0	0	0	0	0	0
$0024	Timer Interrupt Mask 2 Register (TMSK2) See page 108.	Read: Write:	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0
		Reset:	0	0	0	0	0	0	0	0
$0025	Timer Interrupt Flag 2 Register (TFLG2) See page 109.	Read: Write:	TOF	RTIF	PAOVF	PAIF	0	0	0	0
		Reset:	0	0	0	0	0	0	0	0
$0026	Pulse Accumulator Control Register (PACTL) See pages 112 and 114.	Read: Write:	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0
		Reset:	0	0	0	0	0	0	0	0
$0027	Pulse Accumulator Count Register (PACNT) See page 115.	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	Unaffected by reset							
$0028	SPI Control Register (SPCR) See page 92.	Read: Write:	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0
		Reset:	0	0	0	0	0	1	U	U
$0029	SPI Status Register (SPSR) See page 93.	Read: Write:	SPIF	WCOL	0	MODF	0	0	0	0
		Reset:	0	0	0	0	0	0	0	0
$002A	SPI Data I/O Register (SPDR) See page 94.	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	Unaffected by reset							
$002B	Baud Rate Register (BAUD) See page 81.	Read: Write:	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0
		Reset:	0	0	0	0	0	U	U	U
$002C	SCI Control Register 1 (SCCR1) See page 78.	Read: Write:	R8	T8	0	M	WAKE	0	0	0
		Reset:	U	U	0	0	0	0	0	0
$002D	SCI Control Register 2 (SCCR2) See page 79.	Read: Write:	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
		Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented R = Reserved U = Unaffected

Figure 2-2. Register and Control Bit Assignments (Sheet 4 of 5)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$002E	SCI Status Register (SCSR) See page 80.	Read:	TDRE	TC	RDRF	IDLE	OR	NF	FE	0
		Write:								
		Reset:	1	1	0	0	0	0	0	0
$002F	SCI Data Register (SCDR) See page 78.	Read:	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0
		Write:								
		Reset:	Unaffected by reset							
$0030 ↓ $0038	Reserved		R	R	R	R	R	R	R	R
$0039	System Configuration Options Register (OPTION) See page 53.	Read:	0	0	IRQE	DLY	CME	0	CR1	CR0
		Write:								
		Reset:	0	0	0	1	0	0	0	0
$003A	Arm/Reset COP Timer Circuitry Register (COPRST) See page 52.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$003B	PROM Programming Control Register (PPROG) See page 33.	Read:	MBE	0	ELAT	EXCOL	EXROW	0	0	PGM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$003C	Highest Priority I-Bit Interrupt and Miscellaneous Register (HPRIO) See page 63.	Read:	RBOOT	SMOD	MDA	IRVNE	PSEL3	PSEL2	PSEL1	PSEL0
		Write:		▨						
		Reset:	Note 1				0	1	0	1
$003D	RAM and I/O Mapping Register (INIT) See page 30.	Read:	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0
		Write:								
		Reset:	0	0	0	0	0	0	0	1
$003E	Test 1 Register (TEST)	Read:	TILOP	0	OCC4	CBYP	DISR	FCM	FCOP	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$003F	System Configuration Register (CONFIG) See page 31.	Read:	0	0	0	0	0	NOCOP	ROMON	0
		Write:								
		Reset:	0	0	0	0	0	U	U	0

1. The values of the RBOOT, SMOD, IRVNE, and MDA bits at reset depend on the mode during initialization. Refer to **Table 4-3. Hardware Mode Select Summary**.

▨ = Unimplemented R = Reserved U = Unaffected

Figure 2-2. Register and Control Bit Assignments (Sheet 5 of 5)

2.3.2 RAM and I/O Mapping Register

The random-access memory (RAM) and input/output (I/O) mapping register (INIT) is a special-purpose 8-bit register that is used during initialization to change the default locations of RAM and control registers within the MCU memory map. It can be written to only once within the first 64 E-clock cycles after a reset in normal modes. Thereafter, it becomes a read-only register.

Address: $**0**03D

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

Figure 2-3. RAM and I/O Mapping Register (INIT)

RAM2–RAM0 (INIT bits 7–4) specify the starting address for the 192 bytes of static RAM. REG3–REG0 (INIT bits 3–0) specify the starting address for the control and status register block. In each case, the four RAM or REG bits become the four upper bits of the 16-bit address of the RAM or register. Since the INIT register is set to $00 by reset, the internal registers begin at $0000 and RAM begins at $0040.

Throughout this document, control and status register addresses are displayed with the high-order digit shown as a bold **0**. This convention indicates that the register block may be relocated to any 4-K memory page, but that its default location is $0000.

RAM and the control and status registers can be relocated independently. If the control and status registers are relocated in such a way as to conflict with PROM, then the register block takes priority, and the EPROM or OTPROM at those locations becomes inaccessible. No harmful conflicts result. Lower priority resources simply become inaccessible. Similarly, if an internal resource conflicts with an external device, no harmful conflict results, since data from the external device is not applied to the internal data bus. Thus, it cannot interfere with the internal read.

NOTE: *There are unused register locations in the 64-byte control and status register block. Reads of these unused registers return data from the undriven internal data bus, not from another source that happens to be located at the same address.*

2.3.3 Configuration Control Register

The configuration control register (CONFIG) controls the presence of OTPROM or EPROM in the memory map and enables the computer operating properly (COP) watchdog system.

This register is writable only once in expanded and single-chip modes (SMOD = 0). In these mode, the COP watchdog timer is enabled out of reset. In all modes, except normal expanded, EPROM is enabled and located at $F000–$FFFF. In normal expanded mode, EPROM is enabled and located at \$7000–\$7FFF. Should the user wish to be in expanded mode, but with EPROM mapped at \$F000–\$FFFF, he must reset in single-chip mode, and write a 1 to the MDA bit in the HPRIO register.

Address: \$003F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	0	0	0	0	0	NOCOP	ROMON	0
Reset:	0	0	0	0	0	U	U	0

U = Unaffected

Figure 2-4. Configuration Control Register (CONFIG)

Bits 7–3 and 0 — Not implemented
Always read 0.

NOCOP — Computer Operating Properly System Disable Bit
This bit is cleared out of reset in normal modes (single chip and expanded), enabling the COP system. It is writable only once after reset in these modes (SMOD = 0). In the special modes (test and bootstrap) (SMOD = 1), this bit comes out of reset set, and is writable any time.
1 = COP system is disabled.
0 = COP system is enabled, reset forced on timeout.

ROMON — PROM Enable Bit
This bit is set out of reset, enabling the EPROM or OTPROM in all modes. This bit is writable once in normal modes (SMOD = 0), but is writable at any time in special modes (SMOD = 1).
1 = PROM is present in the memory map.
0 = PROM is disabled from the memory map.

NOTE: *In expanded mode out of reset, the EPROM or OTPROM is located at \$7000–\$7FFF. In all other modes, the PROM resides at \$F000–\$FFFF.*

2.4 Programmable Read-Only Memory (PROM)

The MC68HC711D3 has 4-Kbytes of one-time programmable read-only memory (OTPROM). The PROM address is \$F000–\$FFFF in all modes except expanded multiplexed. In expanded- multiplexed mode, the PROM is located at \$7000–\$7FFF after reset.

The on-chip read-only memory (ROM) of an MC68HC711D3 is programmed in MCU mode. In this mode, the PROM is programmed through the MCU in the bootstrap or test modes. The erased state of a PROM byte is \$FF.

Using the on-chip OTPROM programming feature requires an external 12-volt nominal power supply (V_{PP}). Normal programming is accomplished using the OTPROM programming register (PPROG).

As described in the following subsections, these two methods of programming and verifying EPROM are possible:

1. Programming an individual EPROM address
2. Programming the EPROM with downloaded data

2.4.1 Programming an Individual EPROM Address

In this method, the MCU programs its own EPROM by controlling the PPROG register. Use these procedures to program the EPROM through the MCU with:

- The ROMON bit set in the CONFIG register
- The 12-volt nominal programming voltage present on the $\overline{XIRQ}/V_{PP}$ pin
- The $\overline{IRQ}$ pin must be pulled high.

```
EPROG   LDAB   #$20
        STAB   $003B  Set ELAT bit (PGM = 0) to enable
                      EPROM latches.
        STAA   $0,X   Store data to EPROM address
        LDAB   #$21
        STAB   $003B  Set PGM bit with ELAT = 1 to enable
                      EPROM programming voltage
        JSR    DLYEP  Delay 2–4 ms
        CLR    $003B  Turn off programming voltage and set
                      to READ mode
```

2.4.2 Programming the 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 $\overline{XIRQ}/V_{PP}$ pin. To use the resident utility, bootload a 3-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 = \$F000).

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.

2.4.3 PROM Programming Control Register

The PROM programming control register (PPROG) is used to control the programming of the OTPROM or EPROM. PPROG is cleared on reset so that the PROM is configured for normal read.

Address: $003B

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	MBE	0	ELAT	EXCOL	EXROW	0	0	PGM
Reset:	0	0	0	0	0	0	0	0

Figure 2-5. PROM Programming Control Register (PPROG)

MBE — Multiple Byte Program Enable Bit
This bit is reserved for testing.

Bit 6, 2, and 1 — Not implemented
Always read 0.

ELAT — EPROM (OTPROM) Latch Control Bit
1 = PROM address and data bus are configured for programming. Writes to PROM cause address and data to be latched. The PROM cannot be read.
0 = PROM address and data bus are configured for normal reads. PROM cannot be programmed.

EXCOL — Select Extra Columns Bit
This bit is reserved for testing.

EXROW — Select Extra Row Bit
This bit is reserved for testing.

PGM — EPROM (OTPROM) Program Command Bit
This bit may be written only when ELAT = 1.
1 = Programming power is switched on to PROM array.
0 = Programming power is switched off.

Section 3. Central Processor Unit (CPU)

3.1 Introduction

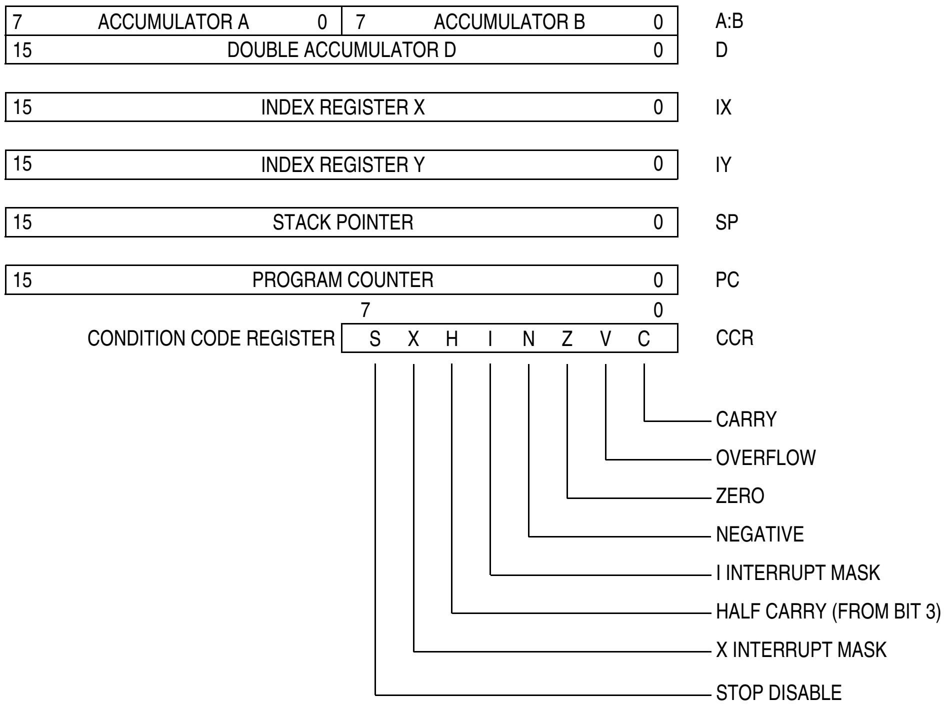
This section presents information on M68HC11 central processor unit (CPU):

- Architecture
- Data types
- Addressing modes
- Instruction set
- Special operations such as subroutine calls and interrupts

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

3.2 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**.



7 ACCUMULATOR A 0 | 7 ACCUMULATOR B 0 — A:B
15 DOUBLE ACCUMULATOR D 0 — D
15 INDEX REGISTER X 0 — IX
15 INDEX REGISTER Y 0 — IY
15 STACK POINTER 0 — SP
15 PROGRAM COUNTER 0 — PC
CONDITION CODE REGISTER 7 S X H I N Z V C 0 — CCR
CARRY
OVERFLOW
ZERO
NEGATIVE
I INTERRUPT MASK
HALF CARRY (FROM BIT 3)
X INTERRUPT MASK
STOP DISABLE

Figure 3-1. Programming Model

3.2.1 Accumulators A, B, and D

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

- The ABX and ABY instructions add the contents of 8-bit accumulator B to the contents of 16-bit register X or Y, but there are no equivalent instructions that use A instead of B.
- The TAP and TPA instructions transfer data from accumulator A to the condition code register or from the condition code register to accumulator A. However, there are no equivalent instructions that use B rather than A.
- The decimal adjust accumulator A (DAA) instruction is used after binary-coded decimal (BCD) arithmetic operations, but there is no equivalent BCD instruction to adjust accumulator B.
- The add, subtract, and compare instructions associated with both A and B (ABA, SBA, and CBA) only operate in one direction, making it important to plan ahead to ensure that the correct operand is in the correct accumulator.

3.2.2 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.

3.2.3 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 **3.4 Opcodes and Operands** for further information.

3.2.4 Stack Pointer (SP)

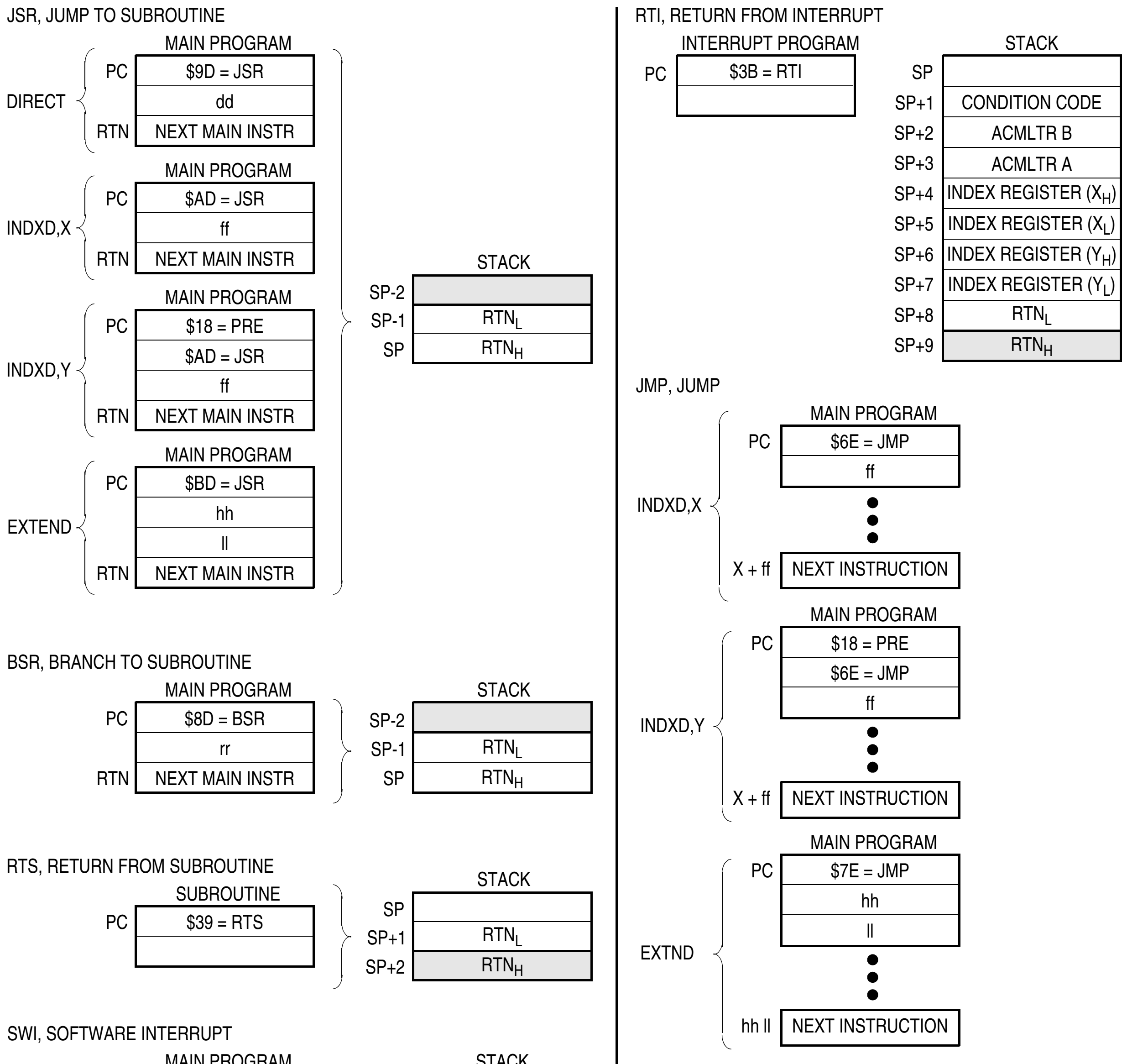
The M68HC11 CPU has an automatic program stack. This stack can be located anywhere in the address space and can be any size up to the amount of memory available in the system. Normally, the SP is initialized by one of the first instructions in an application program. The stack is configured as a data 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.

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

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

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

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



LEGEND:

RTN	Address of next instruction in main program to be executed upon return from subroutine
RTN$_H$	Most significant byte of return address
RTN$_L$	Least significant byte of return address
▨	Shaded cells show 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

3.2.5 Program Counter (PC)

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

Table 3-1. Reset Vector Comparison

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

3.2.6 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 $\overline{\text{XIRQ}}$)
- One stop disable bit (S)

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

3.2.6.1 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.

3.2.6.2 Overflow (V)

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

3.2.6.3 Zero (Z)

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

3.2.6.4 Negative (N)

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

3.2.6.5 I-Interrupt Mask (I)

The interrupt request (IRQ) mask (I bit) is a global mask that disables all maskable interrupt sources. While the I bit is set, interrupts can become pending, but the operation of the CPU continues uninterrupted until the I bit is cleared. After any reset, the I bit is set by default and can be cleared only by a software instruction. When an interrupt is recognized, the I bit is set after the registers are stacked, but before the interrupt vector is fetched. After the interrupt has been serviced, a return-from-interrupt instruction is normally executed, restoring the registers to the values that were present before the interrupt occurred. Normally, the I bit is 0 after a return from interrupt is executed. Although the I bit can be cleared within an interrupt service routine, "nesting" interrupts in this way should be done only when there is a clear understanding of latency and of the arbitration mechanism. Refer to **Section 4. Resets, Interrupts, and Low-Power Modes**.

3.2.6.6 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.

3.2.6.7 X-Interrupt Mask (X)

The XIRQ mask (X) bit disables interrupts from the $\overline{\text{XIRQ}}$ pin. After any reset, X is set by default and must be cleared by a software instruction. When an $\overline{\text{XIRQ}}$ interrupt is recognized, the X and I bits are set after the registers are stacked, but before the interrupt vector is fetched. After the interrupt has been serviced, an RTI instruction is normally executed, causing the registers to be restored to the values that were present before the interrupt occurred. The X interrupt mask bit is set only by hardware ($\overline{\text{RESET}}$ or $\overline{\text{XIRQ}}$ acknowledge). X is cleared only by program instruction (TAP, where the associated bit of A is 0; or RTI, where bit 6 of the value loaded into the CCR from the stack has been cleared). There is no hardware action for clearing X.

3.2.6.8 STOP Disable (S)

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

3.3 Data Types

The M68HC11 CPU supports four data types:

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

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

3.4 Opcodes and Operands

The M68HC11 Family of microcontrollers uses 8-bit opcodes. Each opcode identifies a particular instruction and associated addressing mode to the CPU. Several opcodes are required to provide each instruction with a range of addressing capabilities. Only 256 opcodes would be available if the range of values were restricted to the number able to be expressed in 8-bit binary numbers.

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

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

3.5 Addressing Modes

Six addressing modes can be used to access memory:

1. Immediate
2. Direct
3. Extended
4. Indexed
5. Inherent
6. 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.

3.5.1 Immediate

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

3.5.2 Direct

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

3.5.3 Extended

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

3.5.4 Indexed

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

3.5.5 Inherent

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

3.5.6 Relative

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

3.6 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 8)

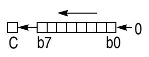
Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
ABA	Add Accumulators	A + B ⇒ A	INH	1B	—	2	—	—	Δ	—	Δ	Δ	Δ	Δ
ABX	Add B to X	IX + (00 : B) ⇒ IX	INH	3A	—	3	—	—	—	—	—	—	—	—
ABY	Add B to Y	IY + (00 : B) ⇒ IY	INH	18 3A	—	4	—	—	—	—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

Table 3-2. Instruction Set (Sheet 2 of 8)

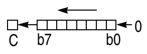
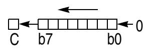
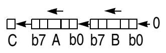
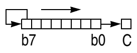
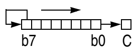
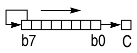
Mnemonic	Operation	Description	Addressing Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	S	X	H	I	N	Z	V	C
ASLA	Arithmetic Shift Left A		A INH	48	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLB	Arithmetic Shift Left B		B INH	58	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLD	Arithmetic Shift Left D		INH	05	—	3	—	—	—	—	Δ	Δ	Δ	Δ
ASR	Arithmetic Shift Right		EXT IND,X IND,Y	77 67 18 67	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
ASRA	Arithmetic Shift Right A		A INH	47	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASRB	Arithmetic Shift Right B		B INH	57	—	2	—	—	—	—	Δ	Δ	Δ	Δ
BCC (rel)	Branch if Carry Clear	? C = 0	REL	24	rr	3	—	—	—	—	—	—	—	—
BCLR (opr) (msk)	Clear Bit(s)	$M \bullet (\overline{mm}) \Rightarrow M$	DIR IND,X IND,Y	15 1D 18 1D	dd mm ff mm ff mm	6 7 8	—	—	—	—	Δ	Δ	0	—
BCS (rel)	Branch if Carry Set	? C = 1	REL	25	rr	3	—	—	—	—	—	—	—	—
BEQ (rel)	Branch if = Zero	? Z = 1	REL	27	rr	3	—	—	—	—	—	—	—	—
BGE (rel)	Branch if Δ Zero	? N ⊕ V = 0	REL	2C	rr	3	—	—	—	—	—	—	—	—
BGT (rel)	Branch if > Zero	? Z + (N ⊕ V) = 0	REL	2E	rr	3	—	—	—	—	—	—	—	—
BHI (rel)	Branch if Higher	? C + Z = 0	REL	22	rr	3	—	—	—	—	—	—	—	—
BHS (rel)	Branch if Higher or Same	? C = 0	REL	24	rr	3	—	—	—	—	—	—	—	—
BITA (opr)	Bit(s) Test A with Memory	A • M	A IMM A DIR A EXT A IND,X A IND,Y	85 95 B5 A5 18 A5	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
BITB (opr)	Bit(s) Test B with Memory	B • M	B IMM B DIR B EXT B IND,X B IND,Y	C5 D5 F5 E5 18 E5	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
BLE (rel)	Branch if Δ Zero	? Z + (N ⊕ V) = 1	REL	2F	rr	3	—	—	—	—	—	—	—	—
BLO (rel)	Branch if Lower	? C = 1	REL	25	rr	3	—	—	—	—	—	—	—	—
BLS (rel)	Branch if Lower or Same	? C + Z = 1	REL	23	rr	3	—	—	—	—	—	—	—	—
BLT (rel)	Branch if < Zero	? N ⊕ V = 1	REL	2D	rr	3	—	—	—	—	—	—	—	—
BMI (rel)	Branch if Minus	? N = 1	REL	2B	rr	3	—	—	—	—	—	—	—	—
BNE (rel)	Branch if not = Zero	? Z = 0	REL	26	rr	3	—	—	—	—	—	—	—	—
BPL (rel)	Branch if Plus	? N = 0	REL	2A	rr	3	—	—	—	—	—	—	—	—
BRA (rel)	Branch Always	? 1 = 1	REL	20	rr	3	—	—	—	—	—	—	—	—
BRCLR(opr) (msk) (rel)	Branch if Bit(s) Clear	? M • mm = 0	DIR IND,X IND,Y	13 1F 18 1F	dd mm rr ff mm rr ff mm rr	6 7 8	—	—	—	—	—	—	—	—
BRN (rel)	Branch Never	? 1 = 0	REL	21	rr	3	—	—	—	—	—	—	—	—

Table 3-2. Instruction Set (Sheet 3 of 8)

Mnemonic	Operation	Description	Addressing Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	S	X	H	I	N	Z	V	C
BRSET(opr) (msk) (rel)	Branch if Bit(s) Set	? $(\overline{M})$ • mm = 0	DIR IND,X IND,Y	12 1E 18 1E	dd mm rr ff mm rr ff mm rr	6 7 8	—	—	—	—	—	—	—	—
BSET (opr) (msk)	Set Bit(s)	M + mm ⇒ M	DIR IND,X IND,Y	14 1C 18 1C	dd mm ff mm ff mm	6 7 8	—	—	—	—	Δ	Δ	0	—
BSR (rel)	Branch to Subroutine	See **Figure 3-2**	REL	8D	rr	6	—	—	—	—	—	—	—	—
BVC (rel)	Branch if Overflow Clear	? V = 0	REL	28	rr	3	—	—	—	—	—	—	—	—
BVS (rel)	Branch if Overflow Set	? V = 1	REL	29	rr	3	—	—	—	—	—	—	—	—
CBA	Compare A to B	A – B	INH	11	—	2	—	—	—	—	Δ	Δ	Δ	Δ
CLC	Clear Carry Bit	0 ⇒ C	INH	0C	—	2	—	—	—	—	—	—	—	0
CLI	Clear Interrupt Mask	0 ⇒ I	INH	0E	—	2	—	—	—	0	—	—	—	—
CLR (opr)	Clear Memory Byte	0 ⇒ M	EXT IND,X IND,Y	7F 6F 18 6F	hh ll ff ff	6 6 7	—	—	—	—	0	1	0	0
CLRA	Clear Accumulator A	0 ⇒ A	A INH	4F	—	2	—	—	—	—	0	1	0	0
CLRB	Clear Accumulator B	0 ⇒ B	B INH	5F	—	2	—	—	—	—	0	1	0	0
CLV	Clear Overflow Flag	0 ⇒ V	INH	0A	—	2	—	—	—	—	—	—	0	—
CMPA (opr)	Compare A to Memory	A – M	A IMM A DIR A EXT A IND,X A IND,Y	81 91 B1 A1 18 A1	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
CMPB (opr)	Compare B to Memory	B – M	B IMM B DIR B EXT B IND,X B IND,Y	C1 D1 F1 E1 18 E1	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
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

Table 3-2. Instruction Set (Sheet 4 of 8)

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
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	—	—
INY	Increment Index Register Y	IY + 1 ⇒ IY	INH	18 08	—	4	—	—	—	—	—	Δ	—	—
JMP (opr)	Jump	See **Figure 3-2**	EXT IND,X IND,Y	7E 6E 18 6E	hh ll ff ff	3 3 4	—	—	—	—	—	—	—	—
JSR (opr)	Jump to Subroutine	See **Figure 3-2**	DIR EXT IND,X IND,Y	9D BD AD 18 AD	dd hh ll ff ff	5 6 6 7	—	—	—	—	—	—	—	—
LDAA (opr)	Load Accumulator A	M ⇒ A	A IMM A DIR A EXT A IND,X A IND,Y	86 96 B6 A6 18 A6	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—

Table 3-2. Instruction Set (Sheet 5 of 8)

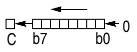
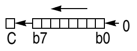
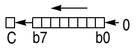
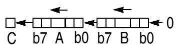
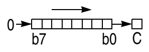
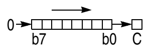
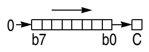
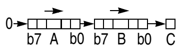
Mnemonic	Operation	Description	Addressing Mode	Instruction Opcode	Instruction Operand	Instruction Cycles	S	X	H	I	N	Z	V	C
LDAB (opr)	Load Accumulator B	M ⇒ B	B IMM B DIR B EXT B IND,X B IND,Y	C6 D6 F6 E6 18 E6	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
LDD (opr)	Load Double Accumulator D	M ⇒ A,M + 1 ⇒ B	IMM DIR EXT IND,X IND,Y	CC DC FC EC 18 EC	jj kk dd hh ll ff ff	3 4 5 5 6	—	—	—	—	Δ	Δ	0	—
LDS (opr)	Load Stack Pointer	M : M + 1 ⇒ SP	IMM DIR EXT IND,X IND,Y	8E 9E BE AE 18 AE	jj kk dd hh ll ff ff	3 4 5 5 6	—	—	—	—	Δ	Δ	0	—
LDX (opr)	Load Index Register X	M : M + 1 ⇒ IX	IMM DIR EXT IND,X IND,Y	CE DE FE EE CD EE	jj kk dd hh ll ff ff	3 4 5 5 6	—	—	—	—	Δ	Δ	0	—
LDY (opr)	Load Index Register Y	M : M + 1 ⇒ IY	IMM DIR EXT IND,X IND,Y	18 CE 18 DE 18 FE 1A EE 18 EE	jj kk dd hh ll ff ff	4 5 6 6 6	—	—	—	—	Δ	Δ	0	—
LSL (opr)	Logical Shift Left		EXT IND,X IND,Y	78 68 18 68	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
LSLA	Logical Shift Left A		A INH	48	—	2	—	—	—	—	Δ	Δ	Δ	Δ
LSLB	Logical Shift Left B		B INH	58	—	2	—	—	—	—	Δ	Δ	Δ	Δ
LSLD	Logical Shift Left Double		INH	05	—	3	—	—	—	—	Δ	Δ	Δ	Δ
LSR (opr)	Logical Shift Right		EXT IND,X IND,Y	74 64 18 64	hh ll ff ff	6 6 7	—	—	—	—	0	Δ	Δ	Δ
LSRA	Logical Shift Right A		A INH	44	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRB	Logical Shift Right B		B INH	54	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRD	Logical Shift Right Double		INH	04	—	3	—	—	—	—	0	Δ	Δ	Δ
MUL	Multiply 8 by 8	A ∗ B ⇒ D	INH	3D	—	10	—	—	—	—	—	—	—	Δ
NEG (opr)	Two's Complement Memory Byte	0 – M ⇒ M	EXT IND,X IND,Y	70 60 18 60	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
NEGA	Two's Complement A	0 – A ⇒ A	A INH	40	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NEGB	Two's Complement B	0 – B ⇒ B	B INH	50	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NOP	No operation	No Operation	INH	01	—	2	—	—	—	—	—	—	—	—

Table 3-2. Instruction Set (Sheet 6 of 8)

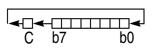
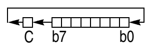
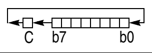
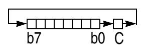
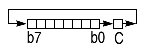
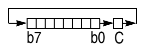
Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
ORAA (opr)	OR Accumulator A (Inclusive)	A + M ⇒ A	A IMM	8A	ii	2	—	—	—	—	Δ	Δ	0	—
			A DIR	9A	dd	3								
			A EXT	BA	hh ll	4								
			A IND,X	AA	ff	4								
			A IND,Y	18 AA	ff	5								
ORAB (opr)	OR Accumulator B (Inclusive)	B + M ⇒ B	B IMM	CA	ii	2	—	—	—	—	Δ	Δ	0	—
			B DIR	DA	dd	3								
			B EXT	FA	hh ll	4								
			B IND,X	EA	ff	4								
			B IND,Y	18 EA	ff	5								
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	79	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND,X	69	ff	6								
			IND,Y	18 69	ff	7								
ROLA	Rotate Left A		A INH	49	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROLB	Rotate Left B		B INH	59	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROR (opr)	Rotate Right		EXT	76	hh ll	6	—	—	—	—	Δ	Δ	Δ	Δ
			IND,X	66	ff	6								
			IND,Y	18 66	ff	7								
RORA	Rotate Right A		A INH	46	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RORB	Rotate Right B		B INH	56	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RTI	Return from Interrupt	See Figure 3-2	INH	3B	—	12	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ
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	82	ii	2	—	—	—	—	Δ	Δ	Δ	Δ
			A DIR	92	dd	3								
			A EXT	B2	hh ll	4								
			A IND,X	A2	ff	4								
			A IND,Y	18 A2	ff	5								

Table 3-2. Instruction Set (Sheet 7 of 8)

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
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	—
STAB (opr)	Store Accumulator B	B ⇒ M	B DIR B EXT B IND,X B IND,Y	D7 F7 E7 18 E7	dd hh ll ff ff	3 4 4 5	—	—	—	—	Δ	Δ	0	—
STD (opr)	Store Accumulator D	A ⇒ M, B ⇒ M + 1	DIR EXT IND,X IND,Y	DD FD ED 18 ED	dd hh ll ff ff	4 5 5 6	—	—	—	—	Δ	Δ	0	—
STOP	Stop Internal Clocks	—	INH	CF	—	2	—	—	—	—	—	—	—	—
STS (opr)	Store Stack Pointer	SP ⇒ M : M + 1	DIR EXT IND,X IND,Y	9F BF AF 18 AF	dd hh ll ff ff	4 5 5 6	—	—	—	—	Δ	Δ	0	—
STX (opr)	Store Index Register X	IX ⇒ M : M + 1	DIR EXT IND,X IND,Y	DF FF EF CD EF	dd hh ll ff ff	4 5 5 6	—	—	—	—	Δ	Δ	0	—
STY (opr)	Store Index Register Y	IY ⇒ M : M + 1	DIR EXT IND,X IND,Y	18 DF 18 FF 1A EF 18 EF	dd hh ll ff ff	5 6 6 6	—	—	—	—	Δ	Δ	0	—
SUBA (opr)	Subtract Memory from A	A – M ⇒ A	A IMM A DIR A EXT A IND,X A IND,Y	80 90 B0 A0 18 A0	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
SUBB (opr)	Subtract Memory from B	B – M ⇒ B	A IMM A DIR A EXT A IND,X A IND,Y	C0 D0 F0 E0 18 E0	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
SUBD (opr)	Subtract Memory from D	D – M : M + 1 ⇒ D	IMM DIR EXT IND,X IND,Y	83 93 B3 A3 18 A3	jj kk dd hh ll ff ff	4 5 6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
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	—	—	—	—	—	—	—	—

Table 3-2. Instruction Set (Sheet 8 of 8)

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
TST (opr)	Test for Zero or Minus	M – 0	EXT IND,X IND,Y	7D 6D 18 6D	hh ll ff ff	6 6 7	—	—	—	—	∆	∆	0	0
TSTA	Test A for Zero or Minus	A – 0	A INH	4D	—	2	—	—	—	—	∆	∆	0	0
TSTB	Test B for Zero or Minus	B – 0	B INH	5D	—	2	—	—	—	—	∆	∆	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
∆	Bit cleared or set, depending on operation
↓	Bit can be cleared, cannot become set

Section 4. Resets, Interrupts, and Low-Power Modes

4.1 Introduction

This section describes the internal and external resets and interrupts of the MC68HC711D3 and its two low power-consumption modes.

4.2 Resets

The microcontroller unit (MCU) can be reset in any of these four ways:

1. An active-low input to the $\overline{\text{RESET}}$ pin
2. A power-on reset (POR) function
3. A clock monitor failure
4. A computer operating properly (COP) watchdog timer timeout

The $\overline{\text{RESET}}$ input consists mainly of a Schmitt trigger that senses the $\overline{\text{RESET}}$ line logic level.

4.2.1 $\overline{\text{RESET}}$ Pin

To request an external reset, the $\overline{\text{RESET}}$ pin must be held low for at least eight E-clock cycles, or for one E-clock cycle if no distinction is needed between internal and external resets.

4.2.2 Power-On Reset (POR)

Power-on reset occurs when a positive transition is detected on V_{DD}. This reset is used strictly for power turn on conditions and should not be used to detect any drop in the power supply voltage. If the external $\overline{\text{RESET}}$ pin is low at the end of the power-on delay time, the processor remains in the reset condition until $\overline{\text{RESET}}$ goes high.

4.2.3 Computer Operating Properly (COP) Reset

The MCU contains a watchdog timer that automatically times out unless it is serviced within a specific time by a program reset sequence. If the COP watchdog timer is allowed to timeout, a reset is generated, which drives the $\overline{\text{RESET}}$ pin low to reset the MCU and the external system.

In the MC68HC711D3, the COP reset function is enabled out of reset in normal modes. If the user does not want the COP enabled, he must write a 1 to the NOCOP bit of the configuration control register (CONFIG) after reset. This bit is writable only once after reset in normal modes (see **2.3.3 Configuration Control Register** for more information). Protected control bits (CR1 and CR0) in the configuration options register (OPTION) allow the user to select one of the four COP timeout rates. **Table 4-1** shows the relationship between CR1 and CR0 and the COP timeout period for various system clock frequencies.

The sequence for resetting the watchdog timer is:

1. Write \$55 to the COP reset register (COPRST) to arm the COP timer clearing mechanism.
2. Write \$AA to the COPRST register to clear the COP timer

Both writes must occur in this sequence prior to the timeout, but any number of instructions can be executed between the two writes.

Table 4-1. COP Time Out Periods

CR0	CR1	E ÷ 2^{15} Divided By	XTAL = 2^{23} Time Out –0/+15.6 ms	XTAL = 8.0 MHz Time Out –0/+16.4 ms	XTAL = 4.9152 MHz Time Out –0/+26.7 ms	XTAL = 4.0 MHz Time Out –0/+32.8 ms	XTAL = 3.6864 MHz Time Out –0/+35.6 ms
0	0	1	15.625 ms	16.384 ms	26.667 ms	32.768 ms	35.556 ms
0	1	4	62.5 ms	65.536 ms	106.67 ms	131.07 ms	142.22 ms
1	0	16	250 ms	262.14 ms	426.67 ms	524.29 ms	568.89 ms
1	1	64	1 sec	1.049 sec	1.707 sec	2.1 sec	2.276 ms
		E =	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	921.6 kHz

Address: \$003A

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

Figure 4-1. Arm/Reset COP Timer Circuitry Register (COPRST)

4.2.4 Clock Monitor Reset

The MCU contains a clock monitor circuit that measures the E-clock frequency. If the E-clock input rate is above approximately 200 kHz, then the clock monitor does not generate an MCU reset. If the E-clock signal is lost or its frequency falls below 10 kHz, then an MCU reset can be generated, and the $\overline{\text{RESET}}$ pin is driven low to reset the external system.

4.2.5 System Configuration Options Register

The system configuration options register (OPTION) is a special-purpose register with several time-protected bits. OPTION is used during initialization to configure internal system options.

Bits 5, 4, 2, 1, and 0 can be written only once during the first 64 E-clock cycles after reset in normal modes (where the HPRIO register bit (SMOD) is cleared). In special modes (where SMOD = 1), the bits can be written at any time. Bit 3 can be written at anytime.

Address: $0039

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

Figure 4-2. System Configuration Options Register (OPTION)

Bits 7, 6, and 2 — Not implemented
Always read 0.

IRQE — $\overline{\text{IRQ}}$ Edge/Level Sensitivity Select
This bit can be written only once during the first 64 E-clock cycles after reset in normal modes.
1 = $\overline{\text{IRQ}}$ is configured to respond only to falling edges.
0 = $\overline{\text{IRQ}}$ is configured for low-level wired-OR operation.

DLY — Stop Mode Exit Turnon Delay
This bit is set during reset and can be written only once during the first 64 E-clock cycles after reset in normal modes. If an external clock source rather than a crystal is used, the stabilization delay can be inhibited because the clock source is assumed to be stable.
1 = A stabilization delay of 4064 E-clock cycles is imposed before processing resumes after a stop mode wakeup.
0 = No stabilization delay is imposed after story recovery.

CME — Clock Monitor Enable
1 = Clock monitor circuit is enabled.
0 = Clock monitor circuit is disabled.

CR1 and CR0 — COP Timer Rate Selects

The COP system is driven by a constant frequency of E ÷ 2^{15}. These two bits specify an additional divide-by value to arrive at the COP timeout rate. These bits are cleared during reset and can be written only once during the first 64 E-clock cycles after reset in normal modes. The value of these bits is:

CR1	CR0	E ÷ 2^{15} Divided By
0	0	1
0	1	4
1	0	16
1	1	64

4.3 Interrupts

Excluding reset-type interrupts, there are 17 hardware interrupts and one software interrupt that can be generated from all the possible sources. These interrupts can be divided into two categories: maskable and non-maskable. Fifteen of the interrupts can be masked using the I bit of the condition code register (CCR). All the on-chip (hardware) interrupts are individually maskable by local control bits. The software interrupt is non-maskable. The external input to the $\overline{\text{XIRQ}}$ pin is considered a non-maskable interrupt because it cannot be masked by software once it is enabled. However, it is masked during reset and upon receipt of an interrupt at the $\overline{\text{XIRQ}}$ pin. Illegal opcode is also a non-maskable interrupt.

Table 4-2 provides a list of the interrupts with a vector location in memory for each, as well as the actual condition code and control bits that mask each interrupt. **Figure 4-3** shows the interrupt stacking order.

Table 4-2. Interrupt and Reset Vector Assignments

Vector Address	Interrupt Source	CCR Mask	Local Mask
$FFC0, $FFC1 ↓ $FFD4, $FFD5	Reserved	—	—
$FFD6, $FFD7	SCI serial system: • SCI transmit complete • SCI transmit data register empty • SCI idle line detect • SCI receiver overrun • SCI receive data register full	I bit	 TCIE TIE ILIE RIE RIE
$FFD8, $FFD9	SPI serial transfer complete	I bit	SPIE
$FFDA, $FFDB	Pulse accumulator input edge	I bit	PAII
$FFDC, $FFDD	Pulse accumulator overflow	I bit	PAOVI

Table 4-2. Interrupt and Reset Vector Assignments (Continued)

Vector Address	Interrupt Source	CCR Mask	Local Mask
$FFDE, $FFDF	Timer overflow	I bit	TOI
$FFE0, $FFE1	Timer input capture 4/output compare 5	I bit	I4/O5I
$FFE2, $FFE3	Timer output compare 4	I bit	OC4I
$FFE4, $FFE5	Timer output compare 3	I bit	OC3I
$FFE6, $FFE7	Timer output compare 2	I bit	OC2I
$FFE8, $FFE9	Timer output compare 1	I bit	OC1I
$FFEA, $FFEB	Timer input capture 3	I bit	IC3I
$FFEC, $FFED	Timer input capture 2	I bit	IC2I
$FFEE, $FFEF	Timer input capture 1	I bit	IC1I
$FFF0, $FFF1	Real time interrupt	I bit	RTII
$FFF2, $FFF3	$\overline{\text{IRQ}}$ (external pin)	I bit	None
$FFF4, $FFF5	$\overline{\text{XIRQ}}$ pin (pseudo non-maskable)	X bit	None
$FFF6, $FFF7	Software interrupt	None	None
$FFF8, $FFF9	Illegal opcode trap	None	None
$FFFA, $FFFB	COP failure (reset)	None	NOCOP
$FFFC, $FFFD	Clock monitor fail (reset)	None	CME
$FFFE, $FFFF	$\overline{\text{RESET}}$	None	None

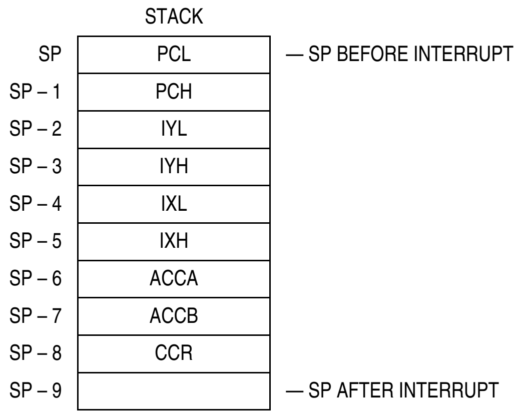


Figure 4-3. Interrupt Stacking Order

4.3.1 Software Interrupt (SWI)

The SWI is executed the same as any other instruction and takes precedence over interrupts only if the other interrupts are masked (with I and X bits in the CCR set). SWI execution is similar to that of the maskable interrupts in that it sets the I bit, stacks the central processor unit (CPU) registers, etc.

NOTE: *The SWI instruction cannot be executed as long as another interrupt is pending. However, once the SWI instruction has begun, no other interrupt can be honored until the first instruction in the SWI service routine is completed.*

4.3.2 Illegal Opcode Trap

Since not all possible opcodes or opcode sequences are defined, an illegal opcode detection circuit has been included in the MCU. When an illegal opcode is detected, an interrupt is required to the illegal opcode vector. The illegal opcode vector should never be left uninitialized.

4.3.3 Real-Time Interrupt (RTI)

The real-time interrupt (RTI) provides a programmable periodic interrupt. This interrupt is maskable by either the I bit in the CCR or the RTI enable (RTII) bit of the timer interrupt mask register 2 (TMSK2). The rate is based on the MCU E clock and is software selectable to the $E \div 2^{13}$, $E \div 2^{14}$, $E \div 2^{15}$, or $E \div 2^{16}$. See PACTL, TMSK2, and TFLG2 register descriptions in **Section 8. Programmable Timer** for control and status bit information.

4.3.4 Interrupt Mask Bits in the CCR

Upon reset, both the X bit and I bit of the CCR are set to inhibit all maskable interrupts and XIRQ. After minimum system initialization, software may clear the X bit by a TAP instruction, thus enabling XIRQ interrupts. Thereafter software cannot set the X bit. So, an XIRQ interrupt is effectively a non-maskable interrupt. Since the operation of the I bit related interrupt structure has no effect on the X bit, the internal $\overline{\text{XIRQ}}$ pin remains effectively non-masked. In the interrupt priority logic, the XIRQ interrupt is 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 the I bit are automatically set by hardware after stacking the CCR. A return-from-interrupt (RTI) instruction restores the X and I bits to their preinterrupt request state.

4.3.5 Priority Structure

Interrupts obey a fixed hardware priority circuit to resolve simultaneous requests. However one I bit related interrupt source may be elevated to the highest I bit priority in the resolution circuit.

Six interrupt sources are not masked by the I bit in the CCR and have these fixed priority relationships:

1. Reset
2. Clock monitor failure
3. COP failure
4. Illegal opcode
5. SWI
6. $\overline{\text{XIRQ}}$

SWI is actually an instruction and has highest priority, other than resets, in that once the SWI opcode is fetched, no other interrupt can be honored until the SWI vector has been fetched.

Each of the previous sources is an input to the priority resolution circuit. The highest I bit masked priority input to the resolution circuit is assigned to be connected to any one of the remaining I bit related interrupt sources. This assignment is made under the software control of the HPRIO register. To avoid timing races, the HPRIO register can be written only while the I bit related interrupts are inhibited (I bit of CCR is logic 1). An interrupt that is assigned to this higher priority position is still subject to masking by any associated control bits or by the I bit in the CCR. The interrupt vector address is not affected by assigning a source to the higher priority position.

Figure 4-4, **Figure 4-5**, and **Figure 4-6** illustrate the interrupt process as it relates to normal processing. **Figure 4-4** shows how the CPU begins from a reset, and how interrupt detection relates to normal opcode fetches. **Figure 4-5** is an expansion of a block in **Figure 4-4** and shows how interrupt priority is resolved. **Figure 4-6** is an expansion of the SCI interrupt block of **Figure 4-4** and shows the resolution of interrupt sources within the SCI subsystem.

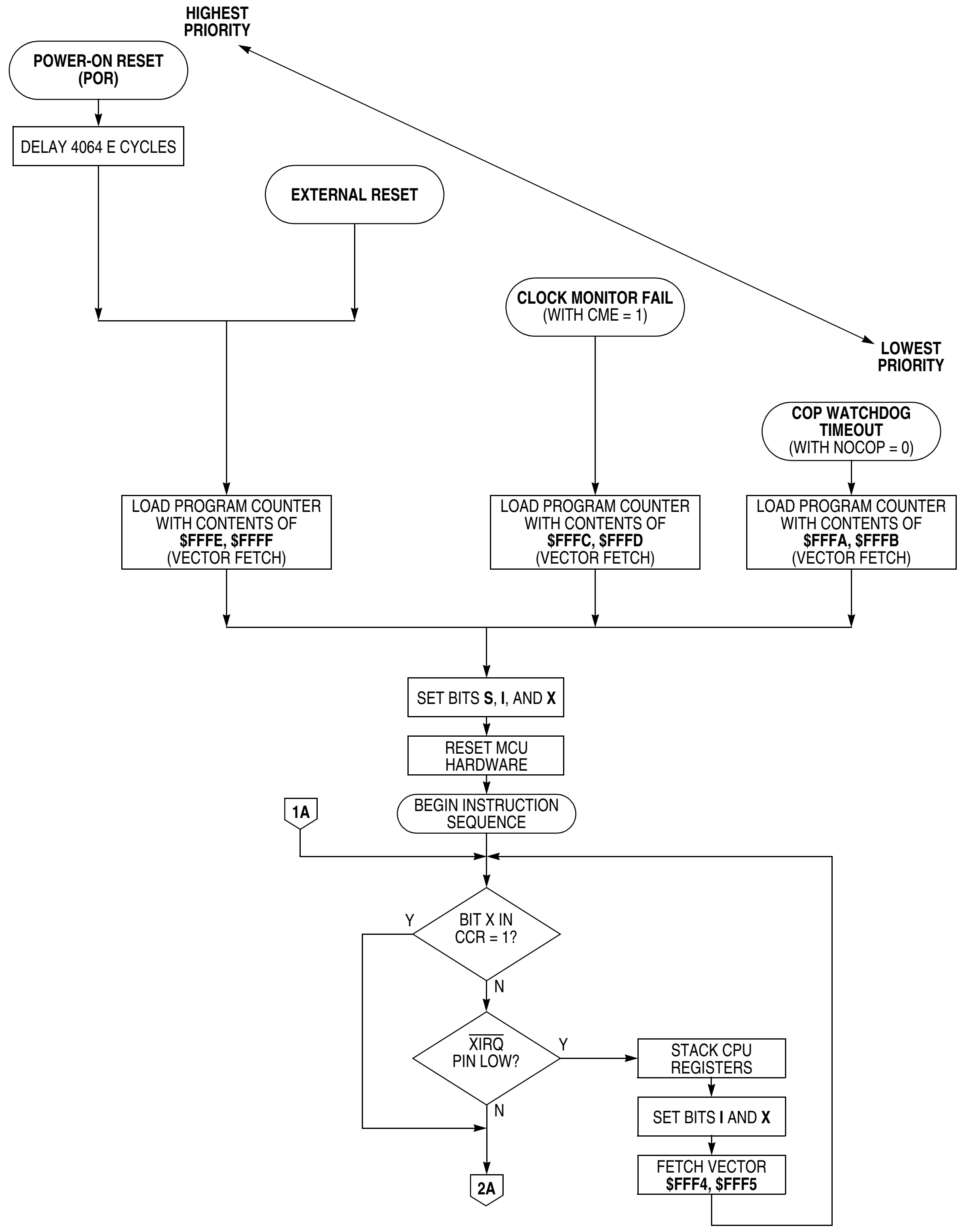

HIGHEST PRIORITY
POWER-ON RESET (POR)
DELAY 4064 E CYCLES
EXTERNAL RESET
CLOCK MONITOR FAIL (WITH CME = 1)
LOWEST PRIORITY
COP WATCHDOG TIMEOUT (WITH NOCOP = 0)
LOAD PROGRAM COUNTER WITH CONTENTS OF $FFFE, $FFFF (VECTOR FETCH)
LOAD PROGRAM COUNTER WITH CONTENTS OF $FFFC, $FFFD (VECTOR FETCH)
LOAD PROGRAM 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

Figure 4-4. Processing Flow Out of Reset (Sheet 1 of 2)

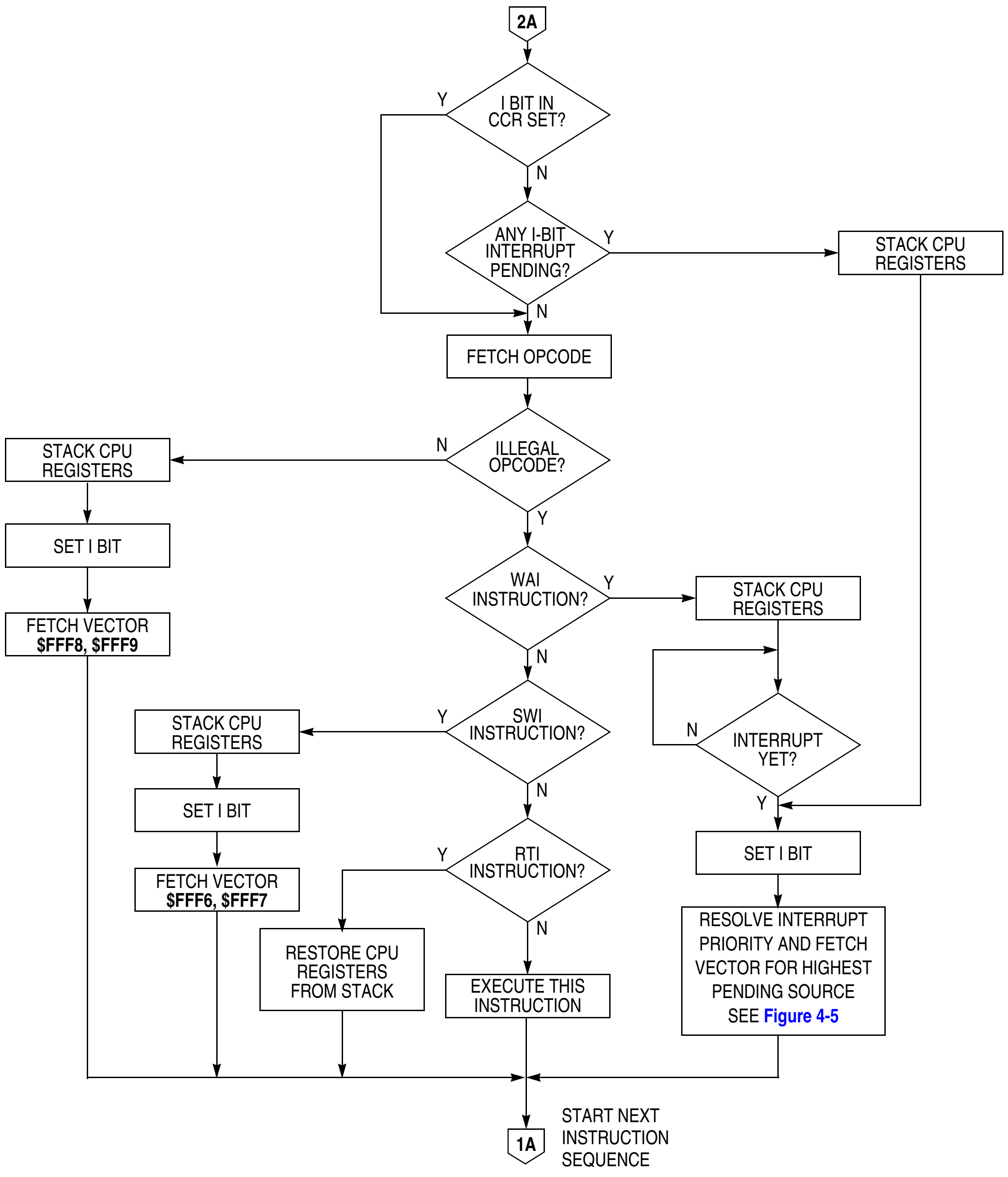


Figure 6-3. Processing Flow Out of Reset (Sheet 2 of 2)

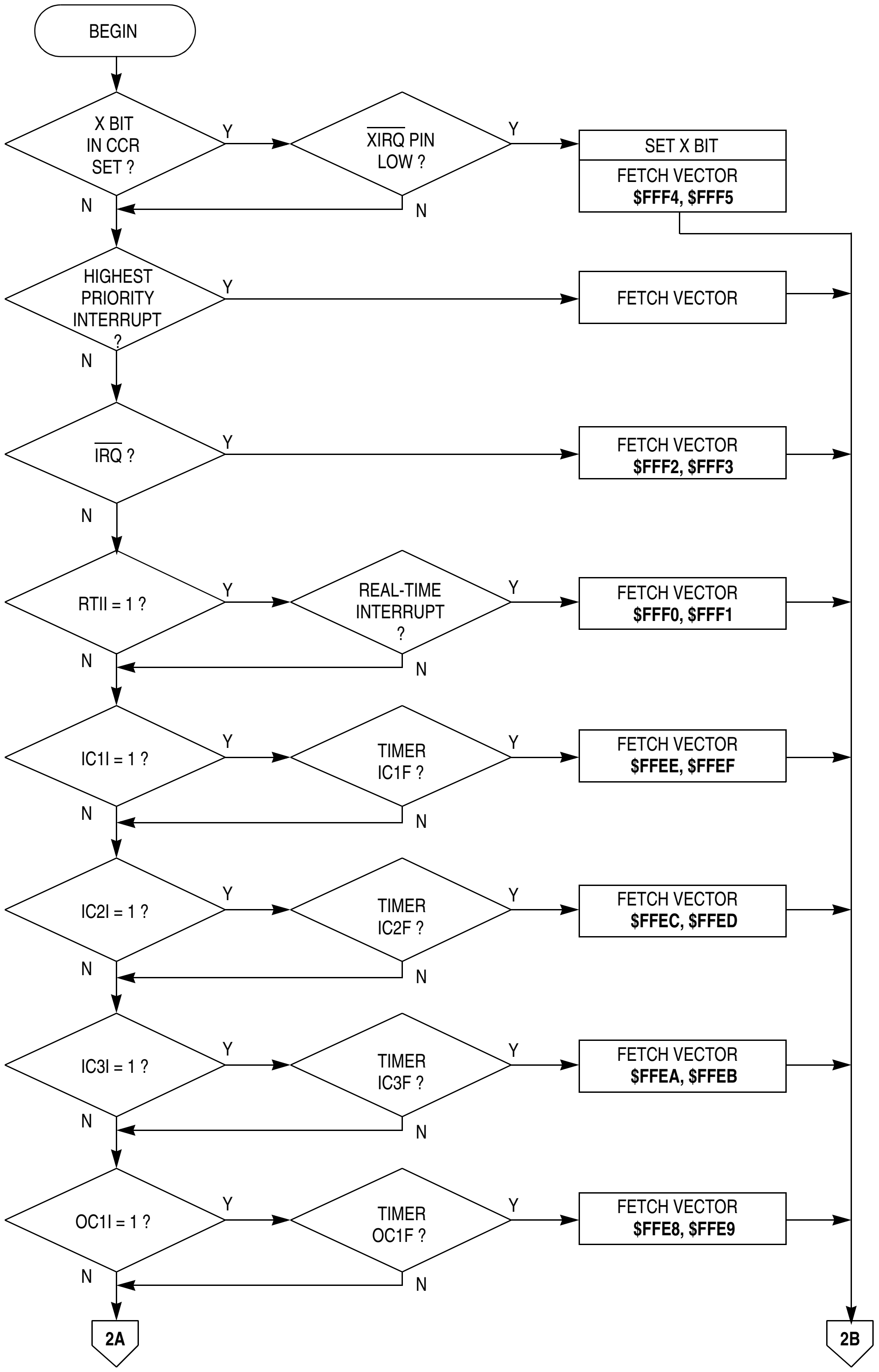

BEGIN
X BIT IN CCR SET ?
Y
N
XIRQ PIN LOW ?
SET X BIT
FETCH VECTOR $FFF4, $FFF5
HIGHEST PRIORITY INTERRUPT ?
FETCH VECTOR
IRQ ?
FETCH VECTOR $FFF2, $FFF3
RTII = 1 ?
REAL-TIME INTERRUPT ?
FETCH VECTOR $FFF0, $FFF1
IC1I = 1 ?
TIMER IC1F ?
FETCH VECTOR $FFEE, $FFEF
IC2I = 1 ?
TIMER IC2F ?
FETCH VECTOR $FFEC, $FFED
IC3I = 1 ?
TIMER IC3F ?
FETCH VECTOR $FFEA, $FFEB
OC1I = 1 ?
TIMER OC1F ?
FETCH VECTOR $FFE8, $FFE9
2A
2B

Figure 4-5. Interrupt Priority Resolution (Sheet 1 of 2)

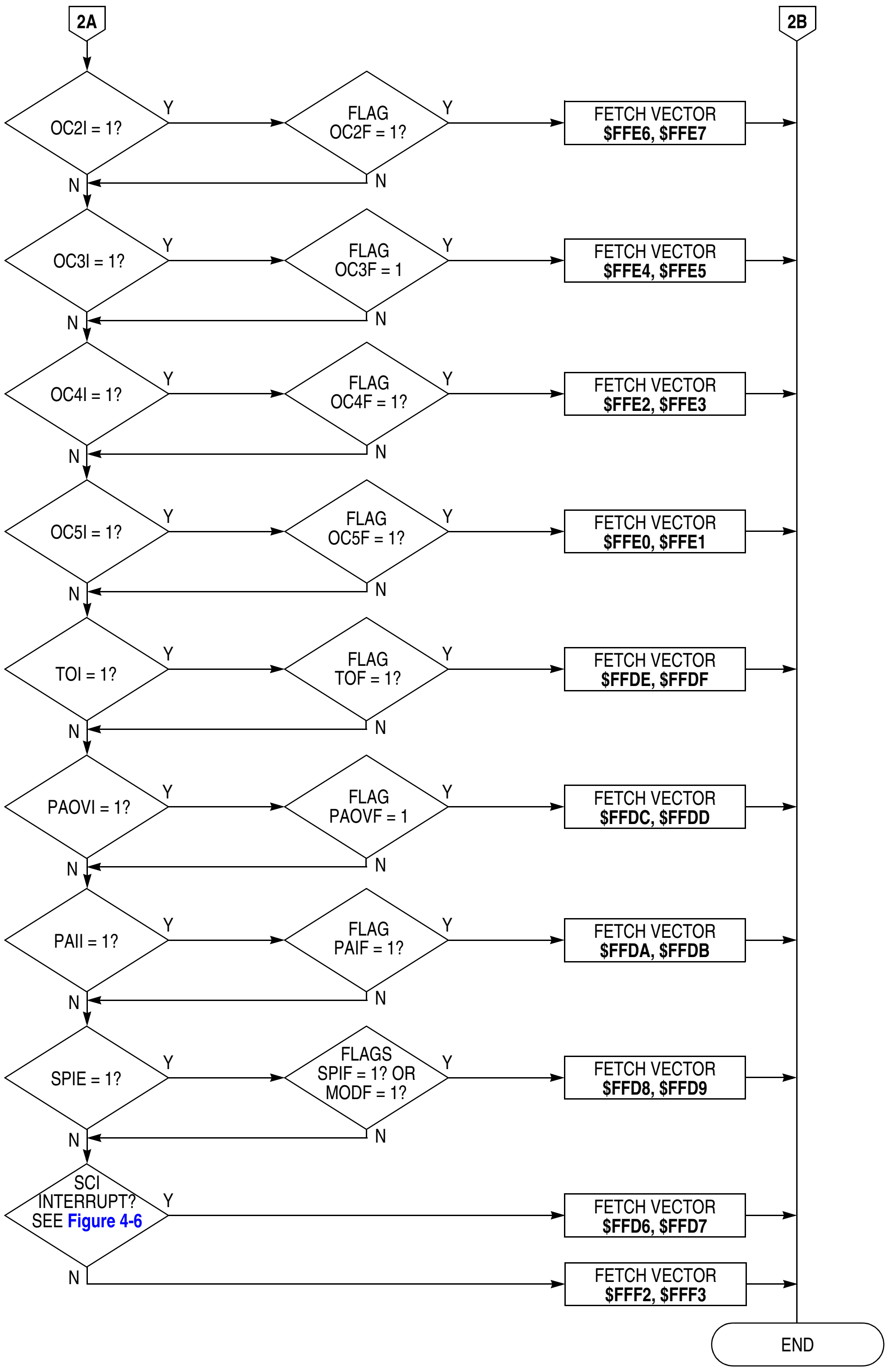

2A
2B
OC2I = 1?
FLAG OC2F = 1?
FETCH VECTOR $FFE6, $FFE7
OC3I = 1?
FLAG OC3F = 1
FETCH VECTOR $FFE4, $FFE5
OC4I = 1?
FLAG OC4F = 1?
FETCH VECTOR $FFE2, $FFE3
OC5I = 1?
FLAG OC5F = 1?
FETCH VECTOR $FFE0, $FFE1
TOI = 1?
FLAG TOF = 1?
FETCH VECTOR $FFDE, $FFDF
PAOVI = 1?
FLAG PAOVF = 1
FETCH VECTOR $FFDC, $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 4-6
FETCH VECTOR $FFD6, $FFD7
FETCH VECTOR $FFF2, $FFF3
Y
N
END

Figure 5-6. Interrupt Priority Resolution (Sheet 2 of 2)

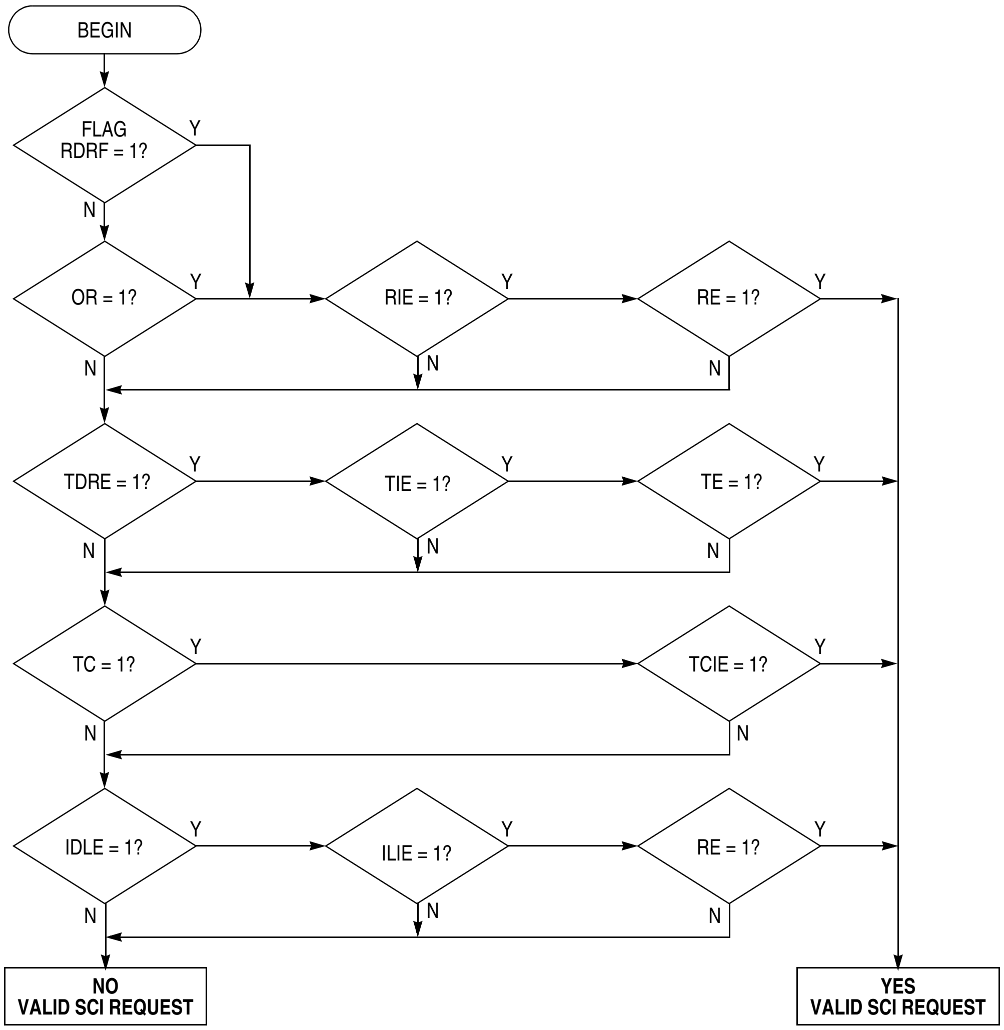


Figure 4-6. Interrupt Source Resolution within SCI

4.3.6 Highest Priority I Interrupt and Miscellaneous Register (HPRIO)

Four bits of this register (PSEL3–PSEL0) are used to select one of the I bit related interrupt sources and to elevate it to the highest I bit masked position of the priority resolution circuit. In addition, four miscellaneous system control bits are included in this register.

Address: $003C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	RBOOT	SMOD	MDA	IRVNE	PSEL3	PSEL2	PSEL1	PSEL0
Write:		▨						
Reset:	Note 1				0	1	0	1

1. The values of the RBOOT, SMOD, IRVNE, and MDA bits at reset depend on the mode during initialization. Refer to **Table 4-3**.

Figure 4-7. Highest Priority I-Bit Interrupt and Miscellaneous Register (HPRIO)

RBOOT — Read Bootstrap ROM
This bit can be read at any time. It can be written only in special modes (SMOD = 1). In special bootstrap mode, it is set during reset. Reset clears it in all other modes.
1 = Bootloader ROM is enabled in the memory map at $BF00–$BFFF.
0 = Bootloader ROM is disabled and is not in the memory map.

SMOD and MDA — Special Mode Select and Mode Select A
These two bits can be read at any time.These bits reflect the status of the MODA and MODB input pins at the rising edge of reset. SMOD may be written only in special modes. It cannot be written to a 1 after being cleared without an interim reset. MDA may be written at any time in special modes, but only once in normal modes. An interpretation of the values of these two bits is shown in **Table 4-3**.

Table 4-3. Hardware Mode Select Summary

Inputs		**Mode**	**Latched at Reset**	
MODB	**MODA**		**SMOD**	**MDA**
1	0	Single chip	0	0
1	1	Expanded multiplexed	0	1
0	0	Special bootstrap	1	0
0	1	Special test	1	1

IRVNE — Internal Read Visibility/Not E

This bit may be read at any time. It may be written once in any mode. IRVNE is set during reset in special test mode only, and cleared by reset in the other modes.

1 = Data from internal reads is driven out on the external data bus in expanded modes.
0 = Data from internal reads is not visible on the external data bus.

As shown in the table, in single-chip and bootstrap modes IRVNE determines whether the E clock is driven out or forced low.

1 = E pin driven low
0 = E clock driven out of the chip

Mode	IRVNE Out of Reset	E Clock Out of Reset	IRV Out of Reset	IRVNE Affects Only	IRVNE May be Written
Single chip	0	On	Off	E	Once
Expanded multiplexed	0	On	Off	IRV	Once
Bootstrap	0	On	Off	E	Once
Special test	1	On	On	IRV	Once

NOTE: *To prevent bus conflicts, when using internal read visibility, the user must disable all external devices from driving the data bus during any internal access.*

PSEL3–PSEL0 — Priority Selects

These four bits are used to specify one I bit related interrupt source, which then becomes the highest priority I bit related interrupt source. These bits may be written only while the I bit in the CCR is set, inhibiting I bit related interrupts. An interpretation of the value of these bits is shown in **Table 4-4**.

During reset, PSEL3–PSEL0 are initialized to 0101, which corresponds to reserved (default to $\overline{\text{IRQ}}$). $\overline{\text{IRQ}}$ becomes the highest priority I bit related interrupt source.

Table 4-4. Highest Priority Interrupt Selection

PSEL3–PSEL0	Interrupt Source Promoted
0 0 0 0	Timer overflow
0 0 0 1	Pulse accumulator overflow
0 0 1 0	Pulse accumulator input edge
0 0 1 1	SPI serial transfer complete
0 1 0 0	SCI serial system
0 1 0 1	Reserved (default to $\overline{\text{IRQ}}$)
0 1 1 0	$\overline{\text{IRQ}}$ (external pin)
0 1 1 1	Real-time interrupt
1 0 0 0	Timer input capture 1

Table 4-4. Highest Priority Interrupt Selection (Continued)

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

4.4 Low-Power Operation

The M68HC11 Family of microcontroller units (MCU) has two programmable low power-consumption modes: stop and wait. In the wait mode, the on-chip oscillator remains active. In the stop mode, the oscillator is stopped. This subsection describes these two low power-consumption modes.

4.4.1 Stop Mode

The STOP instruction places the MCU in its lowest power-consumption mode, provided the S bit in the CCR is cleared. In this mode, all clocks are stopped, thereby halting all internal processing.

To exit the stop mode, a low level must be applied to either the $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$, or $\overline{\text{RESET}}$ pin. An external interrupt used at $\overline{\text{IRQ}}$ is only effective if the I bit in the CCR is cleared. An external interrupt applied at the $\overline{\text{XIRQ}}$ input is effective, regardless of the setting of the X bit of the CCR. However, the actual recovery sequence differs, depending on the X bit setting. If the X bit is cleared, the MCU starts with the stacking sequence leading to the normal service of the $\overline{\text{XIRQ}}$ request. If the X bit is set, the processing always continues with the instruction immediately following the STOP instruction. A low input to the $\overline{\text{RESET}}$ pin always results in an exit from the stop mode, and the start of MCU operations is determined by the reset vector.

The CPU will not exit stop mode correctly when interrupted by $\overline{\text{IRQ}}$ or $\overline{\text{XIRQ}}$ if the instruction preceding STOP is a column 4 or 5 accumulator inherent (opcodes $4X and $5X) instruction, such as NEGA, NEGB, COMA, COMB, etc. These single-byte, two-cycle instructions must be followed by an NOP, then the STOP command. If reset is used to exit stop mode, the CPU will respond properly.

A restart delay is required if the internal oscillator is being used. The delay allows the oscillator to stabilize when exiting the stop mode. If a stable external oscillator is being used, the delay (DLY) bit in the OPTION register can be cleared to bypass the delay. If the DLY bit is clear, the $\overline{\text{RESET}}$ pin would not normally be used to exit the stop mode. The reset sequence sets the DLY bit, and the restart delay would be reimposed.

4.4.2 Wait Mode

The wait (WAI) instruction places the MCU in a low power-consumption mode. The wait mode consumes more power than the stop mode since the oscillator is kept running. Upon execution of the WAI instruction, the machine state is stacked and program execution stops.

The wait state can be exited only by an unmasked interrupt or $\overline{\text{RESET}}$. If the I bit of the CCR is set and the COP is disabled, the timer system is turned off by WAI to further reduce power consumption. The amount of power savings is application dependent. It also depends upon the circuitry connected to the MCU pins, and upon subsystems such as the timer, serial peripheral interface (SPI), or serial communications interface (SCI) that were or were not active when the wait mode was entered.

Section 5. Input/Output (I/O) Ports

5.1 Introduction

The MC68HC711D3 has four 8-bit input/output (I/O) ports; A, B, C, and D. In the 40-pin version, port A bits 4 and 6 are not bonded. Port functions are controlled by the particular mode of operation selected, as shown in **Table 1-1. Port Signal Functions**.

In the single-chip and bootstrap modes, all the ports are configured as parallel input/output (I/O) data ports. In expanded multiplexed and test modes, ports B, C, and lines D6 (AS) and D7 (R/$\overline{\text{W}}$) are configured as a memory expansion bus, with:

- Port B as the high-order address bus
- Port C as the multiplexed address and data bus
- AS as the demultiplexing signal
- R/$\overline{\text{W}}$ as data bus direction control

The remaining ports are unaffected by mode changes.

- Ports A and D can be used as general-purpose I/O ports, though each has an alternate function.
- Port A bits handle the timer functions.
- Port D handles serial peripheral interface (SPI) and serial communications interface (SCI) functions in addition to its bus direction control functions.

5.2 Port A

Port A shares functions with the timer system and has:

- Three input only pins
- Three output only pins
- Two bidirectional I/O pins

Pins PA6 and PA4 are not bonded in the 40-pin dual in-line package (DIP), and their OC output functions are unavailable, but their software interrupts are available.

Address: $0000

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	PA7	PA6[1]	PA5	PA4[1]	PA3	PA2	PA1	PA0
Reset:	Hi-Z	0	0	0	Hi-Z	Hi-Z	Hi-Z	Hi-Z
Alt. Func.:	PAI	OC2	OC3	OC4	IC4/OC5	IC1	IC2	IC3
And/Or:	OC1	OC1	OC1	OC1	OC1	—	—	—

1. This pin is not bonded in the 40-pin version.

Figure 5-1. Port A Data Register (PORTA)

PORTA can be read any time. Inputs return the pin level, whereas outputs return the pin driver input level. If written, PORTA stores the data in an internal latch. It drives the pins only if they are configured as outputs. Writes to PORTA do not change the pin state when the pins are configured for timer output compares.

Out of reset, port A bits 7 and 3–0 are general high-impedance inputs, while bits 6–4 are outputs, driving low. On bidirectional lines PA7 and PA3, the timer forces the I/O state to be an output if the associated output compare is enabled. In this case, the data direction bits DDRA7 and DDRA3 in PACTL will not be changed or have any effect on those bits. When the output compare functions associated with these pins are disabled, the DDR bits in PACTL govern the I/O state.

5.3 Port B

Port B is an 8-bit, general-purpose I/O port with a data register (PORTB) and a data direction register (DDRB).

- In the single-chip mode, port B pins are general-purpose I/O pins (PB7–PB0).
- In the expanded-multiplexed mode, all of the port B pins act as the high-order address bits (A15–A8) of the address bus.

5.3.1 Port B Data Register

Address: $0004

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Reset:	0	0	0	0	0	0	0	0
Alt. Func.:	A15	A14	A13	A12	A11	A10	A9	A8

Figure 5-2. Port B Data Register (PORTB)

PORTB can be read at any time. Inputs return the sensed levels at the pin, while outputs return the input level of the port B pin drivers. If PORTB is written, the data is stored in an internal latch and can be driven only if port B is configured for general-purpose outputs in single-chip or bootstrap mode.

Port B pins are general--purpose inputs out of reset in single-chip and bootstrap modes. These pins are outputs (the high-order address bits) out of reset in expanded multiplexed and test modes.

5.3.2 Port B Data Direction Register

Address: $0006

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
Reset:	0	0	0	0	0	0	0	0

Figure 5-3. Data Direction Register for Port B (DDRB)

DDB7–DDB0 — Data Direction Bits for Port B
1 = Corresponding port B pin configured as output
0 = Corresponding port B pin configured for input only

5.4 Port C

Port C is an 8-bit, general-purpose I/O port with a data register (PORTC) and a data direction register (DDRC). In the single-chip mode, port C pins are general-purpose I/O pins (PC7–PC0). In the expanded-multiplexed mode, port C pins are configured as multiplexed address/data pins. During the address cycle, bits 7–0 of the address are output on PC7–PC0. During the data cycle, bits 7–0 (PC7–PC0) are bidirectional data pins controlled by the R/$\overline{W}$ signal.

5.4.1 Port C Control Register

Address: $0002

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

Figure 5-4. Port C Control Register (PIOC)

CWOM — Port C Wire-OR Mode Bit
1 = Port C outputs are open drain (to facilitate testing)
0 = Port C operates normally

5.4.2 Port C Data Register

Address: $0003

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Reset:	0	0	0	0	0	0	0	0

Figure 5-5. Port C Data Register (PORTC)

PORTC can be read at any time. Inputs return the sensed levels at the pin, while outputs return the input level of the port C pin drivers. If PORTC is written, the data is stored in an internal latch and can be driven only if port C is configured for general-purpose outputs in single-chip or bootstrap mode.

Port C pins are general-purpose inputs out of reset in single-chip and bootstrap modes. These pins are multiplexed low-order address and data bus lines out of reset in expanded-multiplexed and test modes.

5.4.3 Port C Data Direction Register

Address: $0007

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

Figure 5-6. Data Direction Register for Port C (DDRC)

DDC7–DDC0 — Data Direction Bits for Port C
1 = Corresponding port C pin is configured as output
0 = Corresponding port C pin is configured for input only

5.5 Port D

Port D is an 8-bit, general-purpose I/O port with a data register (PORTD) and a data direction register (DDRD). The eight port D bits (D7–D0) can be used for general-purpose I/O, for the serial communications interface (SCI) and serial peripheral interface (SPI) subsystems, or for bus data direction control

5.5.1 Port D Data Register

Address: $0008

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Reset:	0	0	0	0	0	0	0	0

Figure 5-7. Port D Data Register (PORTD)

PORTD can be read at any time and inputs return the sensed levels at the pin; whereas, outputs return the input level of the port D pin drivers. If PORTD is written, the data is stored in an internal latch, and can be driven only if port D is configured as general-purpose output. This port shares functions with the on-chip SCI and SPI subsystems, while bits 6 and 7 control the direction of data flow on the bus in expanded and special test modes.

5.5.2 Port D Data Direction Register

Address: $0009

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
Reset:	0	0	0	0	0	0	0	0

Figure 5-8. Data Direction Register for Port D (DDRD)

DDD7–DDD0 — Data Direction for Port D

When port D is a general-purpose I/O port, the DDRD register controls the direction of the I/O pins as follows:

0 = Configures the corresponding port D pin for input only

1 = Configures the corresponding port D pin for output

In expanded and test modes, bits 6 and 7 are dedicated AS and R/$\overline{W}$.

When port D is functioning with the SPI system enabled, bit 5 is dedicated as the slave select ($\overline{SS}$) input. In SPI slave mode, DDD5 has no meaning or effect. In SPI master mode, DDD5 affects port D bit 5 as follows:

0 = Port D bit 5 is an error-detect input to the SPI.

1 = Port D bit 5 is configured as a general-purpose output line.

If the SPI is enabled and expects port D bits 2, 3, and 4 (MISO, MOSI, and SCK) to be inputs, then they are inputs, regardless of the state of DDRD bits 2, 3, and 4. If the SPI expects port D bits 2, 3, and 4 to be outputs, they are outputs only if DDRD bits 2, 3, and 4 are set.

Section 6. Serial Communications Interface (SCI)

6.1 Introduction

The serial communications interface (SCI) is a universal asynchronous receiver transmitter (UART), one of two independent serial input/output (I/O) subsystems in the MC68HC711D3. It has a standard non-return 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.

6.2 Data Format

The serial data format requires these conditions:

- An idle line in the high state before transmission or reception of a message
- A start bit, logic 0, transmitted or received, that indicates the start of each character
- Data that is transmitted and received least significant bit (LSB) first
- A stop bit, logic 1, 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.
- A break, defined as the transmission or reception of a logic 0 for some multiple number of frames

Selection of the word length is controlled by the M bit in the SCI control register 1 (SCCR1).

6.3 Transmit Operation

The SCI transmitter includes a parallel transmit data register (SCDR) and a serial shift register that puts data from the SCDR into serial form. 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 PD1 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 6-1**, shows the transmit serial shift register and the buffer logic at the top of the figure.

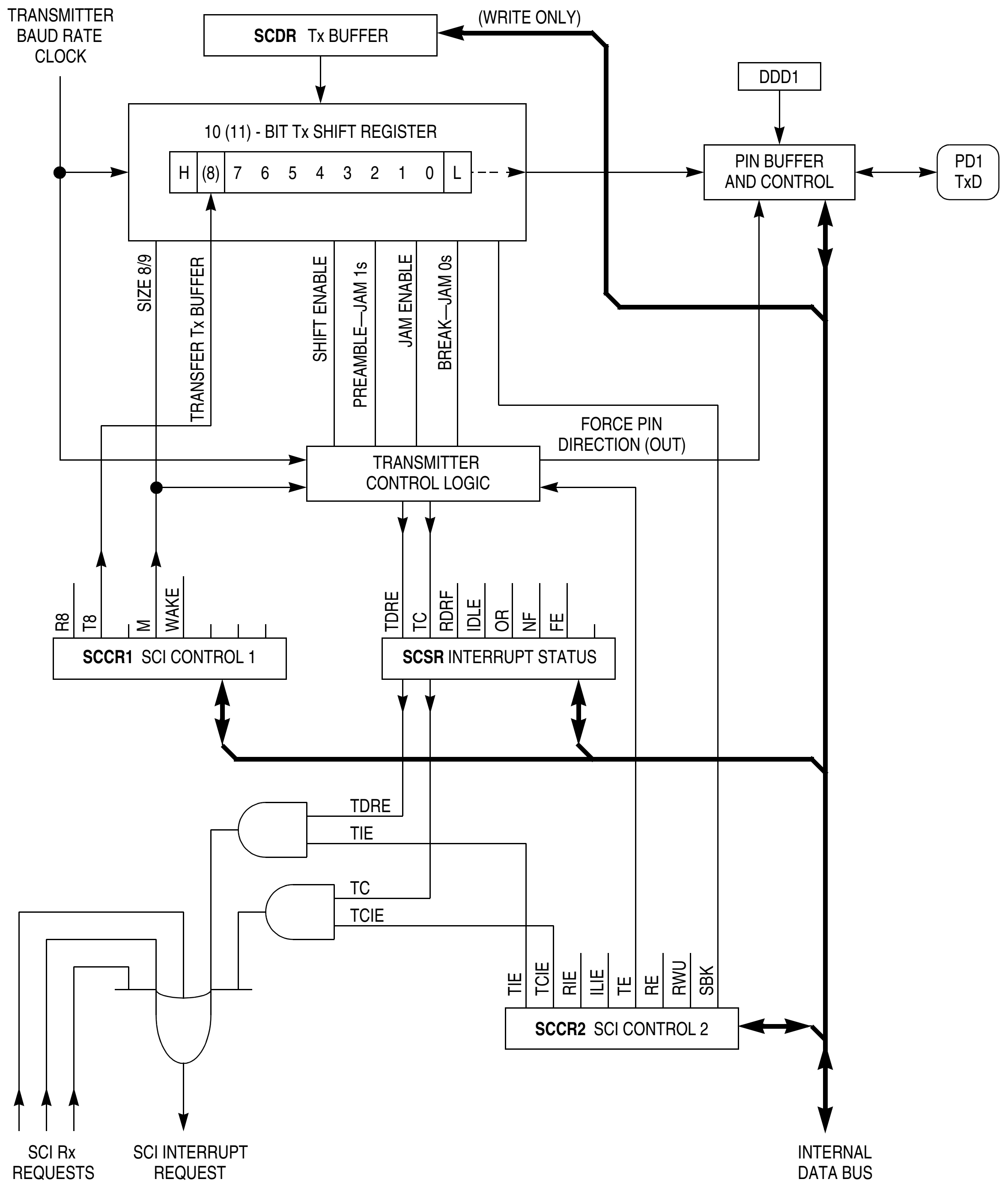


Figure 6-1. SCI Transmitter Block Diagram

6.4 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. Refer to **Figure 6-2**. 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.

6.5 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 1 to the RWU bit in the SCCR2 register. While RWU is 1, 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
- 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 1 in the most significant bit (MSB) of a character wakes up all sleeping receivers.

6.5.1 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.

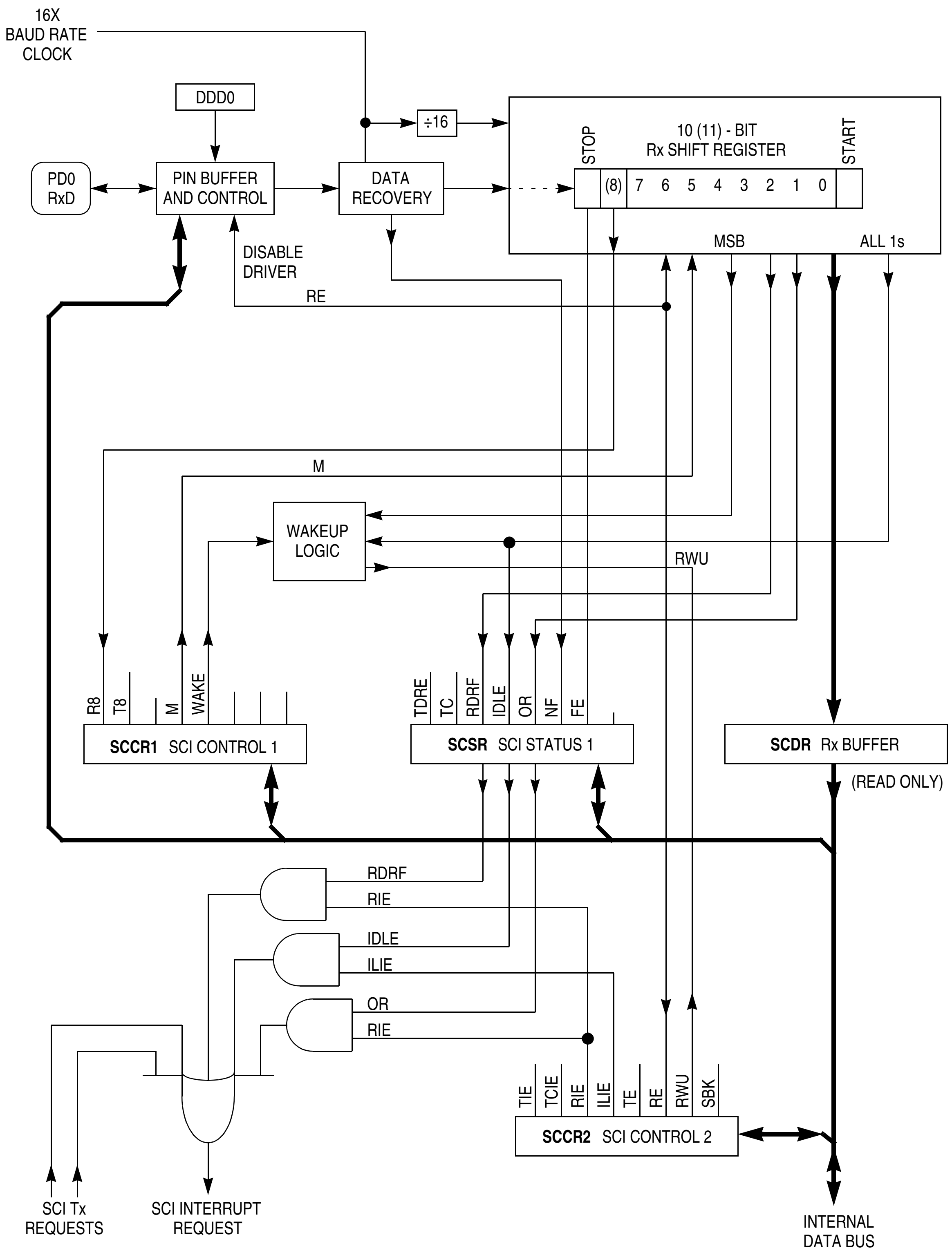


Figure 6-2. SCI Receiver Block Diagram

6.5.2 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 1 — 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.

6.6 SCI Error Detection

Three error conditions can occur during generation of SCI system interrupts:

- Serial communications data register (SCDR) overrun
- Received bit noise
- Framing

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 1) 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 1) followed by a read of the SCDR.

6.7 SCI Registers

This subsection describes the five addressable registers in the SCI.

6.7.1 SCI Data Register

The SCI 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.

Address: $002F

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

Figure 6-3. SCI Data Register (SCDR)

6.7.2 SCI Control Register 1

The SCI control register 1 (SCCR1) provides the control bits that determine word length and select the method used for the wakeup feature.

Address: $002C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	R8	T8	0	M	WAKE	0	0	0
Reset:	U	U	0	0	0	0	0	0

U = Unaffected

Figure 6-4. SCI Control Register 1 (SCCR1)

R8 — Receive Data Bit 8
If M bit is set, R8 stores the ninth bit in the receive data character.

T8 — Transmit Data Bit 8
If M bit is set, T8 stores ninth bit in transmit data character.

M — Mode Bit
The mode bit selects 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 Bit
0 = Wakeup by IDLE line recognition
1 = Wakeup by address mark (most significant data bit set)

6.7.3 SCI Control Register 2

The SCI control register 2 (SCCR2) provides the control bits that enable or disable individual SCI functions.

Address: $002D

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

Figure 6-5. SCI Control Register 2 (SCCR2)

TIE — Transmit Interrupt Enable Bit
1 = TDRE interrupts disabled
1 = SCI interrupt requested when TDRE status flag is set

TCIE — Transmit Complete Interrupt Enable Bit
0 = TC interrupts disabled
1 = SCI interrupt requested when TC status flag is set

RIE — Receiver Interrupt Enable Bit
0 = RDRF and OR interrupts disabled
1 = SCI interrupt requested when RDRF flag or the OR status flag is set

ILIE — Idle Line Interrupt Enable Bit
1 = IDLE interrupts disabled
1 = SCI interrupt requested when IDLE status flag is set

TE — Transmitter Enable Bit
When TE goes from 0 to 1, one unit of idle character time (logic 1) is queued as a preamble.
0 = Transmitter disabled
1 = Transmitter enabled

RE — Receiver Enable Bit
0 = Receiver disabled
1 = Receiver enabled

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

SBK — Send Break Bit
At least one character time of break is queued and sent each time SBK is written to 1. 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 1 and writing the 0 to SBK.
0 = Break generator off
1 = Break codes generated as long as SBK = 1

6.7.4 SCI Status Register

The SCI status register (SCSR) provides inputs to the interrupt logic circuits for generation of the SCI system interrupt.

Address: $002E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	TDRE	TC	RDRF	IDLE	OR	NF	FE	0
Reset:	1	1	0	0	0	0	0	0

Figure 6-6. SCI Status Register (SCSR)

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 active
1 = RxD line 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 Bit

FE is set when a 0 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

6.7.5 Baud Rate Register

The baud rate register (BAUD) is used to select different baud rates for the SCI system. The SCP1 and SCP0 bits function as a prescaler for the SCR2–SCR0 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 6-1** and **Table 6-2** for normal baud rate selections.

Address: $002B

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0
Reset:	0	0	0	0	0	U	U	U

U = Unaffected

Figure 6-7. Baud Rate Register (BAUD)

TCLR — Clear Baud Rate Counters (Test)

RCKB — SCI Baud Rate Clock Check (Test)

SCP1 and SCP0 — SCI Baud Rate Prescaler Select Bits

These two bits select a prescale factor for the SCI baud rate generator that determines the highest possible baud rate.

Table 6-1. Baud Rate Prescale Selects

SCP1 and SCP0	Divide Internal Clock By	Crystal Frequency in MHz			
		4.0 MHz (Baud)	8.0 MHz (Baud)	10.0 MHz (Baud)	12.0 MHz (Baud)
0 0	1	62.50 K	125.0 K	156.25 K	187.5 K
0 1	3	20.83 K	41.67 K	52.08 K	62.5 K
1 0	4	15.625 K	31.25 K	38.4 K	46.88 K
1 1	13	4800	9600	12.02 K	14.42 K

SCR2–SCR0 — SCI Baud Rate Select Bits

These three bits select receiver and transmitter bit rate based on output from baud rate prescaler stage.

Table 6-2. Baud Rate Selects

SCR2–SCR0	Divide Prescaler By	Highest Baud Rate (Prescaler Output from Table 6-1)		
		4800	**9600**	**38.4 K**
0 0 0	1	4800	9600	38.4 K
0 0 1	2	2400	4800	19.2 K
0 1 0	4	1200	2400	9600
0 1 1	8	600	1200	4800
1 0 0	16	300	600	2400
1 0 1	32	150	300	1200
1 1 0	64	—	150	600
1 1 1	128	—	—	300

The prescale bits, SCP1 and SCP0, determine the highest baud rate and the SCR2–SCR0 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 16 X receiver baud rate clock. The SCR2–SCR0 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 6-8 illustrates the SCI baud rate timing chain. The prescale 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.

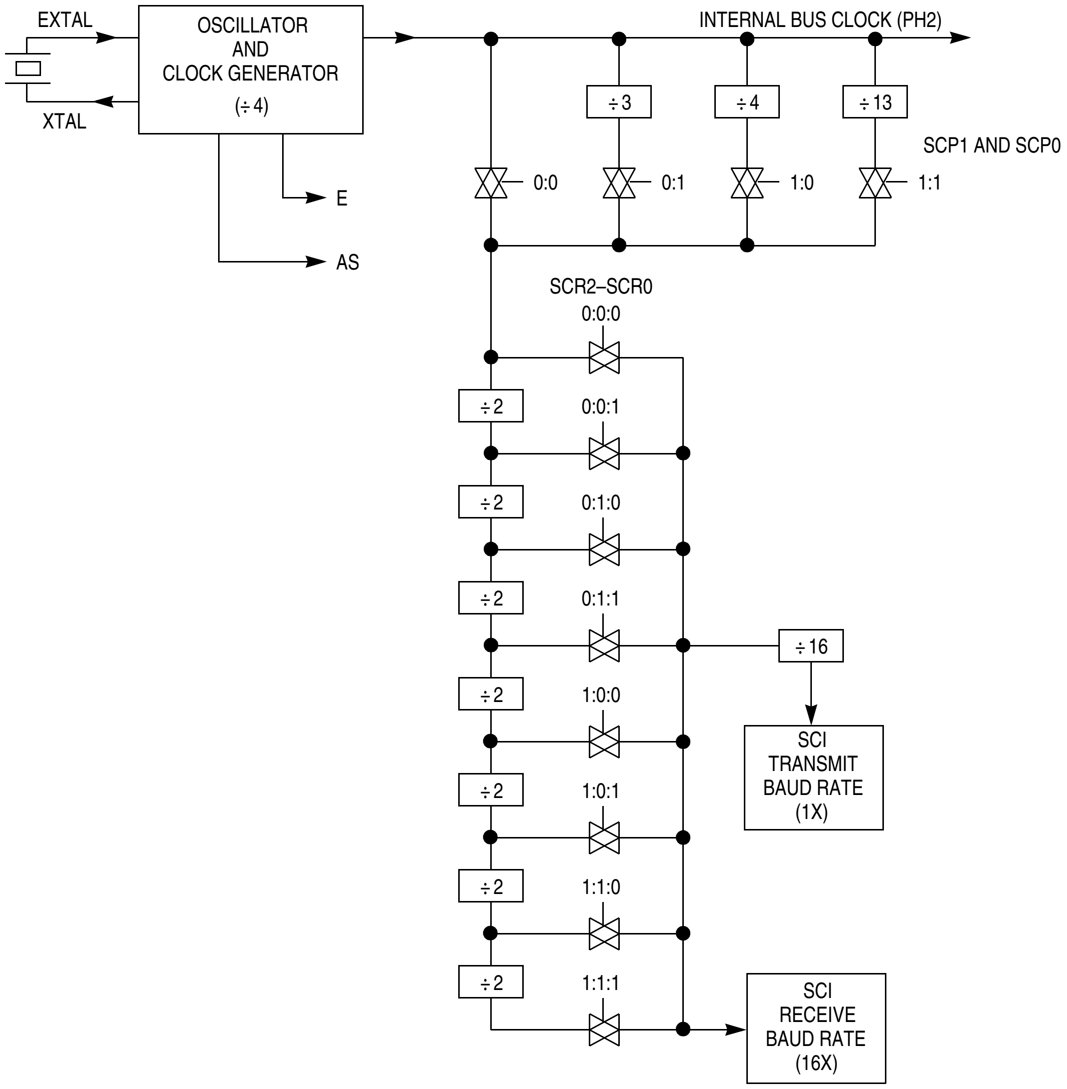


Figure 6-8. SCI Baud Rate Diagram

6.8 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 1). 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 0, TDRE must be polled. When TIE and TDRE are 1, 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 0, TC must be polled; when TCIE is 1 and TC is 1, an interrupt is requested.

Writing a 0 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 0 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 0, 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.

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 6-9**, 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 1 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.

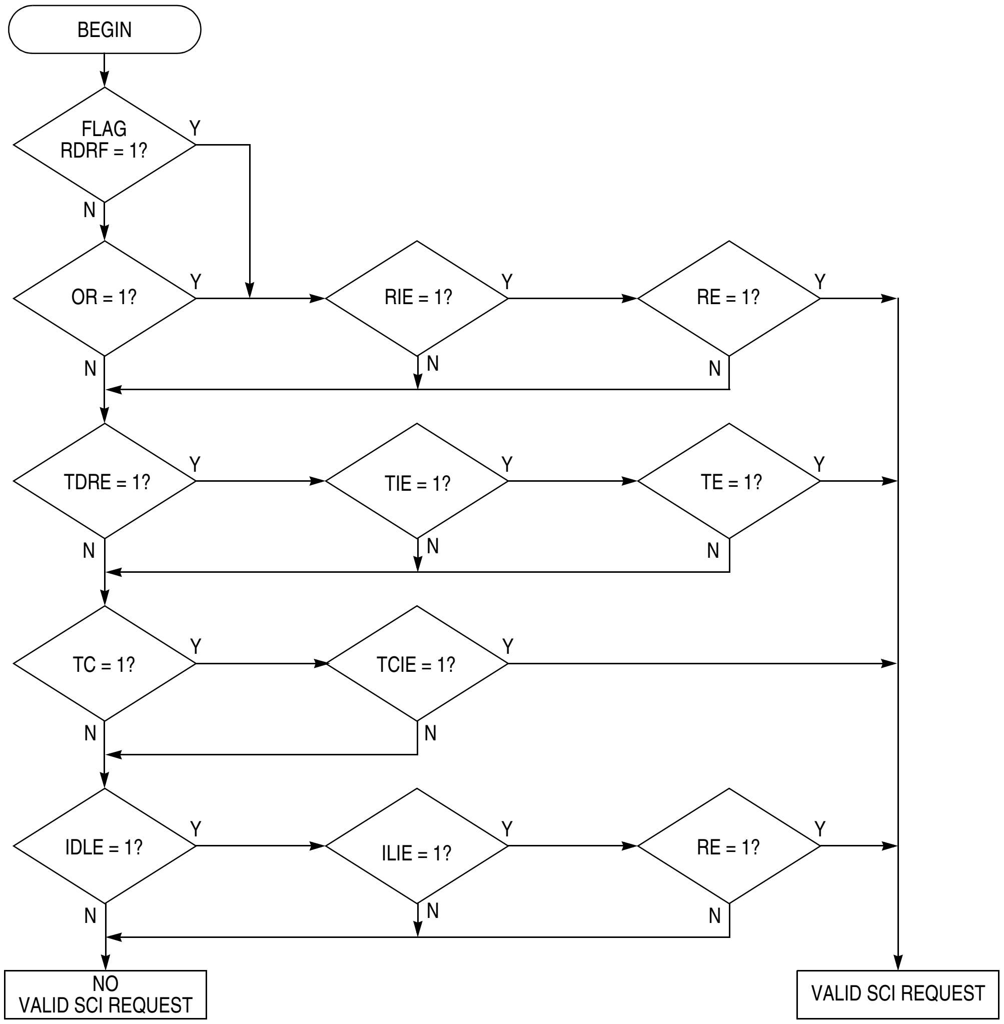


Figure 6-9. Interrupt Source Resolution within SCI

Section 7. Serial Peripheral Interface (SPI)

7.1 Introduction

The serial peripheral interface (SPI), an independent serial communications subsystem, allows the microcontroller unit (MCU) to communicate synchronously with peripheral devices, such as:

- Transistor-transistor logic (TTL) shift registers
- Liquid crystal diode (LCD) display drivers
- Analog-to-digital converter (ADC) subsystems
- Other microprocessors (MCUs)

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 with data rates as high as one half of the E-clock rate when configured as master, and as fast as the E-clock rate when configured as slave.

7.2 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 7-1**, which shows the SPI block diagram.

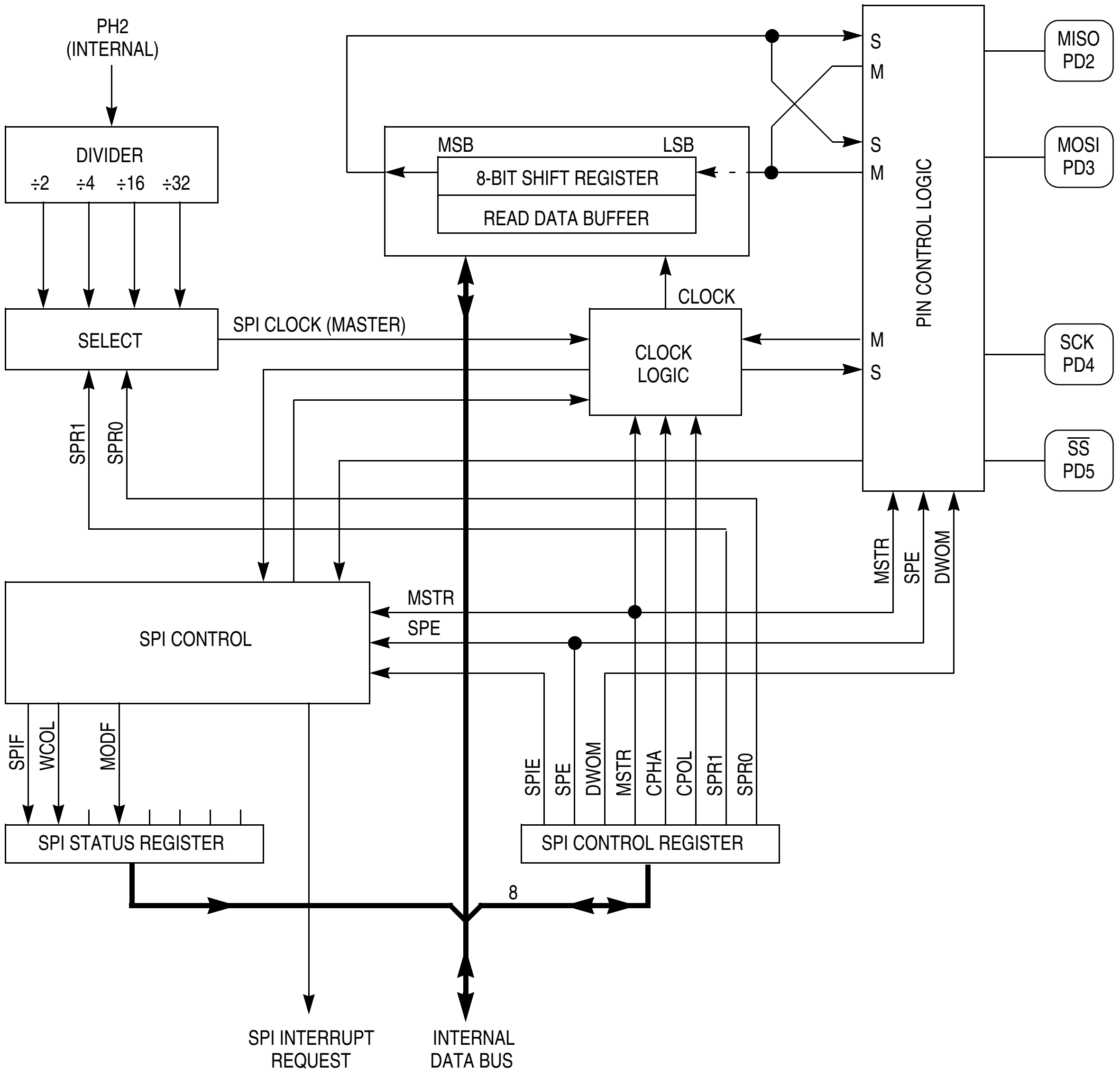


Figure 7-1. SPI Block Diagram

7.3 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 7-2**.

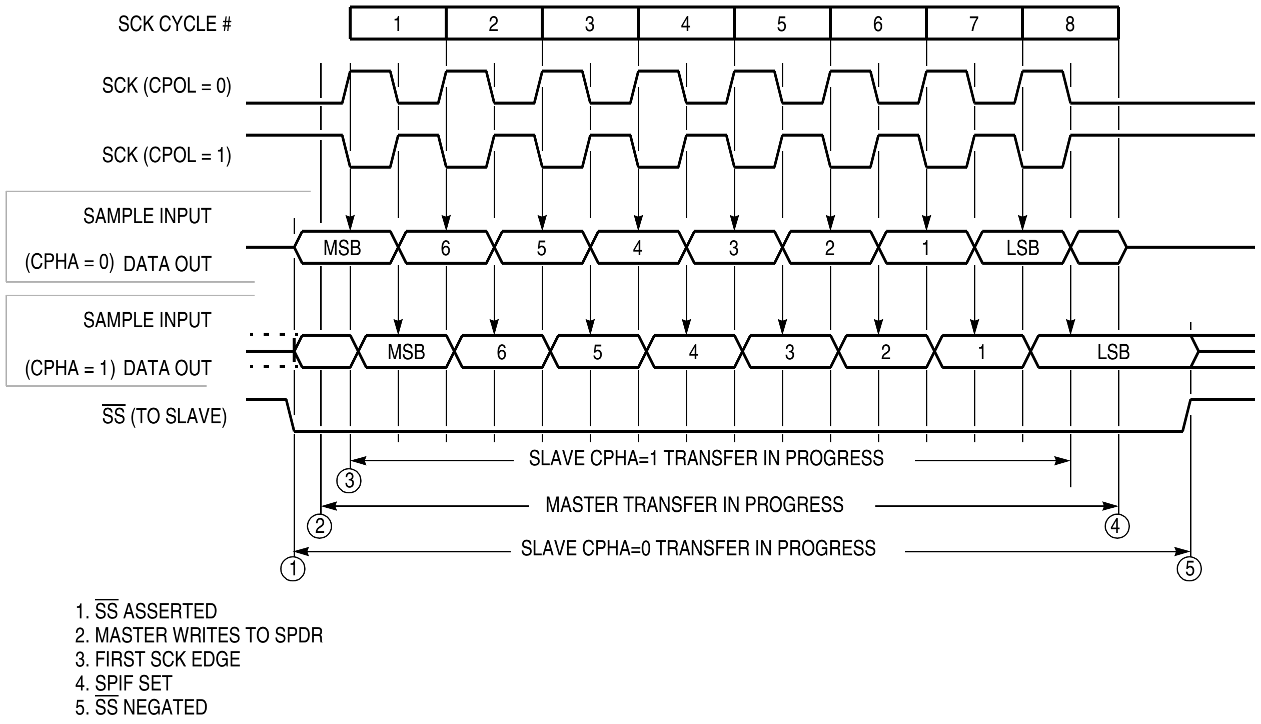


Figure 7-2. SPI Transfer Format

7.4 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 0, the slave select ($\overline{SS}$) 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 active low, a write collision error results.

When CPHA equals 1, the $\overline{SS}$ line can remain low between successive transfers.

7.5 SPI Signals

This subsection contains description of the four SPI signals:

- Master in/slave out (MISO)
- Master out/slave in (MOSI)
- Serial clock (SCK)
- Slave select ($\overline{SS}$)

7.5.1 Master In/Slave Out (MISO)

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.

7.5.2 Master Out/Slave In (MOSI)

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.

7.5.3 Serial Clock (SCK)

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.

Four possible timing relationships 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, SPR1 and SPR0, in the SPCR of the master device, select the clock rate. In a slave device, SPR1 and SPR0 have no effect on the operation of the SPI.

7.5.4 Slave Select ($\overline{SS}$)

The $\overline{SS}$ 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 1 in bit 5 of the port D data direction register. This sets the $\overline{SS}$ pin to act as a general-purpose output. 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.

7.6 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 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. 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. For a slave, after a byte transfer, SCK must be in inactive state for at least 2 E-clock cycles before the next byte transfer begins.

7.7 SPI Registers

The three SPI registers, SPCR, SPSR, and SPDR, provide control, status, and data storage functions. This sub-section provides a description of how these registers are organized.

7.7.1 SPI Control Register

Address: $0028

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

U = Unaffected

Figure 7-3. SPI Control Register (SPCR)

SPIE — Serial Peripheral Interrupt Enable Bit
0 = SPI interrupt disabled
1 = SPI interrupt enabled

SPE — Serial Peripheral System Enable Bit
0 = SPI off
1 = SPI on

DWOM — Port D Wired-OR Mode Bit
DWOM affects all six port D pins.
0 = Normal CMOS outputs
1 = Open-drain outputs

MSTR — Master Mode Select Bit
0 = Slave mode
1 = Master mode

CPOL — Clock Polarity Bit
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 7-2** and **7.4 Clock Phase and Polarity Controls**.

CPHA — Clock Phase Bit
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 7-2** and **7.4 Clock Phase and Polarity Controls**.

SPR1 and SPR0 — SPI Clock Rate Select Bits

These two serial peripheral rate bits select one of four baud rates to be used as SCK if the device is a master; however, they have no effect in the slave mode.

Table 7-1. SPI Clock Rates

SPR1 and SPR0	E Clock Divide By	Frequency at E = 2 MHz (Baud)
0 0	2	1.0 MHz
0 1	4	500 kHz
1 0	16	125 kHz
1 1	32	62.5 kHz

7.7.2 SPI Status Register

Address: $0029

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	SPIF	WCOL	0	MODF	0	0	0	0
Reset:	0	0	0	0	0	0	0	0

Figure 7-4. SPI Status Register (SPSR)

SPIF — SPI Transfer 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 Bit

Clearing the WCOL bit is accomplished by reading the SPSR (with WCOL set) followed by an access of SPDR. Refer to **7.5.4 Slave Select (SS)** and **7.6 SPI System Errors**.

0 = No write collision
1 = Write collision

Bit 5 — Not implemented

Always reads 0.

MODF — Mode Fault Bit

To clear the MODF bit, read the SPSR (with MODF set), then write to the SPCR. Refer to **7.5.4 Slave Select (SS)** and **7.6 SPI System Errors**.

0 = No mode fault
1 = Mode fault

Bits 3–0 — Not implemented

Always reads 0

7.7.3 SPI Data I/O Register

The SPI data I/O register (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.

Address: $002A

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset:	Unaffected by reset							

Figure 7-5. SPI Data I/O Register (SPDR)

NOTE: *SPI is double buffered in and single buffered out.*

Section 8. Programmable Timer

8.1 Introduction

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. Refer to **Figure 8-1**.

All main timer system activities are referenced to this free-running counter. The counter begins incrementing from $0000 as the microcontroller unit (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 or gated time accumulation modes. 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.

RTI is a programmable periodic interrupt circuit that permits pacing the execution of software routines by selecting one of four interrupt rates.

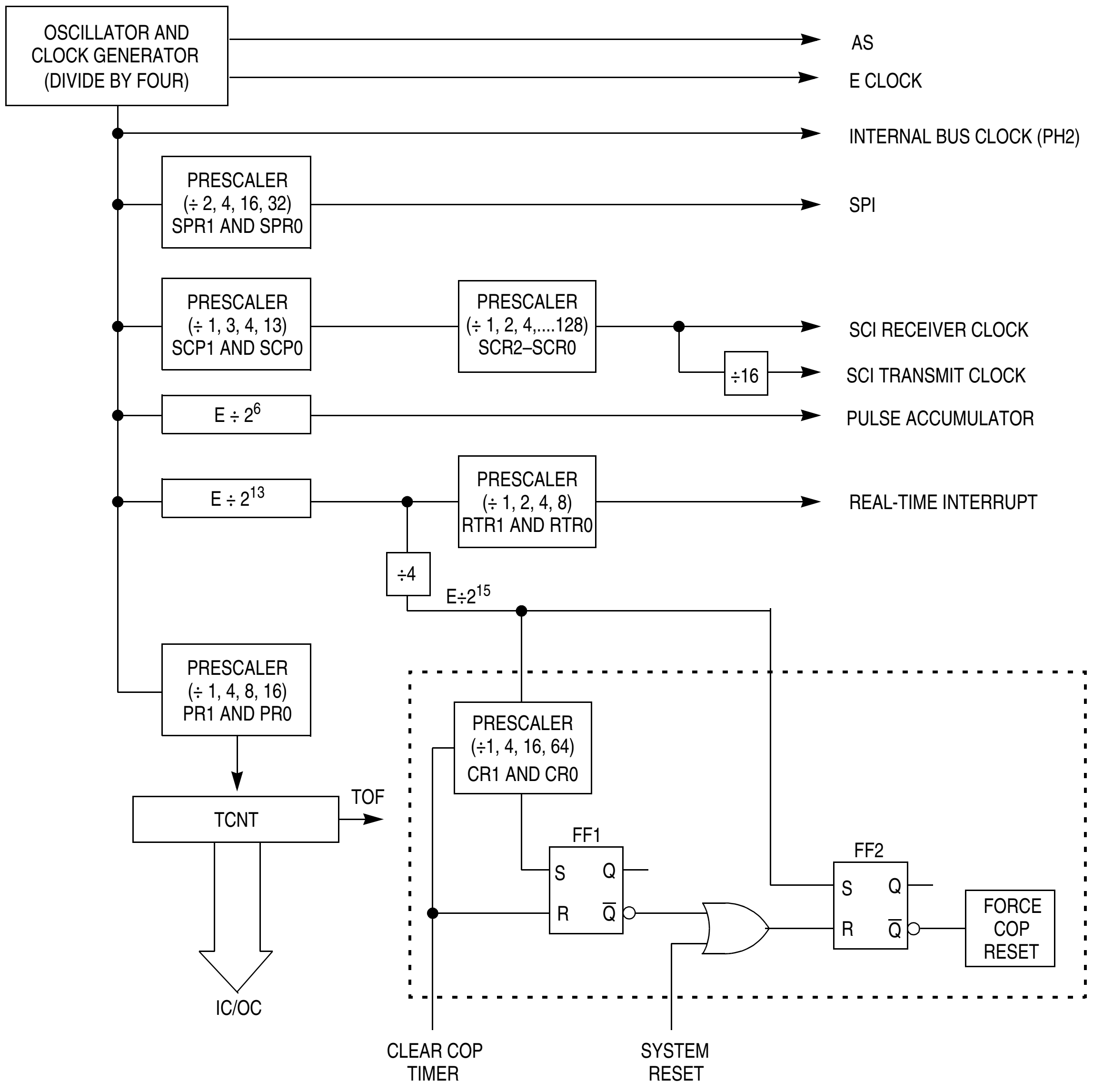

OSCILLATOR AND CLOCK GENERATOR (DIVIDE BY FOUR)
AS
E CLOCK
INTERNAL BUS CLOCK (PH2)
PRESCALER (÷ 2, 4, 16, 32) SPR1 AND SPR0
SPI
PRESCALER (÷ 1, 3, 4, 13) SCP1 AND SCP0
PRESCALER (÷ 1, 2, 4,....128) SCR2–SCR0
SCI RECEIVER CLOCK
÷16
SCI TRANSMIT CLOCK
E ÷ 2^6
PULSE ACCUMULATOR
E ÷ 2^13
PRESCALER (÷ 1, 2, 4, 8) RTR1 AND RTR0
REAL-TIME INTERRUPT
÷4
E÷2^15
PRESCALER (÷ 1, 4, 8, 16) PR1 AND PR0
TCNT
TOF
IC/OC
PRESCALER (÷1, 4, 16, 64) CR1 AND CR0
FF1
S Q
R Q̄
FF2
S Q
R Q̄
FORCE COP RESET
CLEAR COP TIMER
SYSTEM RESET

Figure 8-1. Timer Clock Divider Chains

The COP watchdog clock input (E÷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 8-1** for crystal related frequencies and periods.

Table 8-1. Timer Summary

Control Bits	**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)**
PR1 and PR0	**Main Timer Count Rates**			
0 0 1 count — overflow —	1.0 µs 65.536 ms	500 ns 32.768 ms	333 ns 21.845 ms	(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})

8.2 Timer Structure

Figure 8-2 shows the capture/compare system block diagram. The port A pin control block includes logic for timer functions and for general-purpose input/output (I/O). For pins 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 PA2–PA0 can be read at any time (read PORTA register), even if the pin is being used for the input capture function. Pins PA6–PA4 are used for either general-purpose output or as output compare pins. Pin PA3 can be used for general-purpose I/O, input capture 4, output compare 5, or output compare 1. 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 (OC5–OC2) is related to one of the port A output pins. Output compare 1 (OC1) has extra control logic, allowing it optional control of any combination of the PA7–PA3 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.

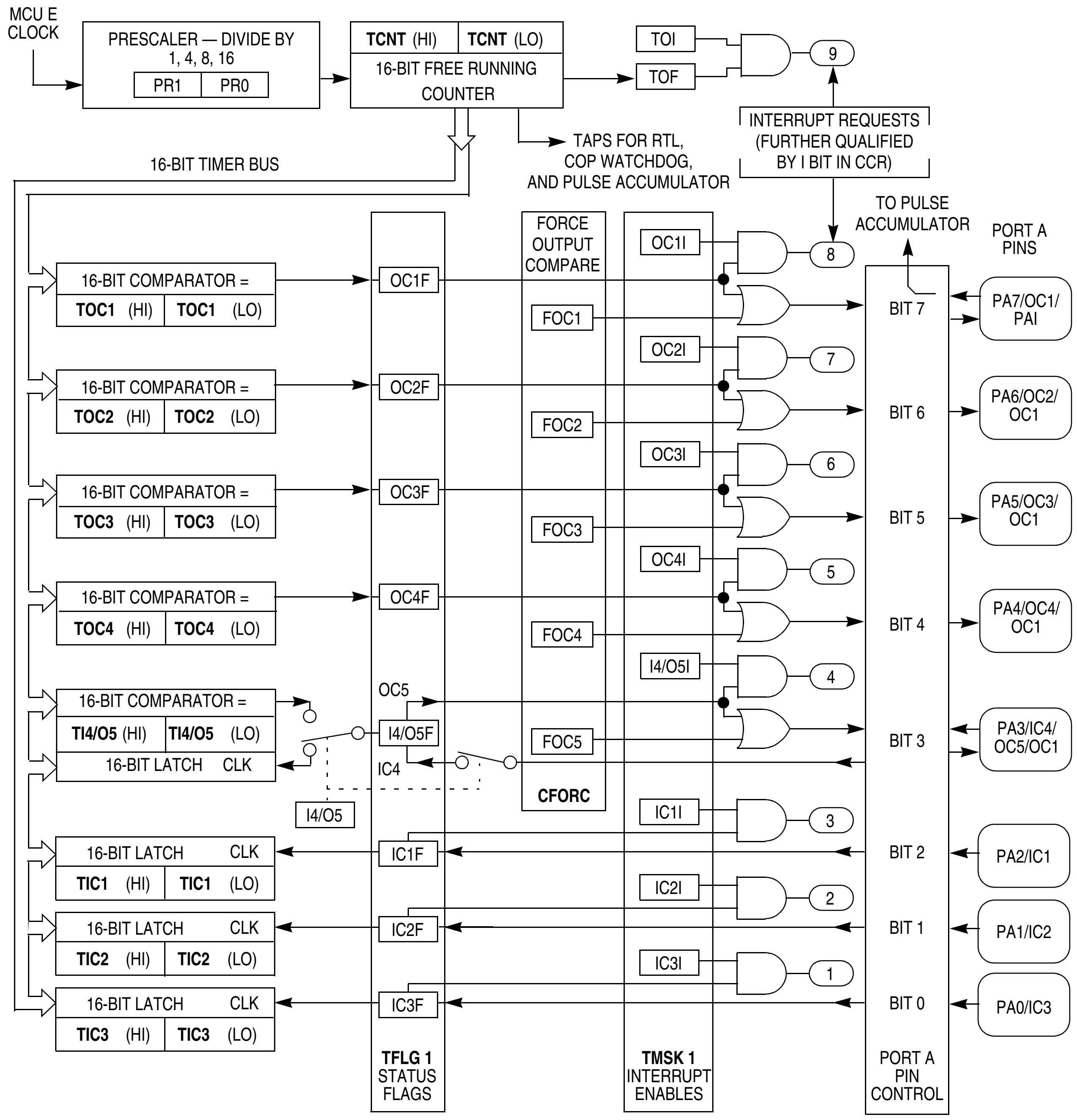


Figure 8-2. Capture/Compare Block Diagram

8.3 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 the PH2 clock. 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.

8.3.1 Timer Control 2 Register

Use the control bits of timer control 2 register (TCTL2) 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.

Address: $0021

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

Figure 8-3. Timer Control 2 Register (TCTL2)

EDGxB and EDGxA — Input Capture Edge Control

There are four pairs of these bits. Each pair is cleared to 0 by reset and must be encoded to configure the corresponding input capture edge detector circuit. IC4 functions only if the I4/O5 bit in PACTL is set. Refer to **Table 8-2** for timer control configuration.

Table 8-2. 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

8.3.2 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 two clock so that the count value is stable whenever a capture occurs. The timer input capture (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.

Address: $0010 — TIC1 (High)

	Bit 15	14	13	12	11	10	9	Bit 8
Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Write:	▨	▨	▨	▨	▨	▨	▨	▨
Reset:	Unaffected by reset							

Address: $0011 — TIC1 (Low)

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write:	▨	▨	▨	▨	▨	▨	▨	▨
Reset:	Unaffected by reset							

Address: $0012 — TIC2 (High)

	Bit 15	14	13	12	11	10	9	Bit 8
Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Write:	▨	▨	▨	▨	▨	▨	▨	▨
Reset:	Unaffected by reset							

▨ = Unimplemented

Figure 8-4. Timer Input Capture Registers (TICx)

Address: $0013 — TIC2 (Low)

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write:	▨	▨	▨	▨	▨	▨	▨	▨
Reset:	Unaffected by reset							

Address: $0014 — TIC3 (High)

	Bit 15	14	13	12	11	10	9	Bit 8
Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Write:	▨	▨	▨	▨	▨	▨	▨	▨
Reset:	Unaffected by reset							

Address: $0015 — TIC3 (Low)

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write:	▨	▨	▨	▨	▨	▨	▨	▨
Reset:	Unaffected by reset							

▨ = Unimplemented

Figure 8-4. Timer Input Capture Registers (TICx) (Continued)

8.3.3 Timer Input Capture 4/Output Compare 5 Register

Use timer input capture 4/output compare 5 (TI4/O5) as either an input capture register or an output compare register, depending on the function chosen for the I4/O5 pin. To enable it as an input capture pin, set the I4/O5 bit in the pulse accumulator control register (PACTL) to logic level 1. To use it as an output compare register, set the I4/O5 bit to a logic level 0. Refer to **8.7 Pulse Accumulator**.

Address: $001E — TI4/O5 (High)

	Bit 15	14	13	12	11	10	9	Bit 8
Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Write:	▨	▨	▨	▨	▨	▨	▨	▨
Reset:	1	1	1	1	1	1	1	1

Address: $001F — TI4/O5 (Low)

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

▨ = Unimplemented

Figure 8-5. Timer Input Capture 4/Output Compare 5 Register (TI4/O5)

8.4 Output Compare (OC)

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 to the output compare register a value representing 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 is 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 OC5–OC2, 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 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.

8.4.1 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.

Address: $0016 — TOC1 (High)

	Bit 15	14	13	12	11	10	9	Bit 8
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	1	1	1	1	1	1	1	1

Address: $0017 — TOC1 (Low)

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

Address: $0018 — TOC2 (High)

	Bit 15	14	13	12	11	10	9	Bit 8
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	1	1	1	1	1	1	1	1

Address: $0019 — TOC2 (Low)

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

Address: $001A — TOC3 (High)

	Bit 15	14	13	12	11	10	9	Bit 8
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	1	1	1	1	1	1	1	1

Figure 8-6. Timer Output Capture Registers (TOCx)

Address: $001B — TOC3 (Low)

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

Address: $001C — TOC4 (High)

	Bit 15	14	13	12	11	10	9	Bit 8
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	1	1	1	1	1	1	1	1

Address: $001D — TOC4 (Low)

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

Figure 8-6. Timer Output Capture Registers (TOCx) (Continued)

8.4.2 Timer Compare Force Register

The timer compare force register (CFORC) allows forced early compares. FOC1–FOC5 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.

Address: $000B

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0
Reset:	0	0	0	0	0	0	0	0

Figure 8-7. Timer Compare Force Register (CFORC)

FOC1–FOC5 — Write 1s to Force Compare Bits
 0 = Not affected
 1 = Output x action occurs

Bits 2–0 — Not implemented, always read 0.

8.4.3 Output Compare 1 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 PA7–PA3.

Address: $000C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0
Reset:	0	0	0	0	0	0	0	0

Figure 8-8. Output Compare 1 Mask Register (OC1M)

OC1M7–OC1M3 — Output Compare Masks
0 = OC1 disabled
1 = OC1 enabled to control the corresponding pin of port A

Bits 2–0 — Not implemented; always read 0.
Set bit(s) to enable OC1 to control corresponding pin(s) of port A.

8.4.4 Output Compare 1 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.

Address: $000D

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0
Reset:	0	0	0	0	0	0	0	0

Figure 8-9. Output Compare 1 Data Register (OC1D)

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 0.

8.4.5 Timer Counter Register

The 16-bit read-only timer count register (TCNT) 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.

Address: $000E — TCNT High

	Bit 15	14	13	12	11	10	9	Bit 8
Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 08
Write:	▨	▨	▨	▨	▨	▨	▨	▨
Reset:	0	0	0	0	0	0	0	0

Address: $000F — TCNT Low

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

▨ = Unimplemented

Figure 8-10. Timer Counter Registers (TCNT)

In normal modes, TCNT is read-only.

8.4.6 Timer Control 1 Register

The bits of the timer control 1 register (TCTL1) specify the action taken as a result of a successful OCx compare.

Address: $0020

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

Figure 8-11. Timer Control 1 Register (TCTL1)

OM2–OM5 — Output Mode Bits
OL2–OL5 — Output Level Bits

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 8-3** for the coding.

Table 8-3. 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 0
1	1	Set OCx output line to 1

8.4.7 Timer Interrupt Mask 1 Register

The timer interrupt mask 1 register (TMSK1) is an 8-bit register used to enable or inhibit the timer input capture and output compare interrupts.

Address: $0022

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

Figure 8-12. Timer Interrupt Mask 1 Register (TMSK1)

OC1I–OC4I — Output Compare x Interrupt Enable Bits
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 or Output Compare 5 Interrupt Enable Bit
When I4/O5 in PACTL is one, I4/O5I is the input capture 4 interrupt enable bit. When I4/O5 in PACTL is 0, I4/O5I is the output compare 5 interrupt enable bit.

IC1I–IC3I — Input Capture x Interrupt Enable Bits
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.*

8.4.8 Timer Interrupt Flag 1 Register

The timer interrupt flag 1 register (TFLG1) bits 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.

Address: $0023

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

Figure 8-13. Timer Interrupt Flag 1 Register (TFLG1)

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

OC1F–OC5F — 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

8.4.9 Timer Interrupt Mask 2 Register

The timer interrupt mask 1 register (TMSK2) is an 8-bit register used to enable or inhibit timer overflow and real-time interrupts. The timer prescaler control bits are included in this register.

Address: $0024

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

Figure 8-14. Timer Interrupt Mask 2 Register (TMSK2)

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

RTII — Real-Time Interrupt Enable Bit
  Refer to **8.5 Real-Time Interrupt**.

PAOVI — Pulse Accumulator Overflow Interrupt Enable Bit
  Refer to **8.7 Pulse Accumulator**.

PAII — Pulse Accumulator Input Edge Interrupt Enable Bit
Refer to **8.7 Pulse Accumulator**.

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

PR1 and PR0 — Timer Prescaler Select Bits
These bits are used to select the prescaler divide-by ratio. In normal modes, PR1 and PR0 can be written once only, and the write must be within 64 cycles after reset. Refer to **Table 8-4** for specific timing values.

Table 8-4. Timer Prescale

PR1 and PR0	Prescaler
0 0	1
0 1	4
1 0	8
1 1	16

8.4.10 Timer Interrupt Flag 2 Register

The timer interrupt flag 2 register (TFLG2) bits 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.

Address: $0025

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	TOF	RTIF	PAOVF	PAIF	0	0	0	0
Reset:	0	0	0	0	0	0	0	0

Figure 8-15. Timer Interrupt Flag 2 Register (TFLG2)

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

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

RTIF — Real-Time (Periodic) Interrupt Flag
Refer to **8.5 Real-Time Interrupt**.

PAOVF — Pulse Accumulator Overflow Interrupt Flag
Refer to **8.7 Pulse Accumulator**.

PAIF — Pulse Accumulator Input Edge Interrupt Flag
Refer to **8.7 Pulse Accumulator**.

Bits 3–0 — Not implemented
Always read 0.

8.5 Real-Time Interrupt

The real-time interrupt 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 RTR1 and RTR0. Refer to **Table 8-5** for the periodic real-time interrupt rates.

Table 8-5. Periodic Real-Time Interrupt Rates

RTR1 and RTR0	E = 1 MHz	E = 2 MHz	E = 3 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 real-time interrupt period elapses before the RTIF flag is set for the first time. Refer to the TMSK2, TFLG2, and PACTL registers.

8.5.1 Timer Interrupt Mask 2 Register

The timer interrupt mask 2 register (TMSK2) contains the real-time interrupt enable bits.

Address: $0024

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

Figure 8-16. Timer Interrupt Mask 2 Register (TMSK2)

TOI — Timer Overflow Interrupt Enable Bit
Refer to **8.4 Output Compare (OC)**.

RTII — Real-Time Interrupt Enable Bit
0 = RTIF interrupts disabled
1 = Interrupt requested

PAOVI — Pulse Accumulator Overflow Interrupt Enable Bit
Refer to **8.7 Pulse Accumulator**.

PAII — Pulse Accumulator Input Edge Bit
Refer to **8.7 Pulse Accumulator**.

Bits 3–2 — Unimplemented
Always read 0.

PR1 and PR0 — Timer Prescaler Select Bits
Refer to **Table 8-4**.

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

8.5.2 Timer Interrupt Flag 2 Register

Bits of the timer interrupt flag 2 register (TFLG2) 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.

Address: $0025

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	TOF	RTIF	PAOVF	PAIF	0	0	0	0
Reset:	0	0	0	0	0	0	0	0

Figure 8-17. Timer Interrupt Flag 2 Register (TFLG2)

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

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

RTIF — Real-Time Interrupt Flag
The RTIF status bit is automatically set to 1 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 **8.7 Pulse Accumulator**.

PAIF — Pulse Accumulator Input Edge Interrupt Flag
Refer to **8.7 Pulse Accumulator**.

Bits 3–0 — Not implemented
Always read 0.

8.5.3 Pulse Accumulator Control Register

Bits RTR1 and RTR0 of the pulse accumulator control register (PACTL) select the rate for the real-time interrupt system. Bit DDRA3 determines whether port A bit three is an input or an output when used for general-purpose I/O. The remaining bits control the pulse accumulator.

Address: $0026

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

Figure 8-18. Pulse Accumulator Control Register (PACTL)

DDRA7 — Data Direction Control for Port A Bit 7
Refer to **8.7 Pulse Accumulator**.

PAEN — Pulse Accumulator System Enable Bit
Refer to **8.7 Pulse Accumulator**.

PAMOD — Pulse Accumulator Mode Bit
Refer to **8.7 Pulse Accumulator**.

PEDGE — Pulse Accumulator Edge Control Bit
Refer to **8.7 Pulse Accumulator**.

DDRA3 — Data Direction Register for Port A Bit 3
Refer to **Section 5. Input/Output (I/O) Ports**.

I4/O5 — Input Capture 4/Output Compare 5 Bit
Refer to **8.3 Input Capture**.

RTR1 and RTR0 — RTI Interrupt Rate Select Bits
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. See **Table 8-6**.

Table 8-6. Real-Time Interrupt Rates

RTR1 and RTR0	E = 1 MHz	E = 2 MHz	E = 3 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	$(E/2^{16})$

8.6 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 CR1 and CR0 bits in the OPTION register and the NOCOP bit in the CONFIG register determine the status of the COP function. Refer to **Section 4. Resets, Interrupts, and Low-Power Modes** for a more detailed discussion of the COP function.

8.7 Pulse Accumulator

The MC68HC711D3 has 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 8-19**.

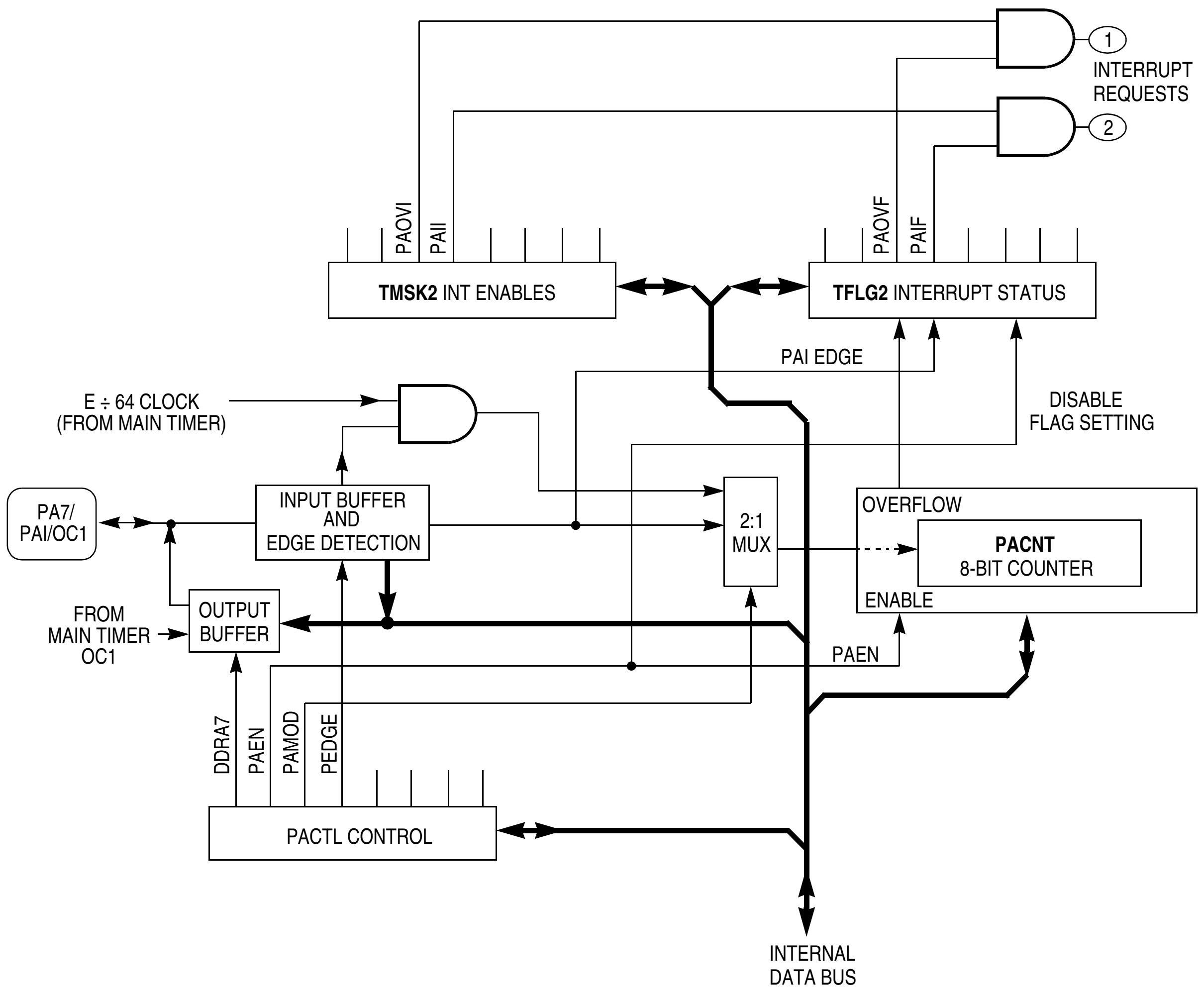


Figure 8-19. Pulse Accumulator

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 8-7**. The pulse accumulator counter can be read or written at any time.

Pulse accumulator control bits are also located within two timer registers, TMSK2 and TFLG2, as described here.

Table 8-7. Pulse Accumulator Timing in Gated Mode

	Selected Crystal	**Common XTAL Frequencies**		
		4.0 MHz	**8.0 MHz**	**12.0 MHz**
CPU Clock	**(E)**	**1.0 MHz**	**2.0 MHz**	**3.0 MHz**
Cycle Time	**(1/E)**	**1000 ns**	**500 ns**	**333 ns**
($E/2^6$)	1 count -	64.0 µs	32.0 µs	21.33 µs
($E/2^{14}$)	overflow -	16.384 ms	8.192 ms	5.461 ms

8.7.1 Pulse Accumulator Control Register

Four of the pulse accumulator control register (PACTL) 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.

Address: $0026

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

Figure 8-20. Pulse Accumulator Control Register (PACTL)

DDRA7 — Data Direction Control for Port A Bit 7
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: *Even when port A bit 7 is configured as an output, the pin still drives the input to the pulse accumulator.*

Refer to **Section 5. Input/Output (I/O) Ports** for more information.

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

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

PEDGE — Pulse Accumulator Edge Control Bit

This bit has different meanings depending on the state of the PAMOD bit, as shown in **Table 8-8**.

Table 8-8. 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 0 on PAI inhibits counting.
1	1	A 1 on PAI inhibits counting.

DDRA3 — Data Direction Register for Port A Bit 3

Refer to **Section 5. Input/Output (I/O) Ports**.

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

Refer to **8.3 Input Capture**.

RTR1 and RTR0 — RTI Interrupt Rate Select Bits

Refer to **8.5 Real-Time Interrupt**.

8.7.2 Pulse Accumulator Count Register

The 8-bit read/write pulse accumulator count register (PACNT) contains the count of external input events at the PAI input or the accumulated count. 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.

Address: $0027

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset:	Unaffected by reset							

Figure 8-21. Pulse Accumulator Count Register (PACNT)

8.7.3 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.

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 1 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 0, pulse accumulator overflow interrupts are inhibited, and the system operates in a polled mode, which requires PAOVF to 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 1 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 0, 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 TFLG register.

Address: $0024

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

Figure 8-22. Timer Interrupt Mask 2 Register (TMSK2)

Address: $0025

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	TOF	RTIF	PAOVF	PAIF	0	0	0	0
Reset:	0	0	0	0	0	0	0	0

Figure 8-23. Timer Interrupt Flag 2 Register (TFLG2)

Section 9. Electrical Characteristics

9.1 Introduction

This section contains electrical specifications.

9.2 Maximum Ratings

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

NOTE: *This device is not guaranteed to operate properly at the maximum ratings. Refer to **9.5 DC Electrical Characteristics** for guaranteed operating conditions.*

Rating	Symbol	Value	Unit
Supply voltage	V_{DD}	–0.3 to +7.0	V
Input voltage	V_{In}	–0.3 to +7.0	V
Current drain per pin(1) Excluding V_{DD}, V_{SS}, V_{RH}, and V_{RL}	I_D	25	mA
Storage temperature	T_{STG}	–55 to +150	°C

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

NOTE: *This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{In} and V_{Out} be constrained to the range $V_{SS} \le (V_{In}$ or $V_{Out}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (for example, either V_{SS} or V_{DD}).*

9.3 Functional Operating Temperature Range

Rating	Symbol	Value	Unit
Operating temperature range MC68HC711D3 MC68HC711D3V	T_A	T_L to T_H –40 to +85 –40 to +105	°C

9.4 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) 40-pin plastic dual in-line package (DIP) 44-pin plastic leaded chip carrier (PLCC) 44-pin plastic quad flat pack (QFP)	Θ_{JA}	 50 50 85	°C/W
Total power dissipation(1)	P_D	$P_{INT} + P_{I/O}$ $K / T_J + 273°C$	W
Device internal power dissipation	P_{INT}	$I_{DD} \times V_{DD}$	W
I/O pin power dissipation(2)	$P_{I/O}$	User-determined	W
A constant(3)	K	$P_D \times (T_A + 273°C)$ $+ \Theta_{JA} \times P_D^2$	W/°C

1. This is an approximate value, neglecting $P_{I/O}$.
2. For most applications, $P_{I/O} \leq P_{INT}$ and can be neglected.
3. K is a constant pertaining to the device. Solve for K with a known T_A and a measured P_D (at equilibrium). Use this value of K to solve for P_D and T_J, iteratively, for any value of T_A.

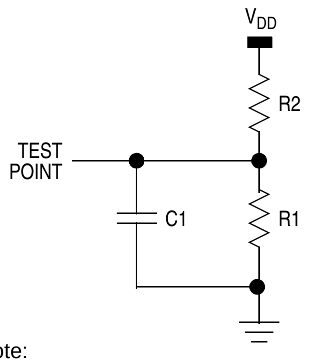
9.5 DC Electrical Characteristics

Characteristic[1]	Symbol	Min	Max	Unit
Output voltage[2] All outputs I_{Load} = ± 10.0 µAAll outputs except $\overline{\text{RESET}}$ and MODA	V_{OL} V_{OH}	— V_{DD} – 0.1	0.1 —	V
Output high voltage[1] All outputs except I_{Load} = – 0.8 mA, V_{DD} = 4.5 V $\overline{\text{RESET}}$, EXTAL, and MODA	V_{OH}	V_{DD} – 0.8	—	V
Output low voltage All outputs except XTAL I_{Load} = 1.6 mA	V_{OL}	—	0.4	V
Input high voltage All inputs except $\overline{\text{RESET}}$ $\overline{\text{RESET}}$	V_{IH}	0.7 x V_{DD} 0.8 x V_{DD}	V_{DD} + 0.3 V_{DD} + 0.3	V
Input low voltage All inputs	V_{IL}	V_{SS} – 0.3	0.2 x V_{DD}	V
I/O ports, three-state leakage PA7, PA3, PC7–PC0, PD7–PD0, V_{In} = V_{IH} or V_{IL} MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$	I_{OZ}	—	±10	µA
Input leakage current V_{In} = V_{DD} or V_{SS} $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$ V_{In} = V_{DD} or V_{SS} MODB/V_{STBY}	I_{In}	 — —	 ±1 ±10	µA
RAM standby voltagePower down	V_{SB}	4.0	V_{DD}	V
RAM standby currentPower down	I_{SB}	—	20	µA
Total supply current[3] RUN: Single-chip mode dc — 2 MHz 3 MHz Expanded multiplexed mode dc — 2 MHz 3 MHz	I_{DD}	 — — — —	 15 27 27 35	mA
WAIT — All peripheral functions shut down: Single-chip mode dc — 2 MHz 3 MHz Expanded multiplexed mode dc — 2 MHz 3 MHz	W_{IDD}	 — — — —	 6 15 10 20	mA
STOP — No clocks, single-chip mode: dc — 2 MHz 3 MHz	S_{IDD}	 — —	 100 150	µA

The dc electrical table continues on next page.

Characteristic[1]	Symbol	Min	Max	Unit
Input capacitancePA3–PA0, $\overline{IRQ}$, $\overline{XIRQ}$, EXTAL PA7, PC7–PC0, PD7–PD0, MODA/$\overline{LIR}$, $\overline{RESET}$	C_{In}	— —	8 12	pF
Power dissipation Single-chip mode dc — 2 MHz 3 MHz Expanded multiplexed mode dc — 2 MHz 3 MHz	P_D	 — — — —	 85 150 150 195	mW
EPROM programming voltage	V_{PP}	11.75	12.75	V
EPROM programming time	t_{PP}	2	4	ms

1. V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted.
2. V_{OH} specification for $\overline{RESET}$ and MODA is not applicable because they are open-drain pins. V_{OH} specification is not applicable to ports C and D in wired-OR mode.
3. All ports configured as inputs: $V_{IL} \leq 0.2$ V, $V_{IH} \leq V_{DD}$ –0.2 V; no dc loads; EXTAL is driven with a square wave; t_{cyc} = 476.5 ns.

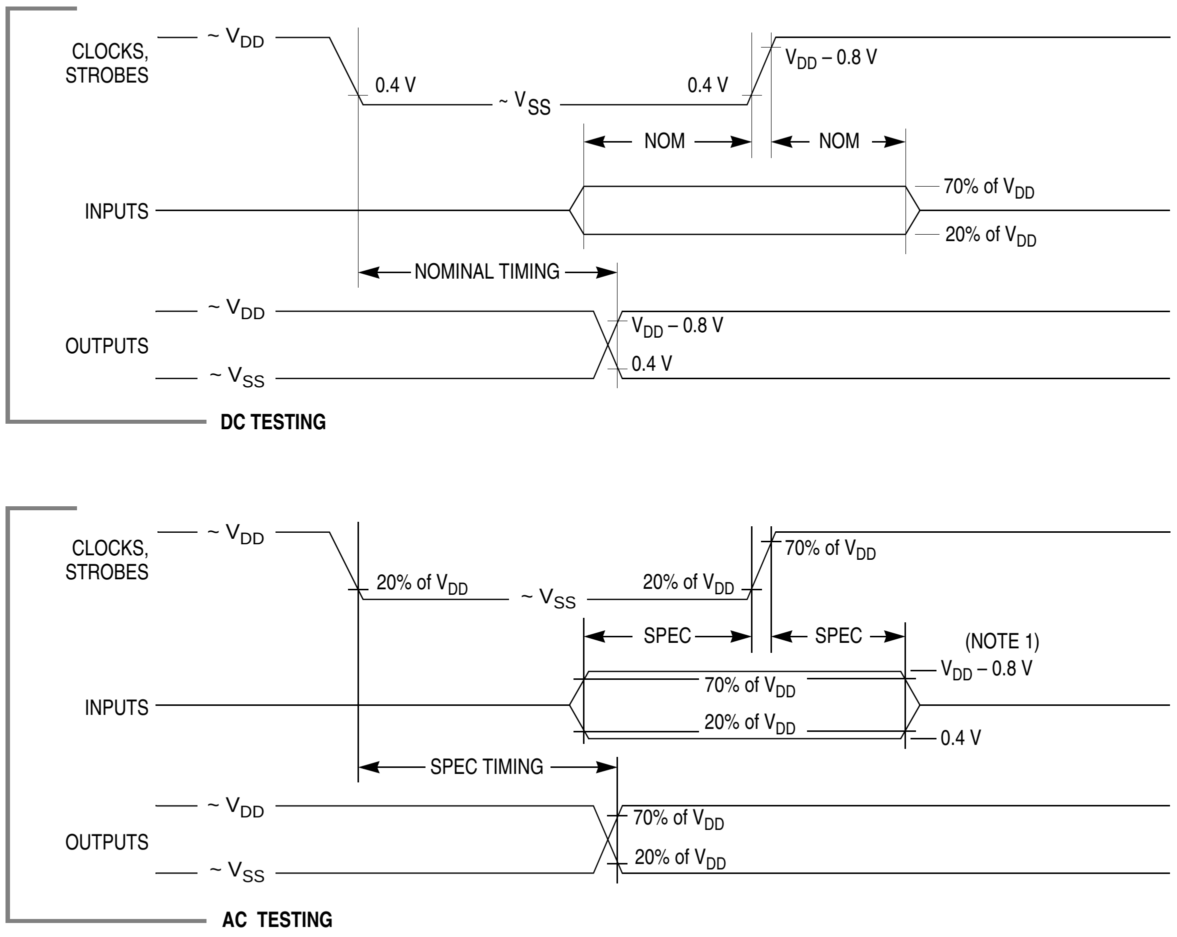


EQUIVALENT TEST LOAD[1]

Pins	R1	R2	C1
PA3–PA7 PB0–PB7 PC0–PC7 PD0, PD5–PD7 E	3.26 K	2.38 K	90 pF
PD1—PD4	3.26 K	2.38 K	200 pF

Note:
1. Full test loads are applied during all ac electrical timing measurements.

Figure 9-1. Equivalent Test Load



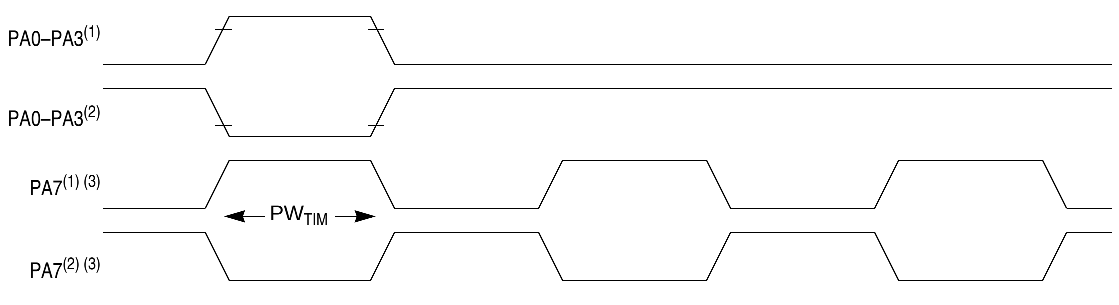
Note:
1. During ac timing measurements, inputs are driven to 0.4 volts and V_{DD} – 0.8 volts while timing measurements are taken at the 20% and 70% of V_{DD} points.

Figure 9-2. Test Methods

9.6 Control Timing

Characteristic[1]	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 timet$_{PCSU}$ = 1/4 t_{cyc} + 50 ns	t_{PCSU}	300	—	175	—	133	—	ns
Reset input pulse width[2] To guarantee external reset vector Minimum input time can be preempted by internal reset	PW_{RSTL}	8 1	— —	8 1	— —	8 1	— —	t_{cyc}
Mode programming setup time	t_{MPS}	2	—	2	—	2	—	t_{cyc}
Mode programming hold time	t_{MPH}	10	—	10	—	10	—	ns
Interrupt pulse width, $PW_{IRQ} = t_{cyc} + 20$ ns $\overline{IRQ}$ edge-sensitive mode	PW_{IRQ}	1020	—	520	—	353	—	ns
Wait recovery startup time	t_{WRS}	—	4	—	4	—	4	t_{cyc}
Timer pulse width $PW_{TIM} = t_{cyc} + 20$ ns Input capture pulse Accumulator input	PW_{TIM}	1020	—	520	—	353	—	ns

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. 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. Input/Output (I/O) Ports** for further details.



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

Figure 9-3. Timer Inputs

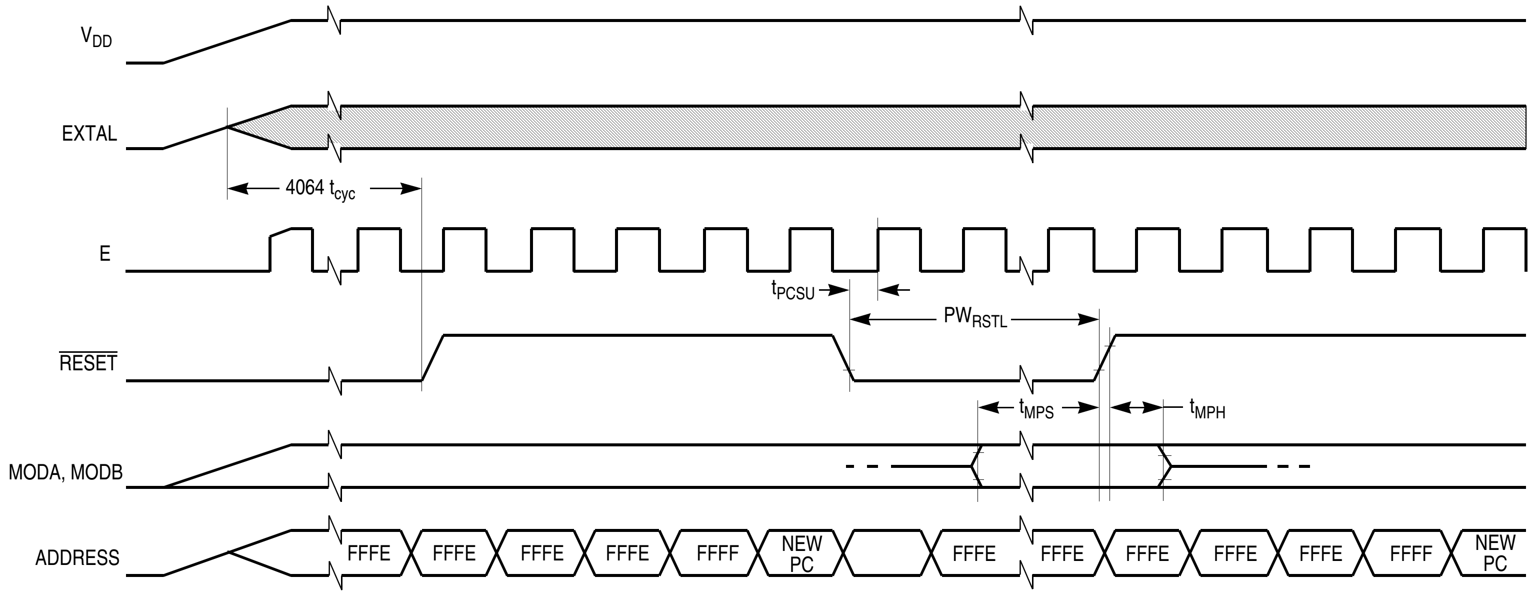
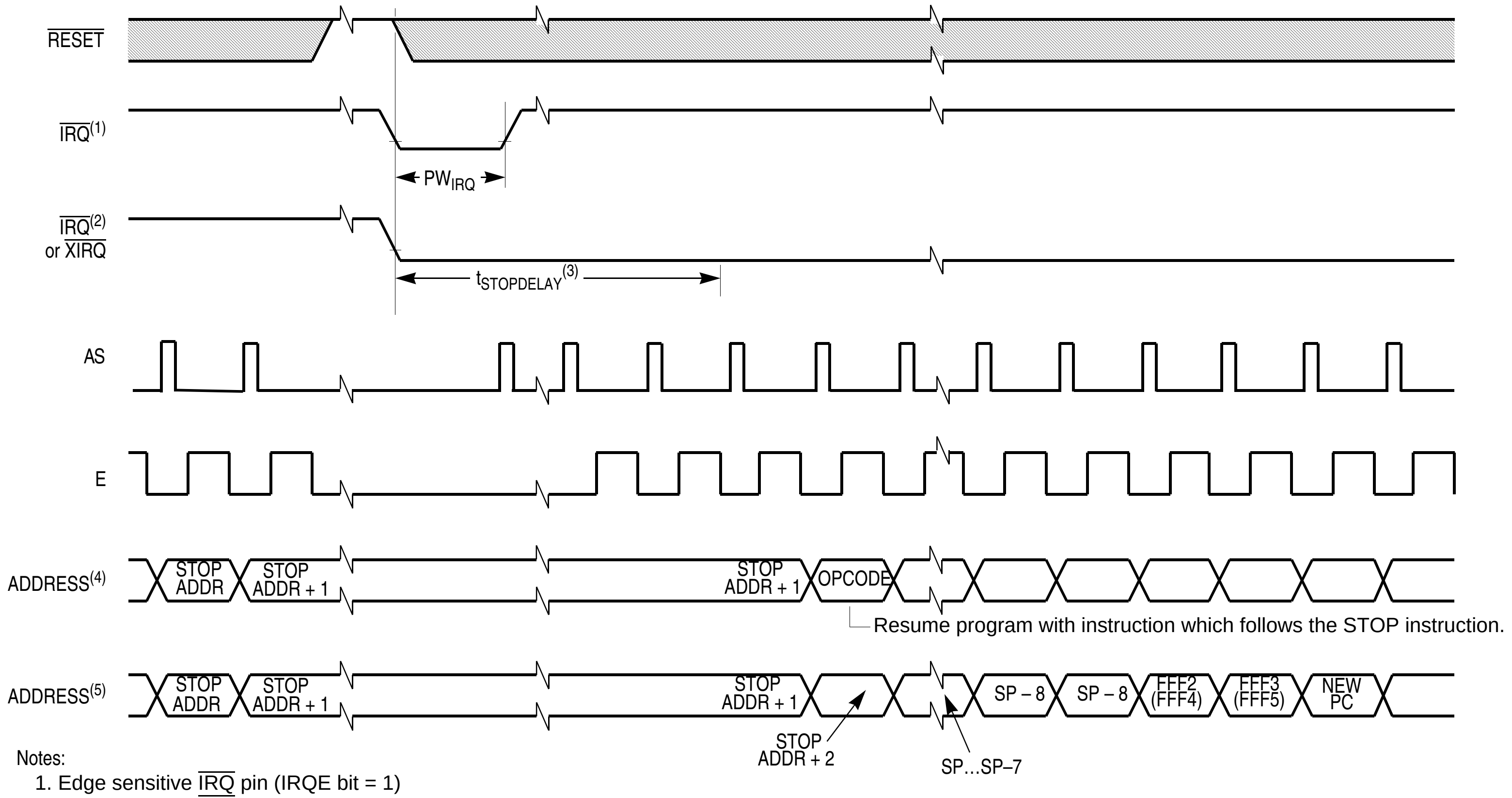
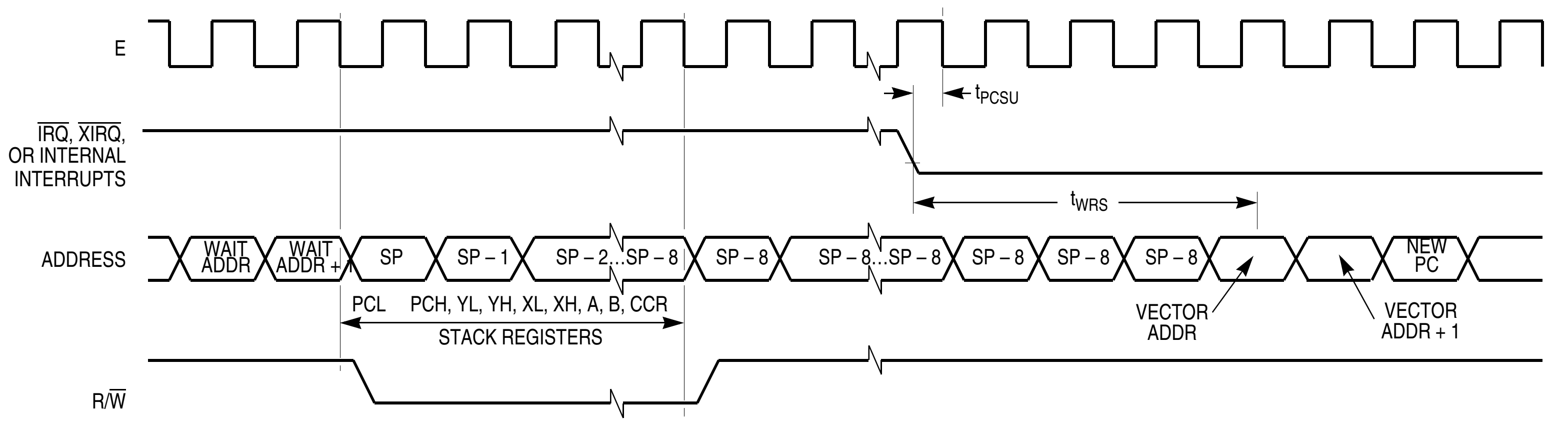


Figure 9-4. POR and External Reset Timing Diagram



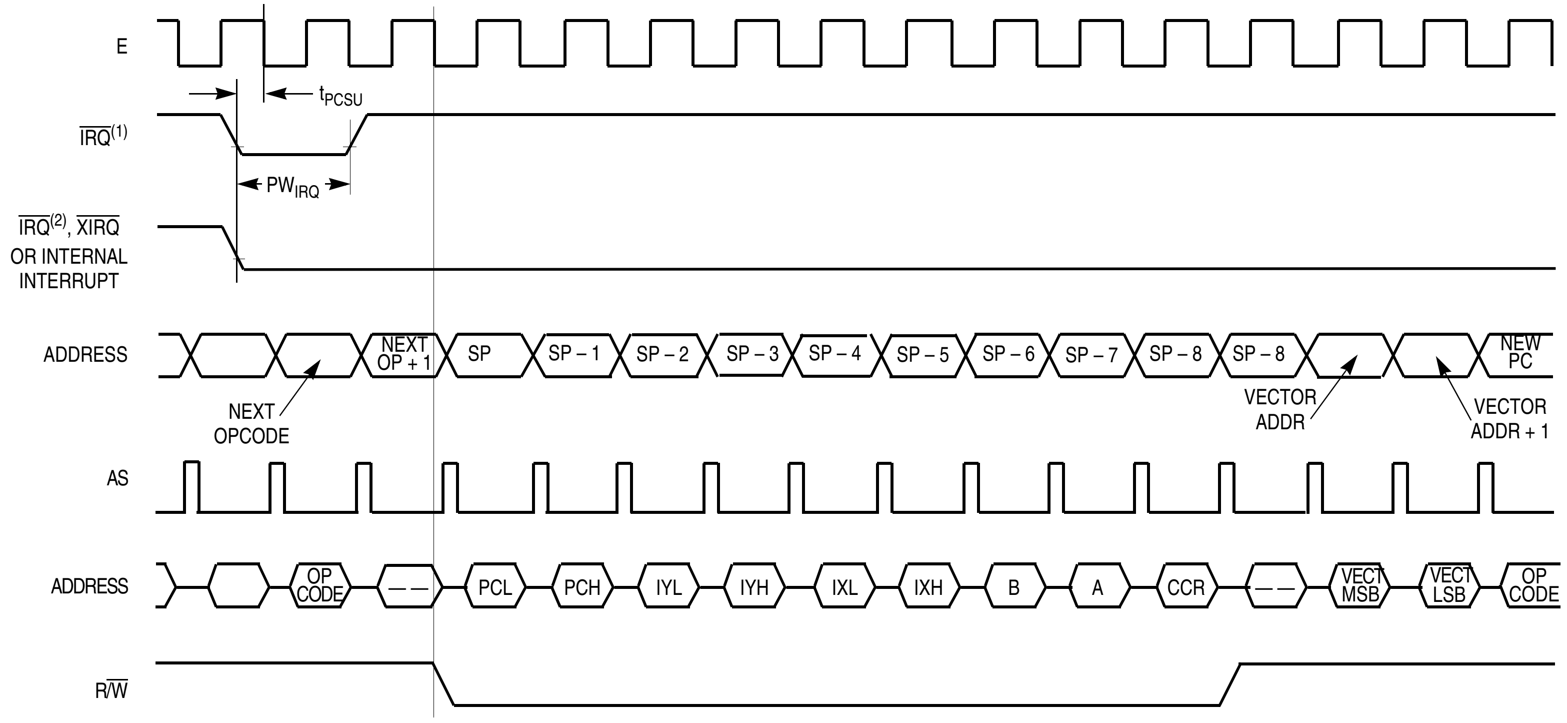
Notes:
1. Edge sensitive $\overline{\text{IRQ}}$ pin (IRQE bit = 1)
2. Edge sensitive $\overline{\text{IRQ}}$ pin (IRQE bit = 0)
3. $t_{STOPDELAY}$ = 4064 t_{cyc} if DLY bit = 1 or 4 t_{cyc} if DLY = 0.
4. $\overline{\text{XIRQ}}$ with X bit in CCR = 1.
5. $\overline{\text{IRQ}}$ or ($\overline{\text{XIRQ}}$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

Figure 9-5. STOP Recovery Timing Diagram



Note: $\overline{\text{RESET}}$ also causes recovery from WAIT.

Figure 9-6. WAIT Recovery Timing Diagram



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

Figure 9-7. Interrupt Timing Diagram

9.7 Peripheral Port Timing

Characteristic[1]	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	1.0	1.0	2.0	2.0	3.0	3.0	MHz
E-clock period	t_{CYC}	1000	—	500	—	333	—	ns
Peripheral data setup time[2] MCU read of ports A, B, C, and D	t_{PDSU}	100	—	100	—	100	—	ns
Peripheral data hold time[2] MCU read of ports A, B, C, and D	t_{PDH}	50	—	50	—	50	—	ns
Delay time, peripheral data write MCU write to port A	t_{PWD}	—	200	—	200	—	200	ns
MCU writes to ports B, C, and D t_{PWD} = 1/4 t_{cyc} + 150 ns	t_{PWD}	—	350	—	225	—	183	ns

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. Port C and D timing is valid for active drive (CWOM and DWOM bits not set in PIOC and SPCR registers respectively).

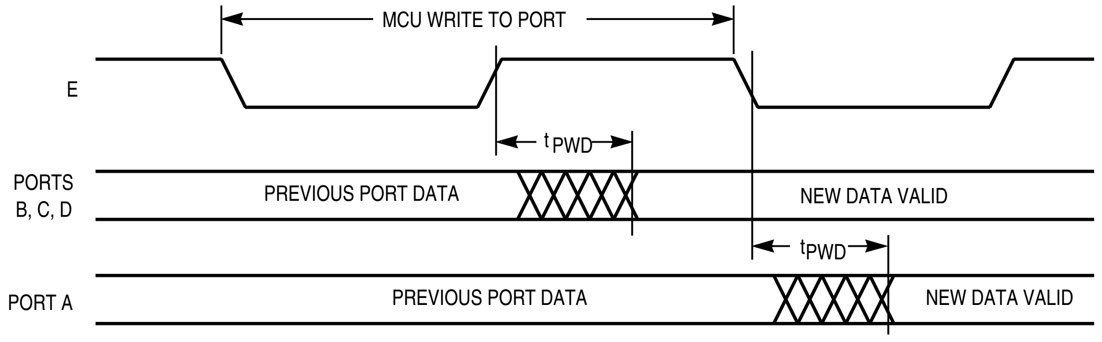


Figure 9-8. Port Write Timing Diagram

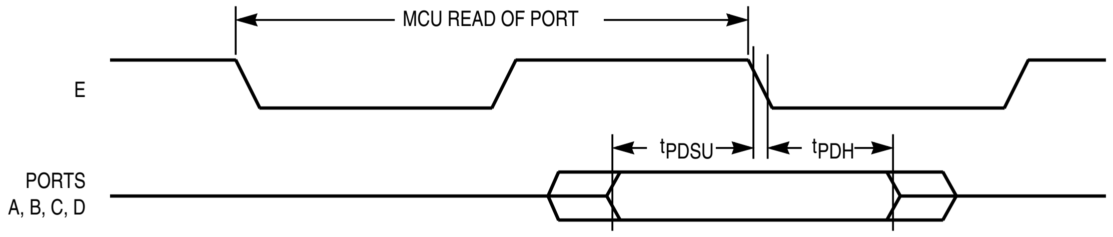
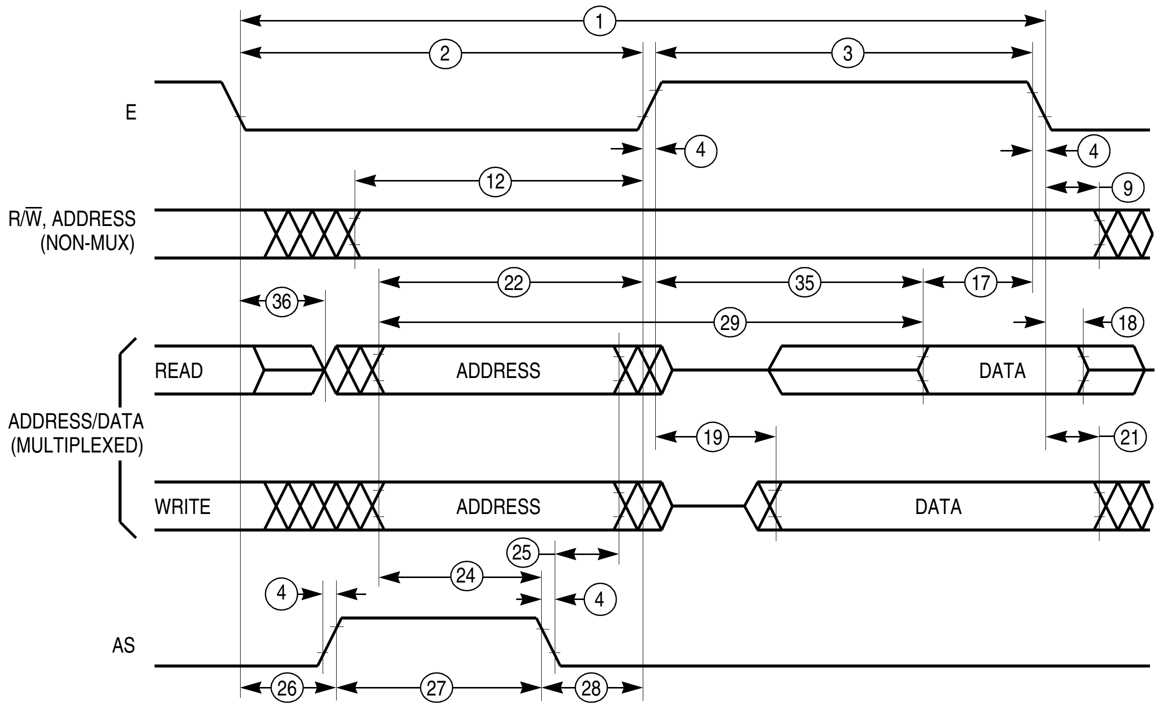


Figure 9-9. Port Read Timing Diagram

9.8 Expansion Bus Timing

Num	Characteristic(1)	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	PW_{EL}	477	—	227	—	146	—	ns
3	Pulse width, E high, PW_{EH} = 1/2 t_{cyc} – 28 ns	PW_{EH}	472	—	222	—	141	—	ns
4A	E and AS rise time	t_r	—	20	—	20	—	20	ns
4B	E and AS fall time	t_f	—	20	—	20	—	15	ns
9	Address hold time(2)a, t_{AH} = 1/8 t_{cyc} – 29.5 ns	t_{AH}	95.5	—	33	—	26	—	ns
12	Non-muxed address valid time to E rise t_{AV} = PW_{EL} – (t_{ASD} + 80 ns)(2)a	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(2)a	t_{DDW}	—	190.5	—	128	—	71	ns
21	Write data hold time, t_{DHW} = 1/8 t_{cyc} – 29.5 ns(2)a	t_{DHW}	95.5	—	33	—	26	—	ns
22	Muxed address valid time to E rise t_{AVM} = PW_{EL} – (t_{ASD} + 90 ns)(2)a	t_{AVM}	271.5	—	84	—	54	—	ns
24	Muxed address valid time to AS fall t_{ASL} = PW_{ASH} – 70 ns	t_{ASL}	151	—	26	—	13	—	ns
25	Muxed address hold time, t_{AHL} = 1/8 t_{cyc} – 29.5 ns(2)b	t_{AHL}	95.5	—	33	—	31	—	ns
26	Delay time, E to AS rise, t_{ASD} = 1/8 t_{cyc} – 9.5 ns(2)a	t_{ASD}	115.5	—	53	—	31	—	ns
27	Pulse width, AS high, PW_{ASH} = 1/4 t_{cyc} – 29 ns	PW_{ASH}	221	—	96	—	63	—	ns
28	Delay time, AS to E rise, t_{ASED} = 1/8 t_{cyc} – 9.5 ns(2)b	t_{ASED}	115.5	—	53	—	31	—	ns
29	MPU address access time(2)a 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	Muxed address delay (previous cycle MPU read) t_{MAD} = t_{ASD} + 30 ns(2)a(3)	t_{MAD}	145.5	—	83	—	51	—	ns

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
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. Formula only for dc to 2 MHz.



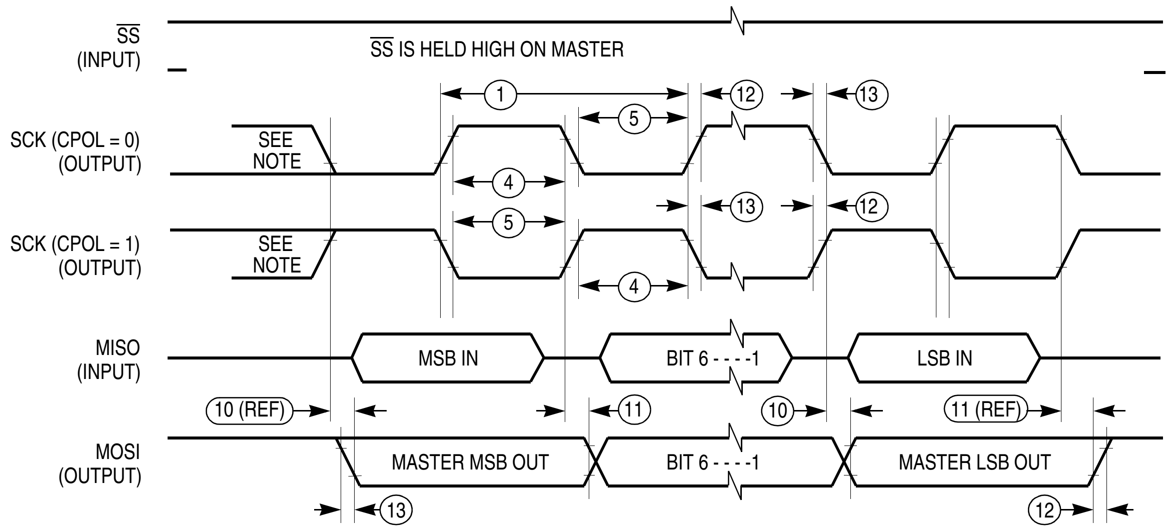
Note: Measurement points shown are 20% and 70% of V_{DD}.

Figure 9-10. Multiplexed Expansion Bus Timing Diagram

9.9 Serial Peripheral Interface Timing

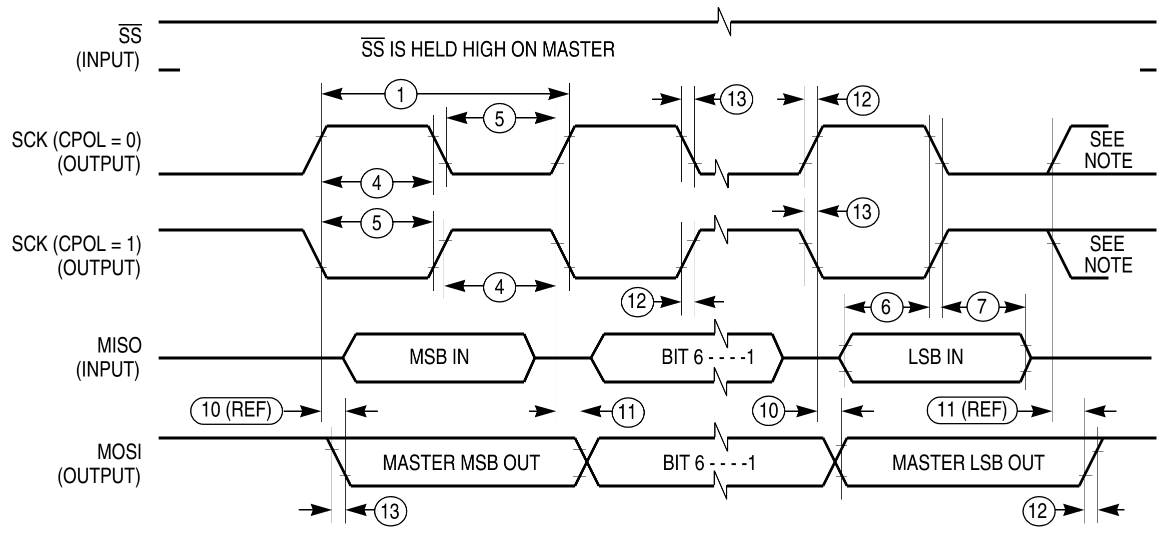
Num	Characteristic[1]	Symbol	2.0 MHz		3.0 MHz		Unit
			Min	Max	Min	Max	
	Operating frequency Master Slave	$f_{op(m)}$ $f_{op(s)}$	dc dc	0.5 2.0	dc dc	0.5 3.0	f_{op} MHz
1	Cycle time Master Slave	$t_{cyc(m)}$ $t_{CYC(s)}$	2.0 500	— —	2.0 333	— —	t_{cyc} ns
2	Enable lead time Master[2] Slave	$t_{lead(m)}$ $t_{lead(s)}$	— 250	— —	— 240	— —	ns
3	Enable lag time Master[2] Slave	$t_{lag(m)}$ $t_{lag(s)}$	— 250	— —	— 240	— —	ns
4	Clock (SCK) high time Master Slave	$t_{w(SCKH)m}$ $t_{w(SCKH)s}$	340 190	— —	227 127	— —	ns
5	Clock (SCK) low time Master Slave	$t_{w(SCKL)m}$ $t_{w(SCKL)s}$	340 190	— —	227 127	— —	ns
6	Data setup time (inputs) Master Slave	$t_{su(m)}$ $t_{su(s)}$	100 100	— —	100 100	— —	ns
7	Data hold time (inputs) Master Slave	$t_{h(m)}$ $t_{h(s)}$	100 100	— —	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)[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) SPI inputs (SCK, MOSI, MISO, and $\overline{SS}$)	t_{rm} t_{rs}	— —	100 2.0	— —	100 2.0	ns µs
13	Fall time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI outputs (SCK, MOSI, and MISO) SPI inputs (SCK, MOSI, MISO, and $\overline{SS}$)	t_{fm} t_{fs}	— —	100 2.0	— —	100 2.0	ns µs

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. Signal production depends on software.
3. Assumes 200 pF load on all SPI pins.



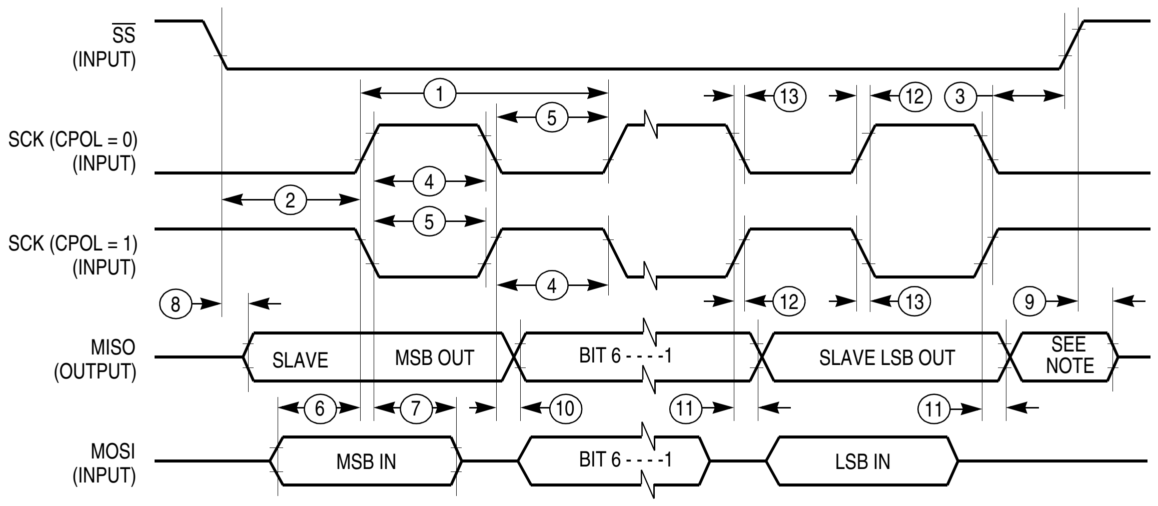
Note: This first clock edge is generated internally but is not seen at the SCK pin.

Figure 9-11. SPI Master Timing (CPHA = 0)



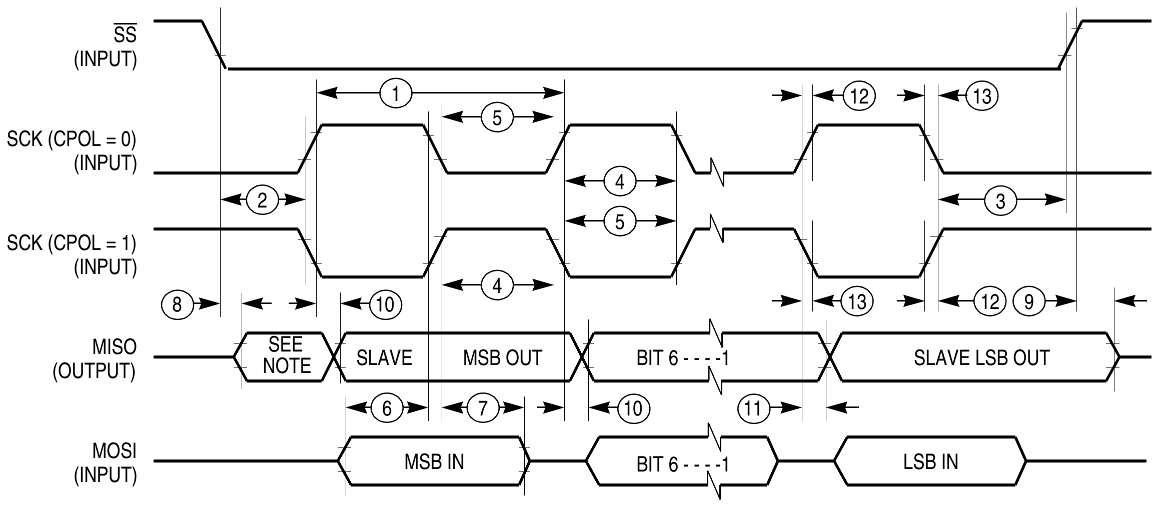
Note: This last clock edge is generated internally but is not seen at the SCK pin.

Figure 9-12. SPI Master Timing (CPHA = 1)



Note: Not defined but normally MSB of character just received

Figure 9-13. SPI Slave Timing (CPHA = 0)



Note: Not defined but normally LSB of character previously transmitted

Figure 9-14. SPI Slave Timing (CPHA = 1)

Section 10. Ordering Information and Mechanical Specifications

10.1 Introduction

This section provides ordering information for the MC68HC711D3. In addition, mechanical specifications are provided for the following packaging options:

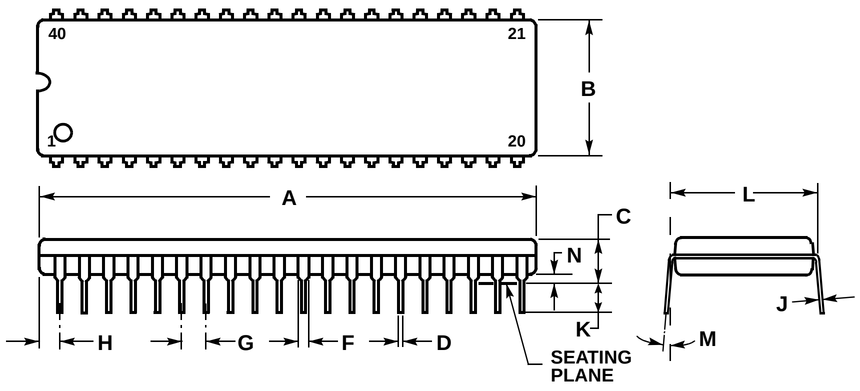
- 40-pin plastic dual in-line package (DIP)
- 44-pin plastic leaded chip carrier (PLCC)
- 44-pin plastic quad flat pack (QFP)

10.2 Ordering Information

Table 10-1. MC Order Numbers

Package Type	Temperature	MC Order Number	
		2 MHz	3 MHz
40-pin DIP	–40 to +85°C	MC68HC711D3CP2	MC68HC711D3CP3
44-pin PLCC	–40 to +85°C	MC68HC711D3CFN2	MC68HC711D3CFN3
	–40 to +105°C	MC68HC711D3VFN2	MC68HC711D3VFN3
44-pin QFP	–40 to +85°C	MC68HC711D3CFB2	MC68HC711D3CFB3

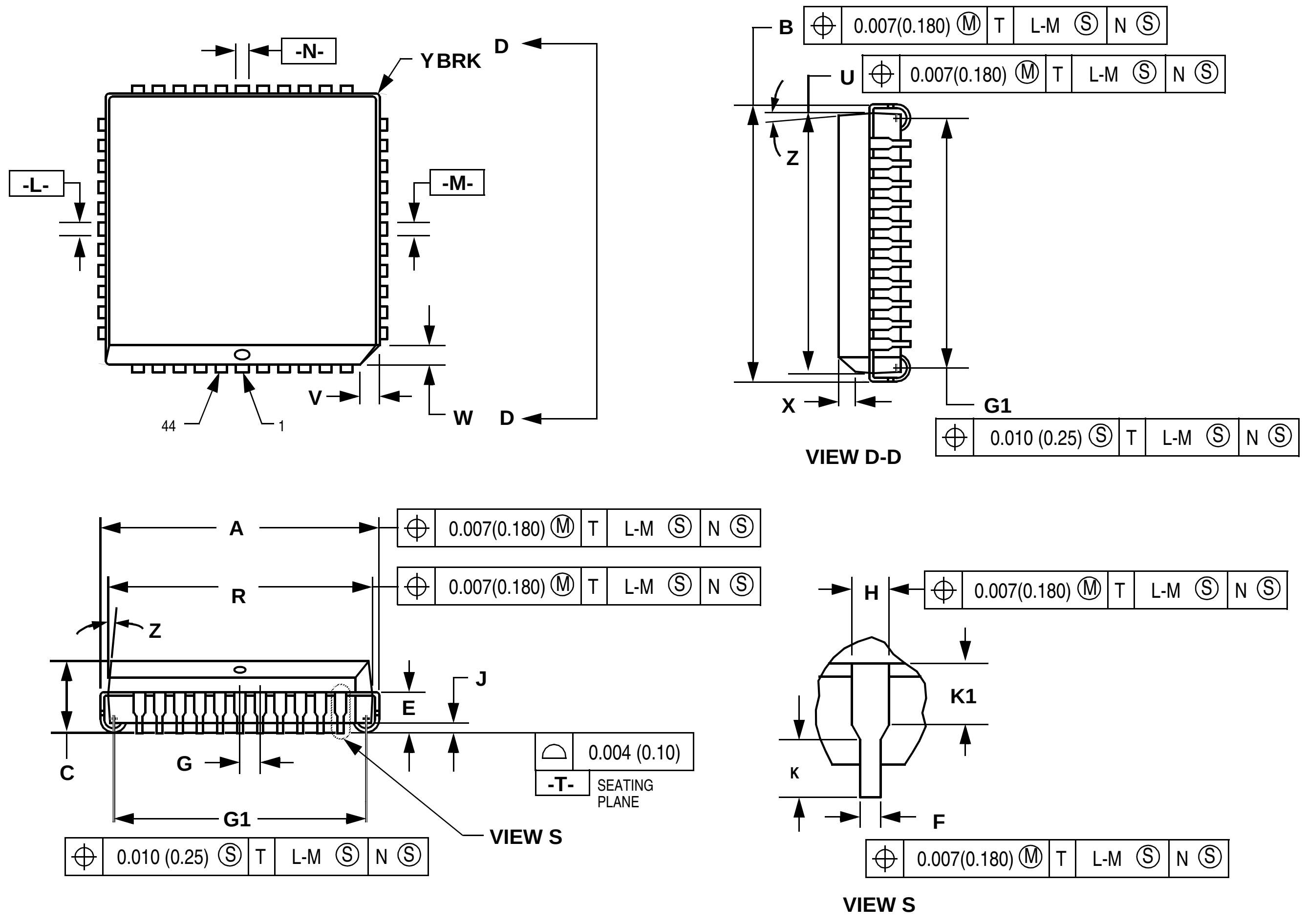
10.3 40-Pin DIP (Case 711-03)



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

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	51.69	52.45	2.035	2.065
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

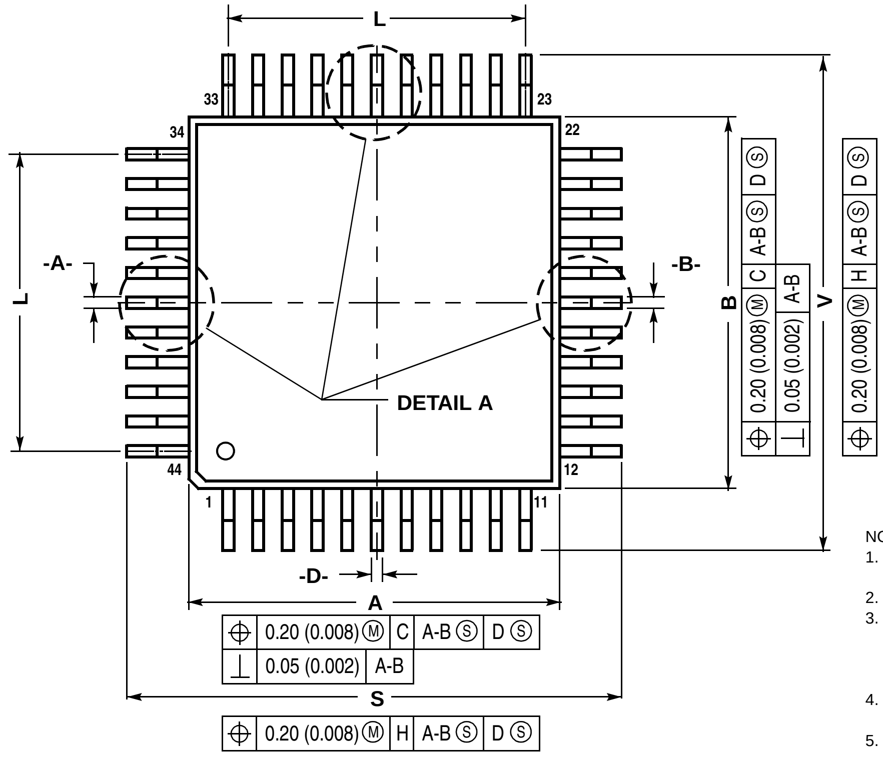
10.4 44-Pin PLCC (Case 777-02)

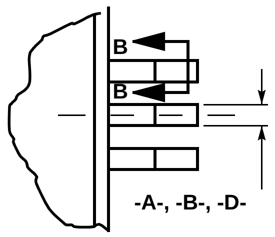

-N-
Y BRK
D
-L-
-M-
V
W
D
44
1
B
0.007(0.180) Ⓜ T L-M Ⓢ N Ⓢ
U
0.007(0.180) Ⓜ T L-M Ⓢ N Ⓢ
Z
X
G1
0.010 (0.25) Ⓢ T L-M Ⓢ N Ⓢ
VIEW D-D
A
0.007(0.180) Ⓜ T L-M Ⓢ N Ⓢ
R
0.007(0.180) Ⓜ T L-M Ⓢ N Ⓢ
Z
J
E
C
G
0.004 (0.10)
-T- SEATING PLANE
G1
0.010 (0.25) Ⓢ T L-M Ⓢ N Ⓢ
VIEW S
H
0.007(0.180) Ⓜ T L-M Ⓢ N Ⓢ
K1
K
F
0.007(0.180) Ⓜ T L-M Ⓢ N Ⓢ
VIEW S

NOTES:
1. DATUMS -L-, -M-, AND -N- ARE DETERMINED WHERE TOP OF LEAD SHOLDERS EXITS PLASTIC BODY AT MOLD PARTING LINE.
2. DIMENSION G1, TRUE POSITION TO BE MEASURED AT DATUM -T-, SEATING PLANE.
3. DIMENSION R AND U DO NOT INCLUDE MOLD FLASH. ALLOWABLE MOLD FLASH IS 0.010 (0.25) PER SIDE.
4. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
5. CONTROLLING DIMENSION: INCH.
6. THE PACKAGE TOP MAY BE SMALLER THAN THE PACKAGE BOTTOM BY UP TO 0.012 (0.300). DIMENSIONS R AND U ARE DETERMINED AT THE OUTERMOST EXTREMES OF THE PLASTIC BODY EXCLUSIVE OF THE MOLD FLASH, TIE BAR BURRS, GATE BURRS AND INTERLEAD FLASH, BUT INCLUDING ANY MISMATCH BETWEEN THE TOP AND BOTTOM OF THE PLASTIC BODY.
7. DIMINSION H DOES NOT INCLUDE DAMBAR PROTRUSION OR INTRUSION. THE DAMBAR PROTUSION(S) SHALL NOT CAUSE THE H DIMINSION TO BE GREATER THAN 0.037 (0.940136). THE DAMBAR INTRUSION(S) SHALL NOT CAUSE THE H DIMINISION TO SMALLER THAN 0.025 (0.635).

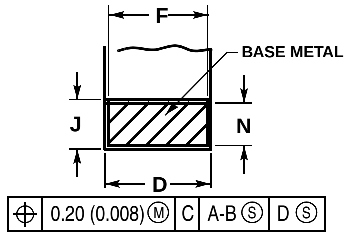
	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.685	0.695	17.40	17.65
B	0.685	0.695	17.40	17.65
C	0.165	0.180	4.20	4.57
E	0.090	0.110	2.29	2.79
F	0.013	0.019	0.33	0.48
G	0.050 BSC		1.27 BSC	
H	0.026	0.032	0.66	0.81
J	0.020	—	0.51	—
K	0.025	—	0.64	—
R	0.650	0.656	16.51	16.66
U	0.650	0.656	16.51	16.66
V	0.042	0.048	1.07	1.21
W	0.042	0.048	1.07	1.21
X	0.042	0.056	1.07	1.42
Y	—	0.020	—	0.50
Z	2°	10°	2°	10°
G1	0.610	0.630	15.50	16.00
K1	0.040	—	1.02	—

10.5 44-Pin QFP (Case 824A-01)

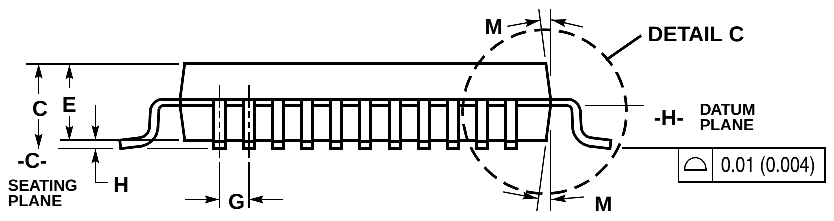


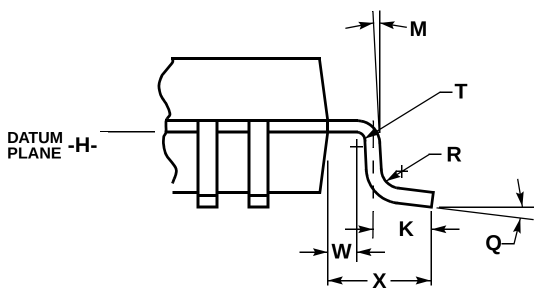


DETAIL A



SECTION B-B





DETAIL C

NOTES:

1. 1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. 2. CONTROLLING DIMENSION: MILLIMETER.
3. 3. DATUM PLANE -H- IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. 4. DATUMS -A-, -B- AND -D- TO BE DETERMINED AT DATUM PLANE -H-.
5. 5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE -C-.
6. 6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -H-.
7. 7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08 (0.003) TOTAL IN EXCESS OF THE D DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	9.90	10.10	0.390	0.398
B	9.90	10.10	0.390	0.398
C	2.10	2.45	0.083	0.096
D	0.30	0.45	0.012	0.018
E	2.00	2.10	0.079	0.083
F	0.30	0.40	0.012	0.016
G	0.80 BSC		0.031 BSC	
H	---	0.25	---	0.010
J	0.13	0.23	0.005	0.009
K	0.65	0.95	0.026	0.037
L	8.00 REF		0.315 REF	
M	5°	10°	5°	10°
N	0.13	0.17	0.005	0.007
Q	0°	7°	0°	7°
R	0.13	0.30	0.005	0.012
S	12.95	13.45	0.510	0.530
T	0.13	---	0.005	---
U	0°	---	0°	---
V	12.95	13.45	0.510	0.530
W	0.40	---	0.016	---
X	1.6 REF		0.063 REF	

Appendix A. MC68HC11D3 and MC68HC11D0

A.1 Introduction

The MC68HC11D3 and MC68HC11D0 are read-only memory (ROM) based high-performance microcontrollers (MCU) based on the MC68HC11E9 design. Members of the Dx series are derived from the same mask and feature a high-speed multiplexed bus capable of running at up to 3 MHz and a fully static design that allows operations at frequencies to dc. The only difference between the MCUs in the Dx series is whether the ROM has been tested and guaranteed.

The information contained in this document applies to both the MC68HC11D3 and MC68HC11D0 with the differences given in this appendix.

Features of the MC68HC11D3 and MC68HC11D0 include:

- 4 Kbytes of on-chip ROM (MC68HC11D3)
- 0 bytes of on-chip ROM (MC68HC11D0)
- 192 bytes of on-chip random-access memory (RAM) all saved during standby
- 16-bit timer system:
  - Three input capture (IC) channels
  - Four output compare (OC) channels
  - One IC or OC software-selectable channel
- 32 input/output (I/O) pins:
  - 26 bidirectional I/O pins
  - 3 input-only pins
  - 3 output-only pins
- Available in these packages:
  - 44-pin plastic leaded chip carrier (PLCC)
  - 44-pin quad flat pack (QFP)

A.2 Block Diagram

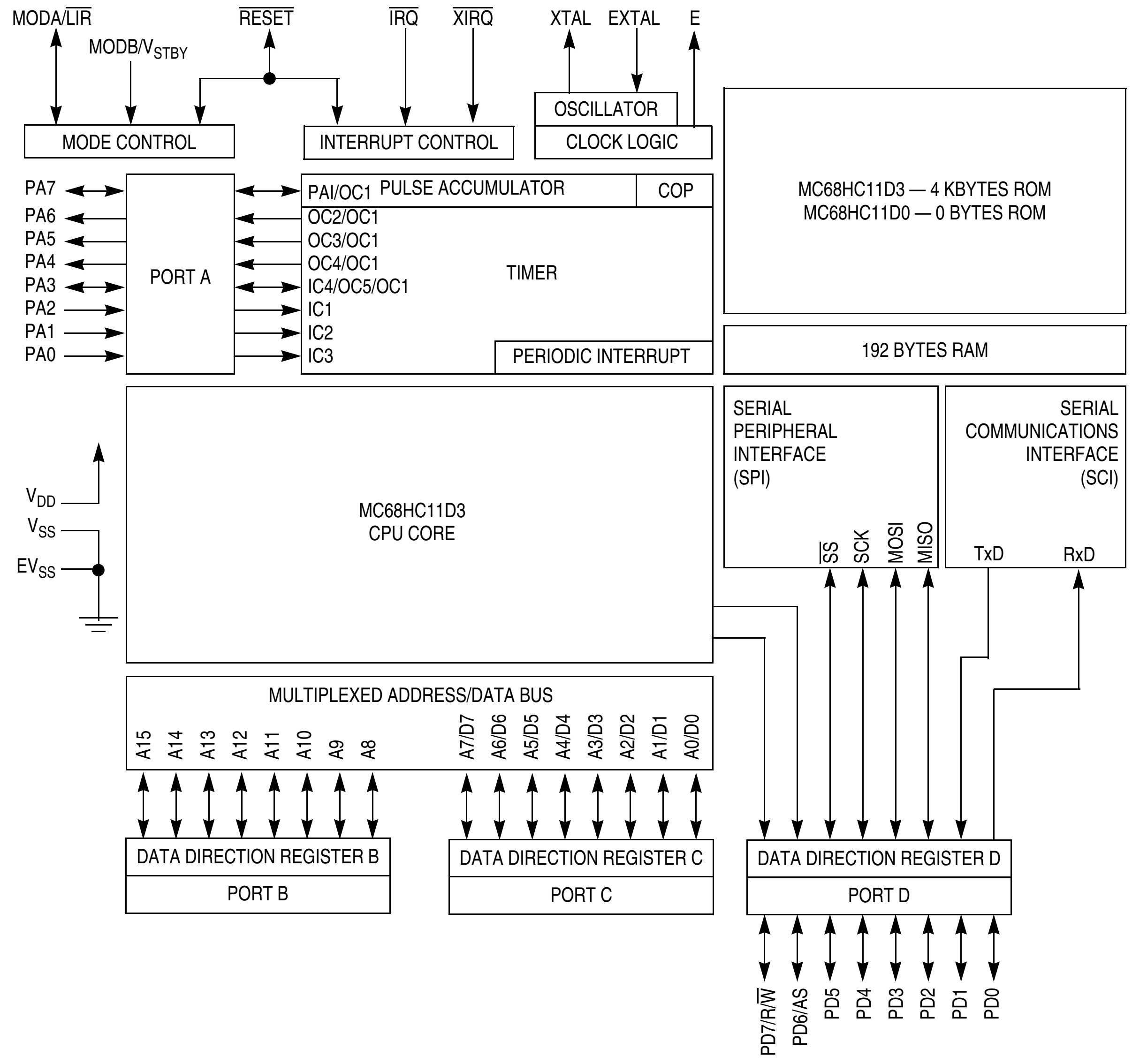


Figure A-1. MC68HC11D3 Block Diagram

A.3 Pin Assignments

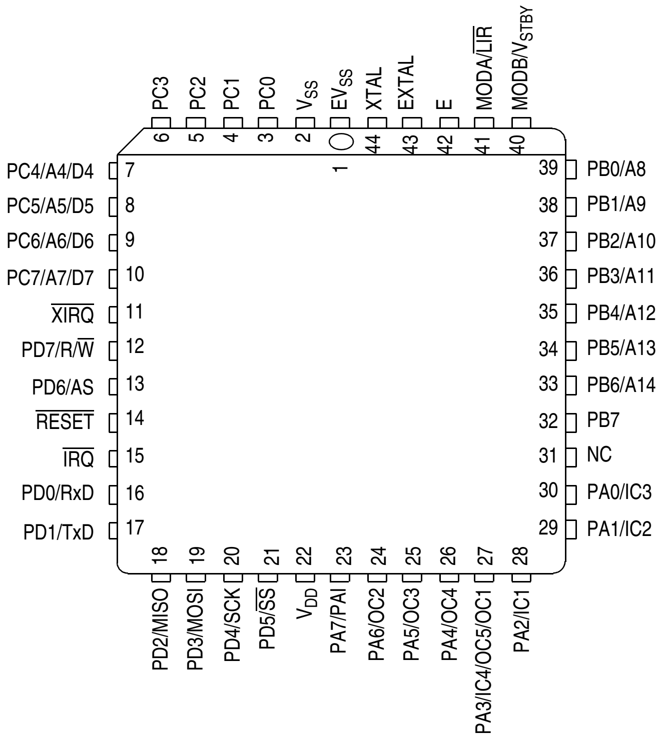


Figure A-2. Pin Assignments for 44-Pin PLCC

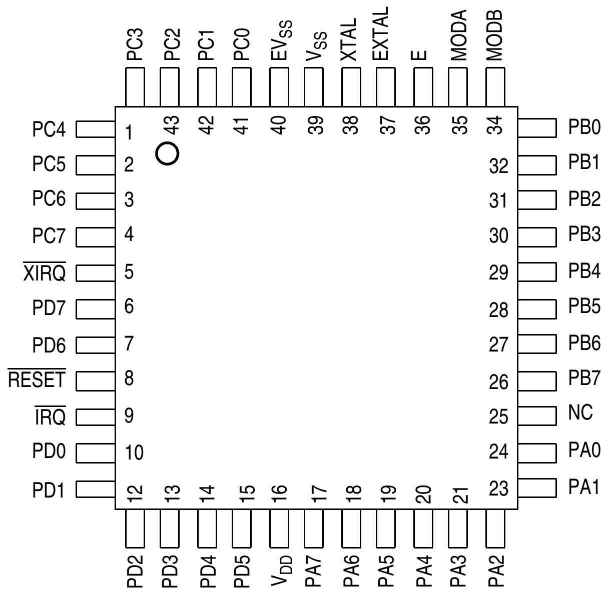


Figure A-3. Pin Assignments for 44-Pin QFP

A.4 Memory Map

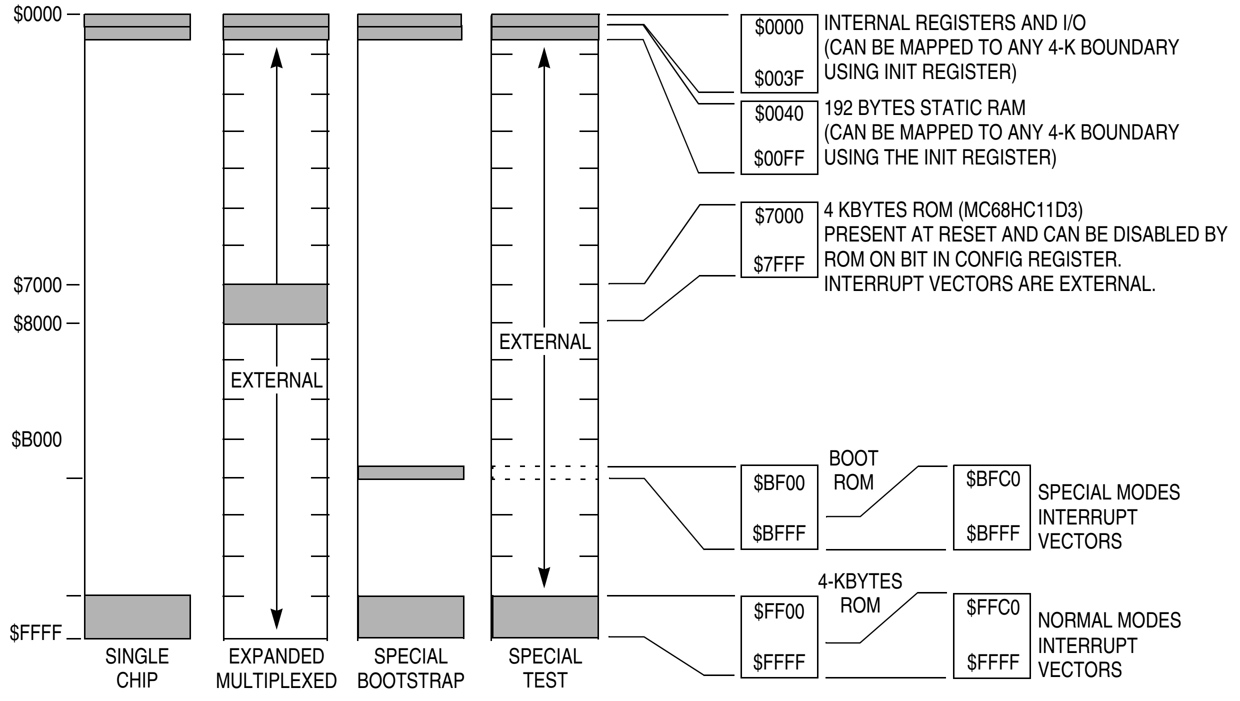

$0000
$7000
$8000
$B000
$FFFF
EXTERNAL
EXTERNAL
SINGLE CHIP
EXPANDED MULTIPLEXED
SPECIAL BOOTSTRAP
SPECIAL TEST
$0000 $003F INTERNAL REGISTERS AND I/O (CAN BE MAPPED TO ANY 4-K BOUNDARY USING INIT REGISTER)
$0040 $00FF 192 BYTES STATIC RAM (CAN BE MAPPED TO ANY 4-K BOUNDARY USING THE INIT REGISTER)
$7000 $7FFF 4 KBYTES ROM (MC68HC11D3) PRESENT AT RESET AND CAN BE DISABLED BY ROM ON BIT IN CONFIG REGISTER. INTERRUPT VECTORS ARE EXTERNAL.
$BF00 $BFFF BOOT ROM
$BFC0 $BFFF SPECIAL MODES INTERRUPT VECTORS
$FF00 $FFFF 4-KBYTES ROM
$FFC0 $FFFF NORMAL MODES INTERRUPT VECTORS

Figure A-4. MC68HC11Dx[1] Memory Map

A.5 MC68HC11D3 and MC68HC11D0 Electrical Characteristics

The parameters given in **Section 9. Electrical Characteristics** apply to the MC68HC11D3 and MC68HC11D0 with the exceptions given here.

A.5.1 Functional Operating Temperature Range

Rating	Symbol	Value	Unit
Operating temperature range MC68HC11D0C	T_A	T_L to T_H –40 to +85	°C

A.5.2 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Package thermal resistance (junction-to-ambient) 44-pin plastic leaded chip carrier (PLCC) 44-pin plastic quad flat pack (QFP	Θ_{JA}	 50 85	°C/W

1. MC68HC11D0 only operates in expanded multiplexed mode and bootstrap mode.

A.6 Ordering Information

MCU	Package	Temperature	MC Order Number	
			2 MHz	**3 MHz**
MC68HC11D3 (Custom ROM)	44-pin PLCC	–40 to +85°C	MC68HC11D3CFN2	MC68HC11D3CFN3
MC68HC11D0 (No ROM)	44-pin PLCC	–40 to +85°C	MC68HC11D0CFN2	MC68HC11D0CFN3
	44-pin QFP	–40 to +85°C	MC68HC11D0CFB2	MC68HC11D0CFB3

Appendix B. MC68L11D0

B.1 Introduction

The MC68L11D0 is an extended-voltage version of the MC68HC11D0 microcontroller that can operate in applications that require supply voltages as low as 3.0 volts. Operation is identical to that of the MC68HC11D0 (see **Appendix A. MC68HC11D3 and MC68HC11D0**) in all aspects other than electrical parameters, as shown in this appendix.

Features of the MC68HC11D0 include:

- Suitable for battery-powered portable and hand-held applications
- Excellent for use in devices such as remote sensors and actuators
- Operating performance is same at 5 V and 3 V

B.2 MC68L11D0 Electrical Characteristics

The parameters given in **Section 9. Electrical Characteristics** apply to the MC68L11D0 with the exceptions given here.

B.2.1 Functional Operating Temperature Range

Rating	Symbol	Value	Unit
Operating temperature range	T_A	T_L to T_H –20 to +70	°C

B.2.2 DC Electrical Characteristics

Characteristic[1]		Symbol	Min	Max	Unit
Output voltage[2] I_{Load} = ± 10.0 µA	All outputs except XTAL All outputs except XTAL, $\overline{\text{RESET}}$, and MODA	V_{OL} V_{OH}	— V_{DD} – 0.1	0.1 —	V
Output high voltage[1] I_{Load} = – 0.5 mA, V_{DD} = 3.0 V I_{Load} = – 0.8 mA, V_{DD} = 4.5 V	All outputs except XTAL, $\overline{\text{RESET}}$, and MODA	V_{OH}	V_{DD} – 0.8	—	V
Output low voltage I_{Load} = 1.6 mA, V_{DD} = 5.0 V I_{Load} = 1.0 mA, V_{DD} = 3.0 V	All outputs except XTAL	V_{OL}	—	0.4	V

The dc electrical table continues on next page.

Characteristic[1]	Symbol	Min	Max	Unit
Input high voltage — All inputs except $\overline{RESET}$ $\overline{RESET}$	V_{IH}	$0.7 \times V_{DD}$ $0.8 \times V_{DD}$	$V_{DD} + 0.3$ $V_{DD} + 0.3$	V
Input low voltage All inputs	V_{IL}	$V_{SS} – 0.3$	$0.2 \times V_{DD}$	V
I/O ports, three-state leakage — PA7, PA3, PC7–PC0, PD7–PD0, MODA/$\overline{LIR}$, $\overline{RESET}$ $V_{In} = V_{IH}$ or V_{IL}	I_{OZ}	—	±10	µA
Input leakage current $V_{In} = V_{DD}$ or V_{SS} — PA2–PA0, $\overline{IRQ}$, $\overline{XIRQ}$ $V_{In} = V_{DD}$ or V_{SS} — MODB/V_{STBY}	I_{In}	 — —	 ±1 ±10	µA
RAM standby voltage — Power down	V_{SB}	2.0	V_{DD}	V
RAM standby current — Power down	I_{SB}	—	10	µA
Input capacitance — PA2–PA0, $\overline{IRQ}$, $\overline{XIRQ}$, EXTAL PA3, PA7, PC7–PC0, PD7–PD0, MODA/$\overline{LIR}$, $\overline{RESET}$	C_{In}	— —	8 12	pF
Output load capacitance — All outputs except PD4–PD1 PD4–PD1	C_L	— —	90 100	pF
Total supply current[3] RUN: Single-chip mode $V_{DD} = 5.5$ V $V_{DD} = 3.0$ V Expanded multiplexed mode $V_{DD} = 5.5$ V $V_{DD} = 3.0$ V	I_{DD}	 8 4 14 7	 15 8 27 14	mA
WAIT — All peripheral functions shut down: Single-chip mode $V_{DD} = 5.5$ V $V_{DD} = 3.0$ V Expanded multiplexed mode $V_{DD} = 5.5$ V $V_{DD} = 3.0$ V	W_{IDD}	 3 1.5 5 2.5	 6 3 10 5	mA
STOP — No clocks, single-chip mode: $V_{DD} = 5.5$ V $V_{DD} = 3.0$ V	S_{IDD}	 50 25	 50 25	µA
Power dissipation Single-chip mode $V_{DD} = 5.5$ V $V_{DD} = 3.0$ V Expanded multiplexed mode $V_{DD} = 5.5$ V $V_{DD} = 3.0$ V	P_D	 44 12 77 21	 85 24 150 42	mW

1. $V_{DD} = 3.0$ Vdc to 5.5 Vdc, $V_{SS} = 0$ Vdc, $T_A = T_L$ to T_H, unless otherwise noted.
2. V_{OH} specification for $\overline{RESET}$ and MODA is not applicable because they are open-drain pins. V_{OH} specification is not applicable to ports C and D in wired-OR mode.
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;
   $V_{IL} \leq 0.2$ V;
   $V_{IH} \geq V_{DD} – 0.2$ V;
   No dc loads

B.2.3 Control Timing

Characteristic[1]	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} + 50 ns	t_{PCSU}	325	—	200	—	ns
Reset input pulse width[2]	PW_{RSTL}					t_{cyc}
To guarantee external reset vector		8	—	8	—	
Minimum input time can be preempted by internal reset		1	—	1	—	
Interrupt pulse width, PW_{IRQ} = t_{cyc} + 20 ns $\overline{IRQ}$ edge-sensitive mode	PW_{IRQ}	1020	—	520	—	ns
Wait recovery startup time	t_{WRS}	—	4	—	4	t_{cyc}
Timer pulse width PW_{TIM} = t_{cyc} + 20 ns Input capture pulse accumulator input	PW_{TIM}	1020	—	520	—	ns

1. V_{DD} = 3.0 Vdc to 5.5 Vdc, V_{SS} = 0 Vdc, T_A = T_L to T_H. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. 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 4. Resets, Interrupts, and Low-Power Modes** for further details.

B.2.4 Peripheral Port Timing

Characteristic[1]	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[2] MCU read of ports A, B, C, and D	t_{PDSU}	100	—	100	—	ns
Peripheral data hold time[2] MCU read of ports A, B, C, and D	t_{PDH}	50	—	50	—	ns
Delay time, peripheral data write	t_{PWD}					ns
MCU write to port A		—	200	—	200	
MCU writes to ports B, C, and D t_{PWD} = 1/4 t_{cyc} + 150 ns		—	350	—	225	

1. V_{DD} = 3.0 Vd to 5.5 Vdc, V_{SS} = 0 Vdc, T_A = T_L to T_H. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
2. Port C and D timing is valid for active drive (CWOM and DWOM bits not set in PIOC and SPCR registers respectively).

B.2.5 Expansion Bus Timing

Num	Characteristic[1]	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, $PW_{EL} = 1/2\ t_{cyc}$ — 23 ns	PW_{EL}	475	—	225	—	ns
3	Pulse width, E high, $PW_{EH} = 1/2\ t_{cyc} - 28$ ns	PW_{EH}	470	—	220	—	ns
4A	E and AS rise time	t_r	—	25	—	25	ns
4B	E and AS fall time	t_f	—	25	—	25	ns
9	Address hold time[(2)a], $t_{AH} = 1/8\ t_{cyc} - 29.5$ ns	t_{AH}	95	—	33	—	ns
12	Non-muxed address valid time to E rise $t_{AV} = PW_{EL} - (t_{ASD} + 80\text{ ns})$[(2)a]	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, $t_{DDW} = 1/8\ t_{cyc} + 65.5$ ns[(2)a]	t_{DDW}	—	195	—	133	ns
21	Write data hold time, $t_{DHW} = 1/8\ t_{cyc} - 29.5$ ns[(2)a]	t_{DHW}	95	—	33	—	ns
22	Muxed address valid time to E rise $t_{AVM} = PW_{EL} - (t_{ASD} + 90\text{ ns})$[(2)a]	t_{AVM}	265	—	78	—	ns
24	Muxed address valid time to AS fall $t_{ASL} = PW_{ASH} - 70$ ns	t_{ASL}	150	—	25	—	ns
25	Muxed address hold time, $t_{AHL} = 1/8\ t_{cyc} - 29.5$ ns[(2)b]	t_{AHL}	95	—	33	—	ns
26	Delay time, E to AS rise, $t_{ASD} = 1/8\ t_{cyc} - 9.5$ ns[(2)a]	t_{ASD}	120	—	58	—	ns
27	Pulse width, AS high, $PW_{ASH} = 1/4\ t_{cyc} - 29$ ns	PW_{ASH}	220	—	95	—	ns
28	Delay time, AS to E rise, $t_{ASED} = 1/8\ t_{cyc} - 9.5$ ns[(2)b]	t_{ASED}	120	—	58	—	ns
29	MPU address access time[(2)a] $t_{ACCA} = t_{cyc} - (PW_{EL} - t_{AVM}) - t_{DSR} - t_f$	t_{ACCA}	735	—	298	—	ns
35	MPU access time , $t_{ACCE} = PW_{EH} - t_{DSR}$	t_{ACCE}	—	440	—	190	ns
36	Muxed address delay (previous cycle MPU read) $t_{MAD} = t_{ASD} + 30$ ns[(2)a]	t_{MAD}	150	—	88	—	ns

1. V_{DD} = 3.0 Vdc to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.
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).

B.2.6 Serial Peripheral Interface Timing

Num	Characteristic[1]	Symbol	1.0 MHz		2.0 MHz		Unit
			Min	Max	Min	Max	
	Operating frequency Master Slave	$f_{op(m)}$ $f_{op(s)}$	dc dc	0.5 1.0	dc dc	0.5 2.0	f_{op} MHz
1	Cycle time Master Slave	$t_{cyc(m)}$ $t_{CYC(s)}$	2.0 1000	— —	2.0 500	— —	t_{cyc} ns
2	Enable lead time Master[2] Slave	$t_{lead(m)}$ $t_{lead(s)}$	— 500	— —	— 250	— —	ns
3	Enable lag time Master[2] Slave	$t_{lag(m)}$ $t_{lag(s)}$	— 500	— —	— 250	— —	ns
4	Clock (SCK) high time Master Slave	$t_{w(SCKH)m}$ $t_{w(SCKH)s}$	680 380	— —	340 190	— —	ns
5	Clock (SCK) low time Master Slave	$t_{w(SCKL)m}$ $t_{w(SCKL)s}$	680 380	— —	340 190	— —	ns
6	Data setup time (inputs) Master Slave	$t_{su(m)}$ $t_{su(s)}$	100 100	— —	100 100	— —	ns
7	Data hold time (inputs) Master Slave	$t_{h(m)}$ $t_{h(s)}$	100 100	— —	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)[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) SPI inputs (SCK, MOSI, MISO, and $\overline{SS}$)	t_{rm} t_{rs}	— —	100 2.0	— —	100 2.0	ns µs
13	Fall time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI outputs (SCK, MOSI, and MISO) SPI inputs (SCK, MOSI, MISO, and $\overline{SS}$)	t_{fm} t_{fs}	— —	100 2.0	— —	100 2.0	ns µs

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

B.3 Ordering Information

Package	Frequency	Features	MC Order Number
44-pin PLCC	2 MHz	No ROM	MC68L11D0FN2
44-pin QFP	2 MHz	No ROM	MC68L11D0FB2

HOW TO REACH US:

USA/EUROPE/LOCATIONS NOT LISTED:
Motorola Literature Distribution
P.O. Box 5405
Denver, Colorado 80217
1-800-521-6274 or 480-768-2130

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

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

HOME PAGE:
http://motorola.com/semiconductors

MOTOROLA

MC68HC711D3/D
Rev. 2
9/2003

68HC711D3 : Microcontroller

SUBSCRIBE FOR UPDATES

The MC68HC711D3 EPROM-based microcontrollers (MCUs)is a low cost alternative to applications needing a high-performance MCU. The MC68HC711D3 features a multiplexed bus capable of running at up to 3 MHz, and a fully static design that allows operations at frequencies to dc.

Rate this Page

○ -- ○ - ○ 0 ○ + ○ ++

Submit

Care to Comment?

68HC711D3 Features

- MC68HC11 CPU
- Power Saving STOP and WAIT Modes
- 4 Kbytes of On-Chip EPROM
- 192 Bytes of On-Chip RAM (All Saved During Standby)
- 16-Bit Timer System
  - 3 Input Capture (IC) Channels
  - 4 Output Compare (OC) Channels
  - One IC or OC Channel (Software Selectable)
- 8-Bit Pulse Accumulator
- Real-Time Interrupt Circuit
- Computer Operating Properly (COP) Watchdog System

Return to Top

68HC711D3 Documentation

Documentation

Application Note

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
AN1010/D	MC68HC11 EEPROM Programming from a Personal Computer	MOTOROLA	pdf	291	1	5/20/2002	ORDER
AN1050_D	Designing for Electromagnetic Compatibility (EMC) with HCMOS Microcontrollers	MOTOROLA	pdf	82	0	1/01/2000	-
AN1058/D	Reducing A/D Errors in Microcontroller Applications	MOTOROLA	pdf	245	0	2/01/2001	ORDER
AN1060/D	M68HC11 Bootstrap Mode	MOTOROLA	pdf	289	1	1/01/1999	ORDER
AN1060SW	Software Files for AN1060 zipped	MOTOROLA	zip	176	0	1/01/1995	-
AN1064/D	Use of Stack Simplifies M68HC11 Programming	MOTOROLA	pdf	522	0	1/11/2001	ORDER
AN1067/D	Pulse Generation and Detection with Microcontroller Units	MOTOROLA	pdf	242	1	5/31/2002	ORDER
AN1220_D	Optical Character Recognition Using Fuzzy Logic	MOTOROLA	pdf	317	0	1/01/1996	-

AN1259/D	System Design and Layout Techniques for Noise Reduction in MCU-Based Systems	MOTOROLA	pdf	78	0	1/01/1995	ORDER
AN1263/D	Designing for Electromagnetic Compatibility with Single-Chip Microcontrollers	MOTOROLA	pdf	104	0	1/01/1995	ORDER
AN1705/D	Noise Reduction Techniques for Microcontroller-Based Systems	MOTOROLA	pdf	67	0	1/01/1999	ORDER
AN1706/D	Microcontroller Oscillator Circuit Design Considerations	MOTOROLA	pdf	103	0	1/01/1997	ORDER
AN1744/D	Resetting Microcontrollers During Power Transitions	MOTOROLA	pdf	80	0	1/01/1998	ORDER
AN1752/D	Data Structures for 8-Bit Microcontrollers	MOTOROLA	pdf	213	1	5/07/2001	ORDER
AN1771/D	Precision Sine-Wave Tone Synthesis Using 8-Bit MCUs	MOTOROLA	pdf	250	0	1/01/1998	ORDER
AN1775/D	Expanding Digital Input with an A/D Converter	MOTOROLA	pdf	86	1	1/01/1998	ORDER
AN1783/D	Determining MCU Oscillator Start-Up Parameters	MOTOROLA	pdf	48	1	1/01/1999	ORDER
AN2103/D	Local Interconnect Network (LIN) Demonstration	MOTOROLA	pdf	953	0	12/01/2000	ORDER
AN2321/D	Designing for Board Level Electromagnetic Compatibility	MOTOROLA	pdf	1628	0	8/15/2002	ORDER
AN427/D	MC68HC11 EEPROM Error Correction Algorithms in C	MOTOROLA	pdf	147	0	1/01/2000	ORDER
AN432/D	128K Byte Addressing with the M68HC11	MOTOROLA	pdf	435	0	1/11/2001	ORDER
AN461/D	An Introduction to the HC16 for HC11 Users	MOTOROLA	pdf	667	0	1/01/2000	ORDER
AN494/D	An HC11-Controlled Multiband RDS Radio	MOTOROLA	pdf	1052	1	2/23/2001	ORDER
AN495/D	RDS decoding for an HC11-controlled	MOTOROLA	pdf	3840	0	10/01/1994	ORDER
AN974/D	MC68HC11 Floating-Point Package	MOTOROLA	pdf	299	0	1/01/2000	ORDER
AN997/D	CONFIG Register Issues Concerning the M68HC11 Family	MOTOROLA	pdf	53	0	1/01/2000	ORDER
ANE415/D	MC68HC11 Implementation of IEEE-488 Interface for DSP56000 Monitor	MOTOROLA	pdf	1619	0	1/01/1988	ORDER

Brochure

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
FLYREMBEDFLASH/D	Embedded Flash: Changing the Technology World for the Better	MOTOROLA	pdf	68	2	5/21/2003	ORDER

Data Sheets

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
MC68HC711D3	MC68HC711D3 Technical Data	MOTOROLA	pdf	2041	2	9/17/2003	ORDER

Engineering Bulletin

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
EB182/D	How the Romon Bit Behaves on the E Series HC11 MCUs	MOTOROLA	pdf	28	0	1/01/1998	ORDER
EB184/D	Enabling the Security Feature on the MC68HC711E9 Devices with Pcbug11 on the M68HC711E9PGMR	MOTOROLA	pdf	30	0	1/01/1998	ORDER
EB185/D	Simplify MC68HC711E9PROM Programming with Pcbug11 and the M68HC711EPGMR Board	MOTOROLA	pdf	29	0	1/01/1998	ORDER
EB187/D	Programming MC68HC711E9 Devices with Pcbug11 and the M68HC711EVB	MOTOROLA	pdf	34	0	1/01/1998	ORDER
EB188/D	Enabling the Security Feature on M68HC811E2 Devices with PCbug11 on the M68HC711E9PGMR	MOTOROLA	pdf	29	0	1/01/1998	ORDER
EB189/D	Programming MC68HC811E2 Devices with Pcbug11 and the M68HC711E9PGMR	MOTOROLA	pdf	36	0	1/01/1998	ORDER
EB191/D	Programming EPROM and EEPROM on the M68HC11EVM	MOTOROLA	pdf	26	0	1/01/1999	ORDER
EB192/D	A Quick PWM Tutorial for MC68HC11 K, KA, KW, P and PH Series Controllers	MOTOROLA	pdf	101	0	1/14/2003	ORDER
EB193/D	Replacing 68HC11A Series MCUs with 68HC11E Series MCUs	MOTOROLA	pdf	96	1	1/01/1999	ORDER
EB195/D	How to Configure the Reset Pin on the MC68HC11	MOTOROLA	pdf	25	0	1/01/1999	ORDER
EB197/D	Using Pseudo-Interrupt Vectors on the M68HC11EVBU	MOTOROLA	pdf	15	0	1/01/1999	ORDER
EB198/D	Turn Off Your E Clock to Reduce Noise Emission on the MC68HC11	MOTOROLA	pdf	57	0	1/01/1998	ORDER
EB254/D	Setting the Programming Voltage on Modular Microcontrollers with FLASH EEPROM	MOTOROLA	pdf	20	0	1/01/1998	ORDER
EB284/D	C Macro Definitions for the MC68HC(7)11D3/D0	MOTOROLA	pdf	23	0	1/01/1998	ORDER
EB285/D	C Macro Definitions for the MC68HC(7)11E20	MOTOROLA	pdf	23	0	1/01/1998	ORDER
EB287/D	C Macro Definitions for the MC68HC(7)11E9/E8/E1/E0	MOTOROLA	pdf	25	0	1/01/1999	ORDER
EB289/D	C Macro Definitions for the MC68HC11F1	MOTOROLA	pdf	26	0	1/01/1998	ORDER
EB291/D	Programming MC68HC811E2 Devices with Pcbug11 and the M68HC11EVBU	MOTOROLA	pdf	37	0	1/01/1998	ORDER
EB292/D	Initialization Considerations When Moving from the BUFFALO Monitor to a Standalone MC68HC11	MOTOROLA	pdf	22	0	1/01/1998	ORDER
EB293/D	Simplify MC68HC711E20 EPROM Programming with Pcbug11	MOTOROLA	pdf	30	0	1/01/1998	ORDER
EB294/D	How to Write to the 64-Cycle Time-Protected Registers on M68HC11 Development Tools	MOTOROLA	pdf	46	0	1/01/1999	ORDER
EB295/D	Programming the EEPROM on the MC68HC811E2 with the M68HC11EVM Board	MOTOROLA	pdf	26	0	1/01/1998	ORDER
EB296/D	Programming MC68HC711E9 Devices with Pcbug11 and the M68HC11EVBU	MOTOROLA	pdf	28	0	1/01/1998	ORDER
EB298/D	Programming the BUFFALO Monitor into an MC68HC711E9	MOTOROLA	pdf	22	0	1/01/1999	ORDER
EB299/D	Why M68HC711D3PGMR Software Does Not Run on 486 33-MHz Computers	MOTOROLA	pdf	19	0	1/01/1998	ORDER
EB301/D	Programming EEPROM on the MC68HC811E2 during Program Execution	MOTOROLA	pdf	24	0	1/01/1999	ORDER
EB303/D	Handling Considerations for Avoiding Intermittent Programming and Execution Failures with MC68HC11-Windowed EPROM Devices	MOTOROLA	pdf	33	0	1/01/1998	ORDER

EB312/D	Replacing 68HC11KA4/KA2 MCUs with 68HC11KS2/KS8 MCUs	MOTOROLA	pdf	69	0	1/01/1999	ORDER
EB349/D	RAM Data Retention Considerations for Motorola Microcontrollers	MOTOROLA	pdf	45	1	6/22/2000	ORDER
EB378/D	CONFIG Register Programming for EEPROM-Based M68HC11 Microcontrollers	MOTOROLA	pdf	114	0	1/31/2001	ORDER
EB380/D	Migrating from the MC68HC811E2 to the MC68HC711E9	MOTOROLA	pdf	104	0	3/02/2001	ORDER
EB381/D	Migrating from the MC68HC811E2 to the MC68HC11F1	MOTOROLA	pdf	137	0	5/10/2001	ORDER
EB396/D	Use of OSC2/XTAL as a Clock Output on Motorola Microcontrollers	MOTOROLA	pdf	49	0	6/19/2002	ORDER
EB413/D	Resetting MCUs	MOTOROLA	pdf	62	0	1/01/2000	ORDER
EB422/D	Enhanced M68HC11 Bootstrap Mode	MOTOROLA	pdf	1377	0	1/01/2000	ORDER

Errata - Click here for important errata information

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
68HC711D3MSE4/D	XC68HC711D3 Device Information: C45A Mask Sets	MOTOROLA	pdf	10	4	3/26/1998	-

Product Change Notices

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
PCN7701	QFP 10X10 ASSY MOVE FROM SHC TO BAT3	MOTOROLA	htm	16	-	7/09/2002	-

Quick Reference Guide

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
M68HC11CFG/D	CONFIG Register Programming for EEPROM-based M68HC11 Microcontrollers	MOTOROLA	pdf	505	1	1/01/1995	ORDER

Reference Manual

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
M68HC11ERG	M68HC11E Programming Reference Guide	MOTOROLA	pdf	1238	2	10/31/2003	ORDER
M68HC11RM/D	M68HC11 Reference Manual	MOTOROLA	pdf	6400	6	4/09/2002	ORDER
MC68HC11D3RG/AD	MC68HC11D3 Programming Reference Guide	MOTOROLA	pdf	4697	0	6/01/1990	ORDER
MC68HC11F1RG/AD	MC68HC11F1 Programming Reference Guide	MOTOROLA	pdf	4765	2	4/01/1992	ORDER

Selector Guide

ID	Name	Vendor ID	Format	Size K	Rev #	Date Last Modified	Order Availability
SG1006	Microcontrollers Selector Guide - Quarter 4, 2003	MOTOROLA	pdf	826	0	10/24/2003	ORDER
SG1011	Software and Development Tools Selector Guide - Quarter 4, 2003	MOTOROLA	pdf	287	0	10/24/2003	ORDER

Return to Top

Hardware Tools

Emulators/Probes/Wigglers

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
AX-6811	AX-6811	HITEX	-	-	-	-
IC10000	iC1000 PowerEmulator	ISYS	-	-	-	-
IC20000	iC2000 PowerEmulator	ISYS	-	-	-	-
IC40000	iC4000 ActiveEmulator	ISYS	-	-	-	-
EMUL68-PC	EMUL68-PC	NOHAU	-	-	-	-

Evaluation/Development Boards and Systems

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
M68CBL05C	Low-noise Flex Cable	MOTOROLA	-	-	-	BUY

Programmers

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
AP520	Automated Programming System	SYSGEN	-	-	-	-
T9600	High-speed universal gang programmer	SYSGEN	-	-	-	-

Software

Application Software

Application Development Framework

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
HC11D3HCOD	C Header File for 68HC11D3	MOTOROLA	zip	10	-	-

Bootloader Code

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
BOOT7D3FW	Bootstrap Mode Code	MOTOROLA	lst	18	-	-

Code Examples

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
AN1010SW	Software Files for AN1010 zipped Software files for AN1010	MOTOROLA	zip	113	0	-
B2D04COD	Binary to BCD routine	MOTOROLA	asm	1	-	-
D2B04COD	BCD to Binary routine	MOTOROLA	asm	0	-	-
DIV48COD	24-Bit Multiply and 48-Bit Divide routines	MOTOROLA	zip	4	-	-
EXAMPLESCOD	Examples from HC11 Reference Manual	MOTOROLA	zip	14	1	-
FFTHC11COD	FFT routine for HC11	MOTOROLA	asm	12	-	-
FLOAT11COD	Floating Point routines	MOTOROLA	zip	6	-	-
FP11COD	Floating Point routines	MOTOROLA	asm	11	-	-
GMATHCOD	General Math routines	MOTOROLA	asm	9	-	-
HC11FP11COD	Floating Point routines	MOTOROLA	zip	19	-	-
MUL16C11COD	16 x 16 Multiply routine	MOTOROLA	asm	1	-	-
SOUNDFXCOD	Sound Effects example	MOTOROLA	zip	3	-	-

Operating Systems

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
MCX11V15RTOS	Microcontroller Executive	MOTOROLA	arc	92	-	-
CMX-RTX	CMX-RTX	CMX	-	-	-	-

Software Tools

Assemblers

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
68HC11AS11ASM	DOS based freeware assembler	MOTOROLA	exe	18	-	-
AS11NEWASM	DOS based freeware assembler	MOTOROLA	exe	19	-	-
BASIC11COD	Old source code for BASIC11	MOTOROLA	zip	57	-	-
ADX-11	ADX-11 Macro Assembler-Linker and IDE	AVOCET	-	-	-	-
AX6811	AX6811 relocatable and absolute macro assembler for HC11	COSMIC	-	-	-	-

Compilers

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
CWHC11	CodeWarrior Development Tools for HC11	METROWERKS	-	-	-	BUY
ADC-11	ADC-11 Compiler, Assembler, Simulator, IDE	AVOCET	-	-	-	-
CX6811	CX6811 C Cross Compiler for HC11	COSMIC	-	-	-	-
ICC11	ICC11 V6 STD	IMAGE	-	-	-	-

Debuggers

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
PCBUG11EXEDBG	Bootstrap Mode Programmer & Debugger	MOTOROLA	exe	167	-	-
PCBUG342DBG	Bootstrap Mode Programmer & Debugger	MOTOROLA	exe	138	-	-
PCBUGBDBG	Bootstrap Mode Programmer & Debugger	MOTOROLA	zip	107	-	-
CWHC11	CodeWarrior Development Tools for HC11	METROWERKS	-	-	-	BUY
ZAP 6811 SIM	ZAP 6811 Simulator Debugger	COSMIC	-	-	-	-
AX-6811	AX-6811	HITEX	-	-	-	-
NOICE11	NoICE11	IMAGE	-	-	-	-

IDE (Integrated Development Environment)

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
CWHC11	CodeWarrior Development Tools for HC11	METROWERKS	-	-	-	BUY
IDEA11	IDEA11 integrated development environment for HC11	COSMIC	-	-	-	-
THRSIM11 4.00	THRSim11	HBROE	-	-	-	-
IC-SW-OPR	winIDEA	ISYS	-	-	-	-

Performance and Testing

ID	Name	Vendor ID	Format	Size K	Rev #	Order Availability
AX-6811	AX-6811	HITEX	-	-	-	-

Return to Top

Orderable Parts Information

PartNumber	Package Info	Tape and Reel	Life Cycle Description (code)	Budgetary Price QTY 1000+ ($US)	Additional Info	Order Availability
KMC711D3CFB3	QFP 44 10*10*2.0P0.8	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	BUY

KMC711D3CFN3	PLCC 44	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	BUY
KMC711D3CP3	PDIP 40	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	BUY
KMC711D3MFN3	PLCC 44	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	BUY
KMC711D3MP2	PDIP 40	No	REMOVED FROM ACTIVE PORTFOLIO(8)	-	more	BUY
MC68HC711D3CFB2	QFP 44 10*10*2.0P0.8	No	PRODUCT MATURITY/SATURATION(4)	$4.68	more	BUY
MC68HC711D3CFB3	QFP 44 10*10*2.0P0.8	No	PRODUCT MATURITY/SATURATION(4)	$4.91	more	BUY
MC68HC711D3CFN2	PLCC 44	No	PRODUCT MATURITY/SATURATION(4)	$4.68	more	BUY
MC68HC711D3CFN3	PLCC 44	No	PRODUCT MATURITY/SATURATION(4)	$4.91	more	BUY
MC68HC711D3CP2	PDIP 40	No	PROD PHASE OUT/SEE LAST ORD DT(6)	$4.68	more	BUY
MC68HC711D3CP3	PDIP 40	No	PROD PHASE OUT/SEE LAST ORD DT(6)	$4.91	more	BUY
MC68HC711D3MFN2	PLCC 44	No	PRODUCT MATURITY/SATURATION(4)	$5.14	more	BUY
MC68HC711D3MFN3	PLCC 44	No	PRODUCT MATURITY/SATURATION(4)	-	more	-
MC68HC711D3MP2	PDIP 40	No	PROD PHASE OUT/SEE LAST ORD DT(6)	$5.15	more	BUY
MC68HC711D3VFN2	PLCC 44	No	PRODUCT MATURITY/SATURATION(4)	$4.91	more	BUY
MC68HC711D3VFN3	PLCC 44	No	PRODUCT MATURITY/SATURATION(4)	$5.15	more	BUY
MCC68HC711D3	CHIPS SM <50000 SQ MILS	No	PRODUCT MATURITY/SATURATION(4)	$4.12	more	BUY

NOTE: Are you looking for an obsolete orderable part? Click **HERE** to check our distributors' inventory.

Return to Top

Related Links

- Automotive
- Microcontrollers

Return to Top