MC68HC711D3/D
Rev. 1

HC11

MC68HC711D3

TECHNICAL
DATA

MOTOROLA

MC68HC711D3

HCMOS MICROCONTROLLER UNIT

Motorola reserves the right to make changes without further notice to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. Motorola products are not authorized for use as components in life support devices or systems intended for surgical implant into the body or intended to support or sustain life. Buyer agrees to notify Motorola of any such intended end use whereupon Motorola shall determine availability and suitability of its product or products for the use intended. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Employment Opportunity/Affirmative Action Employer.

©MOTOROLA INC., 1990

Replaces NP442

PREFACE

MCU Device Part Number Prefixes

Early MCU samples and devices are marked with an "XC" prefix. "XC" devices are tested but are not fully characterized or qualified over the full range of normal manufacturing process variations. After full characterization and qualification, devices will be marked with the "MC" prefix.

MCU Device Date Codes

Device markings indicate the week of manufacture and the mask set used. The data is coded as four numerical digits where the first two digits indicate the year and the last two digits indicate the work week. The date code "9015" would indicate the 15th week of the year 1990.

MCU Device Mask Set Identification

The mask set is identified by a four-character code consisting of a letter, two numerical digits, and a letter (e.g., C45A). Slight variations to the mask set identification code may result in an optional numerical digit preceding the standard four-character code (e.g., 2C45A).

Whenever contacting a Motorola representative for assistance, please have the MCU device mask set and date code information available.

XC68HC711D3 EPROM MCU Programming

On XC68HC711D3 mask set C45A devices, the EPROM emulation mode of EPROM programming does not function correctly. Therefore, the 4K EPROM must be programmed using the special test or bootstrap modes. Motorola recommends that the M68HC11EVM Evaluation Module be used to program the EPROM MCU. Refer to EVM design upgrade number M68HC11EVM/DU9 for EVM hardware/software modifications and adapter information required to program the XC68HC711D3 EPROM MCU.

Two new products (M68HC711D3PGMR Programmer Board and M68HC711D3EVB Evaluation Board) also provide XC68HC711D3 EPROM MCU programming. Contact your Motorola representative for product availability.

The 4K EPROM MCU may be programmed using the special test and bootstrap modes of operation as described in the following paragraphs. Erasure of the EPROM MCU takes approximately 30 to 60 minutes using standard ultraviolet (UV) EPROM erasing equipment.

Program the EPROM MCU at room temperature only.

Cover device window with an opaque label after erasing and before programming the EPROM MCU.

An external +11.75 to +12.75-volt supply must be connected to the XIRQ pin in order to program the EPROM MCU. The programming power supply should be limited to approximately 25 mA to prevent damage to internal MCU circuitry. The high-voltage supply must be off before +5-volt power is applied to or removed from the MCU device.

A special bit, EPROM latch control (ELAT), implemented in bit 5 of the PPROG register, controls EPROM programming. When ELAT is clear, the EPROM array is in the read mode. When ELAT is set, the 4K EPROM array can be programmed but not read. The external programming voltage must be present on the XIRQ pin before ELAT is set.

The following procedure details the sequence to perform byte programming of the 4K EPROM array. Note that the sequence is similar to EEPROM byte programming, except that the ELAT bit is used instead of the EELAT bit.

Step	**Comments**
1. Write $20 to PPROG.	Set ELAT bit (PGM=0) to enable EPROM latches.
2. Write data to EPROM.	
3. Write $21 to PPROG.	Set PGM bit (ELAT=1) to enable EPROM high voltage.
4. Delay 2 to 4 ms.	
5. Write $20 to PPROG.	Turn off high voltage to EPROM array.
6. Repeat steps 2 thru 5 as required.	
7. Write $00 to PPROG.	Return to read mode.

MCU EPROM endurance and data retention performance is not characterized on "XC" devices. While the units have received the same stresses and tests as production MC68HC11 MCU devices, the reliability database has not been established.

XC68HC711D3 EPROM MCU (C45A) ERRATA

The RAM standby current and STOP mode current on C45A mask set devices are higher than expected:

100 μA maximum for RAM standby (ISB)

900 μA maximum for STOP mode (SIDD)

The CPU will not exit STOP mode correctly when interrupted by IRQ or XIRQ if the instruction immediately 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 a NOP, then the STOP command. If reset is used to exit STOP mode, the CPU will respond correctly.

The MCU RAM and registers should not be remapped into the EPROM space ($F000 to $FFFF) on C45A mask set XC68HC711D3 EPROM MCU devices.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1
INTRODUCTION

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

This document contains condensed information on the MC68HC711D3 MCU. For more detailed information, see M68HC11RM/AD, *M68HC11 Reference Manual* available at the local Motorola sales office.

1.1 THE MOTOROLA MC68HC711D3 MCU

The MC68HC711D3 MCU contains highly sophisticated on-chip peripheral functions. This high-speed, low-power programmable read-only memory (PROM) MCU has a nominal bus speed of 2 MHz. The fully static design allows operations at frequencies down to dc.

1.2 SPECIAL FEATURES

Refer to Figure 1-1 and the following list for hardware and software features of the MC68HC711D3:

- Expanded 16-Bit Timer System with Four-Stage Programmable Prescaler
- Enhanced Nonreturn-to-Zero (NRZ) Serial Communications Interface (SCI)
- Power-Saving STOP and WAIT Modes
- 64K Memory Addressability
- Multiplexed Address/Data Bus
- Serial Peripheral Interface (SPI)
- 4K Bytes of Erasable Programmable Read-Only Memory (EPROM) or One-Time Programmable Read-Only Memory (OTPROM)
- 8-Bit Pulse Accumulator Circuit
- 192 Bytes of Static RAM (All Saved during Standby)
- Real-Time Interrupt Circuit
- Computer Operating Properly (COP) Watchdog System

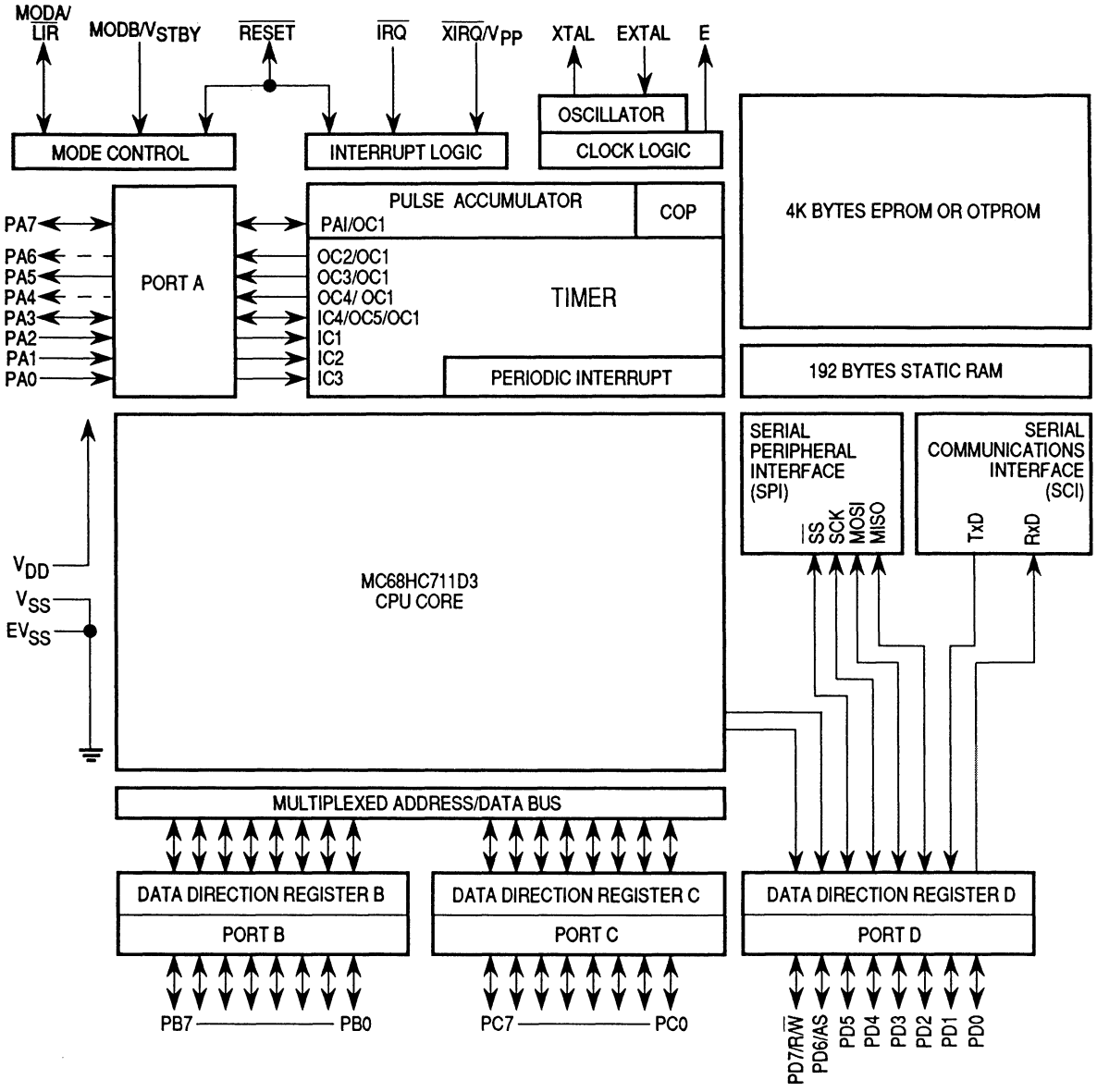


Figure 1-1. MC68HC711D3 Block Diagram

SECTION 2
OPERATING MODES AND SIGNAL DESCRIPTIONS

This section describes the operating modes and signals of the MC68HC711D3.

2.1 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 MCU status or mode selected is latched on bits SMOD and MDA of the HPRIO register 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. Mode selection according to the values encoded on the MODA and MODB pins, and the value latched in the SMOD and MDA bits, are shown in the following table:

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	(Special) EPROM Emulation (PROG)	X	X

2.1.1 Single-Chip Mode

In the single-chip mode, the MCU functions as a self-contained microcontroller and has no external address or data bus. The 4K-byte EEPROM 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.

2.1.2 Expanded-Multiplexed Mode

In the expanded-multiplexed mode, the MCU can address up to 64K bytes 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/$\overline{W}$ 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 modes.

NOTE

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

2.1.3 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 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 2-1), 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 PROM (EPROM or OTPROM) through the MCU is entered automatically if $\overline{IRQ}$ is low and programming voltage is available on the V_{PP} pin. $\overline{IRQ}$ should be pulled up while in reset with MODA and MODB configured for bootstrap mode to prevent unintentional programming of the EPROM. The PROG aspect of bootstrap mode, used for programming the MCU as though it were a standard 27256-type EPROM, is entered by holding a low signal on the MODA, MODB, and $\overline{RESET}$ pins. See **SECTION 7 PROGRAMMABLE READ-ONLY MEMORY (PROM)** for details on the PROG mode.

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 2-1. 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
00E2	Timer Input Capture 3
00E5	Timer Input Capture 2
00E8	Timer Input Capture 1
00EB	Real-Time Interrupt
00EE	$\overline{\text{IRQ}}$
00F1	$\overline{\text{XIRQ}}$
00F4	SWI
00F7	Illegal Opcode
00FA	COP Fail
00FD	Clock Monitor
BF00 (Boot)	Reset

2.1.4 Test Mode

This special expanded mode is primarily intended for 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.

2.1.5 PROM Emulation Mode (PROG)

PROM emulation mode, the PROG aspect of bootstrap mode, is used for programming the MCU as though it were a standard 27256-type EPROM. This mode is entered by holding a low signal on the MODA, MODB, and $\overline{\text{RESET}}$ pins. A socket adapter is required for OTPROM or EPROM programming in this mode. See **SECTION 7 PROGRAMMABLE READ-ONLY MEMORY (PROM)** for details on the PROG mode.

2.2 SIGNAL DESCRIPTION

The following paragraphs describe the signals necessary to the various functions of the MCU.

2.2.1 V_{DD}, V_{SS}, and EV_{SS}

Power is supplied to the MCU using these two pins. V_{DD} is power (+5 V ±10%), V_{SS} is ground (0 V), and EV_{SS} is an extra ground pin available on the 44-pin PLCC packaging version of the MC68HC711D3.

2.2.2 $\overline{\text{RESET}}$

This active-low bidirectional control pin is used as an input to initialize the MCU to a known startup state. It is also used as an open-drain output to indicate that an internal failure has been detected in either the clock monitor or in the computer operating properly (COP) circuit. In addition, the state of this pin is one of the factors governing the selection of the BOOT/PROG mode.

2.2.3 XTAL and EXTAL

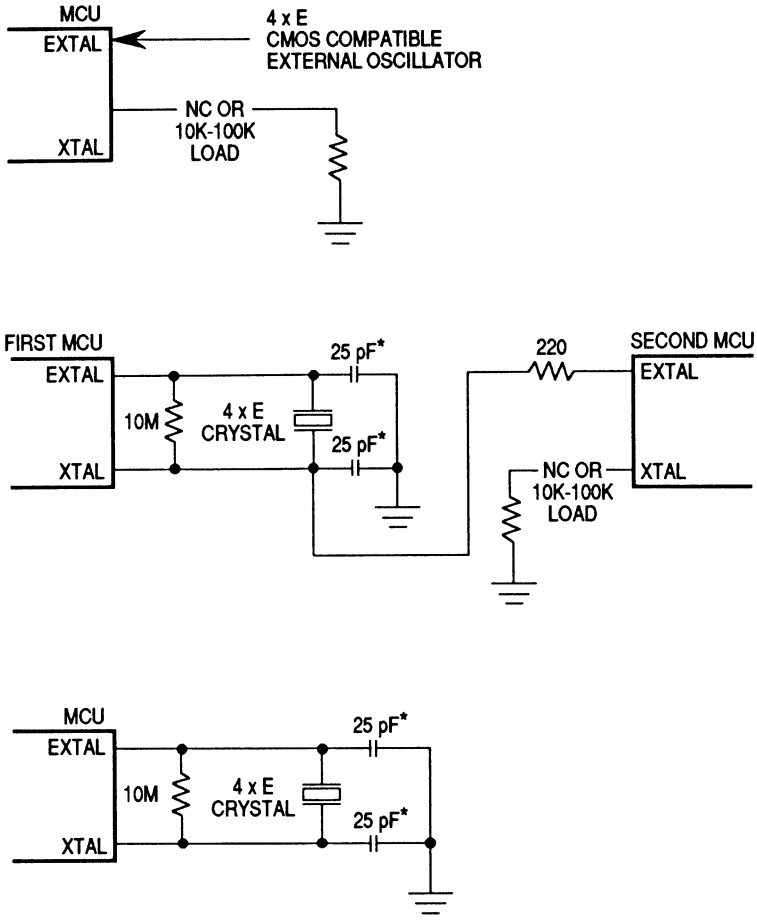
These pins provide the interface for either a crystal or a CMOS-compatible clock to control the internal clock generator circuitry. The frequency applied must be four times higher than the desired clock rate. Refer to Figure 2-1 for crystal and clock connections.

2.2.4 E Clock

This pin provides an output for the internally generated E clock, which can be used for 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 **5.2.6 Highest Priority I Interrupt and Miscellaneous Register (HPRIO)** for details.

2.2.5 $\overline{\text{IRQ}}$/$\overline{\text{CE}}$

The $\overline{\text{IRQ}}$ pin provides the capability for asynchronously applying interrupts to the MCU. Either negative edge-sensitive triggering or level-sensitive triggering is program selectable by using the IRQE bit of the OPTION register. This pin is configured as level sensitive during reset. While the PROM is



* Values include all stray capacitances.

Figure 2-1. Oscillator Connections

being programmed and verified in PROG mode, this pin provides the chip enable ($\overline{CE}$) signal. An external resistor is required on $\overline{IRQ}$ to pull the pin to V_{DD} to prevent accidental programming of the PROM during reset.

2.2.6 $\overline{\text{XIRQ}}/V_{PP}$

The $\overline{\text{XIRQ}}$ pin 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 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 (PROG), this pin is used to supply EPROM or 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.

2.2.7 MODA/$\overline{\text{LIR}}$ and MODB/V_{STBY}

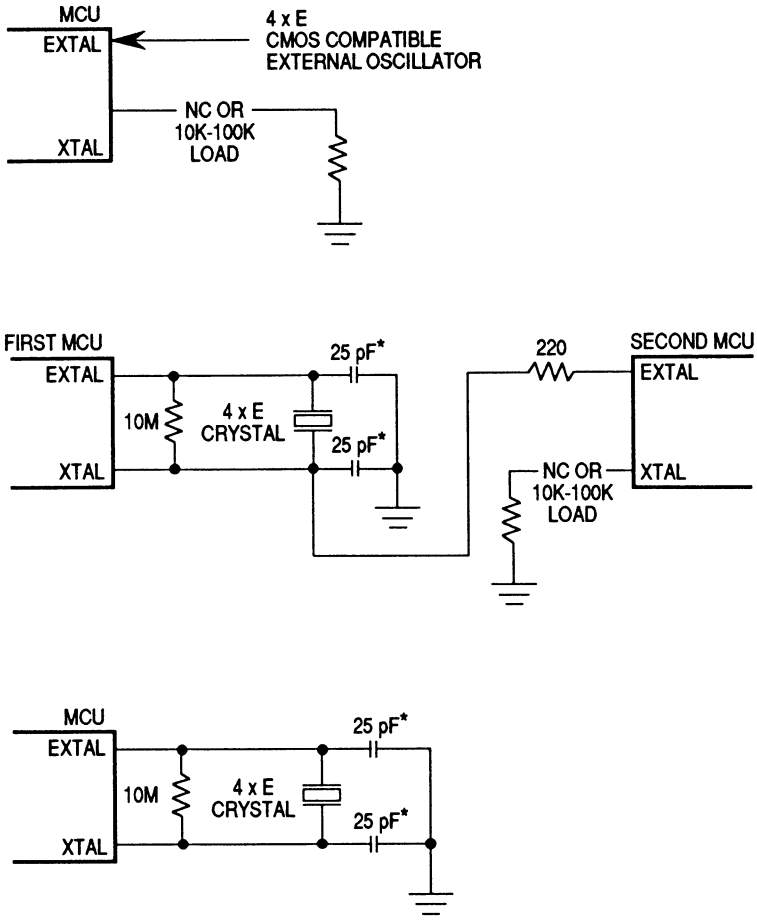
As reset transitions, these pins are used to latch the part into one of the four CPU controlled modes of operation. The $\overline{\text{LIR}}$ output can be used as an aid to debugging once reset is completed. The open-drain $\overline{\text{LIR}}$ pin goes to an active low during the first E-clock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

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

This pin provides two different functions, depending on the operating mode. In single-chip mode and bootstrap mode, the pin functions as input/output port D bit 7. In the expanded-multiplexed and test modes, the pin provides the read-write (R/$\overline{\text{W}}$) function. R/$\overline{\text{W}}$ is used to control the direction of transfers on the external data bus.

2.2.9 AS

This pin provides two different 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 the address strobe (AS) function. AS is used to demultiplex the address and data signals at port C.



* Values include all stray capacitances.

Figure 2-1. Oscillator Connections

being programmed and verified in PROG mode, this pin provides the chip enable ($\overline{CE}$) signal. An external resistor is required on $\overline{IRQ}$ to pull the pin to V_{DD} to prevent accidental programming of the PROM during reset.

2.2.6 $\overline{\text{XIRQ}}/V_{PP}$

The $\overline{\text{XIRQ}}$ pin 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 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 (PROG), this pin is used to supply EPROM or 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.

2.2.7 MODA/$\overline{\text{LIR}}$ and MODB/V_{STBY}

As reset transitions, these pins are used to latch the part into one of the four 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 RAM contents during power-down.

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

This pin provides two different functions, depending on the operating mode. In single-chip mode and bootstrap mode, the pin functions as input/output port D bit 7. In the expanded-multiplexed and test modes, the pin provides the read-write (R/$\overline{\text{W}}$) function. R/$\overline{\text{W}}$ is used to control the direction of transfers on the external data bus.

2.2.9 AS

This pin provides two different 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 the address strobe (AS) function. AS is used to demultiplex the address and data signals at port C.

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

In the 44-pin plastic leaded chip carrier (PLCC) package, there are 32 input/output (I/O) lines which are arranged into four 8-bit ports, 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 I/O, depending on the operating mode or peripheral functions selected. Note that ports B, C, and two bits of port D are available for I/O only in single-chip and boot modes. Table 2-2 is a summary of pin functions to operating modes, by line and by port.

Table 2-2. Port Signal Functions

Port	Bit	Single-Chip and Bootstrap Modes	Expanded-Nonmultiplexed and Special Test Modes
A	0	PA0/IC3	PA0/IC3
A	1	PA1/IC2	PA1/IC2
A	2	PA2/IC1	PA2/IC1
A	3	PA3/OC5/IC4 (and/or OC1)	PA3/OC5/IC4 (and/or OC1)
A	4*	PA4/OC4 (and/or OC1)	PA4/OC4 (and/or OC1)
A	5	PA5/OC3 (and/or OC1)	PA5/OC3 (and/or OC1)
A	6*	PA6/OC2 (and/or OC1)	PA6/OC2 (and/or OC1)
A	7	PA7/PAI (and/or OC1)	PA7/PAI (and/or OC1)
B	0	PB0	A8
B	1	PB1	A9
B	2	PB2	A10
B	3	PB3	A11
B	4	PB4	A12
B	5	PB5	A13
B	6	PB6	A14
B	7	PB7	A15
C	0	PC0	A0/D0
C	1	PC1	A1/D1
C	2	PC2	A2/D2
C	3	PC3	A3/D3
C	4	PC4	A4/D4
C	5	PC5	A5/D5
C	6	PC6	A6/D6
C	7	PC7	A7/D7
D	0	PD0/RxD	PD0/RxD
D	1	PD1/TxD	PD1/TxD
D	2	PD2/MISO	PD2/MISO
D	3	PD3/MOSI	PD3/MOSI
D	4	PD4/$\overline{\text{SCK}}$	PD4/$\overline{\text{SCK}}$
D	5	PD5/$\overline{\text{SS}}$	PD5/$\overline{\text{SS}}$
D	6	PD6	$\overline{\text{AS}}$
D	7	PD7	R/$\overline{\text{W}}$

*In the 40-pin package, pins A4 and A6 are not bonded. Their attendant I/O and output compare functions are not available externally. They may still be used as internal software timers.

SECTION 3
MEMORY AND CONTROL AND STATUS REGISTERS

This section describes the memory and the mapping of the control and status registers of the MC68HC711D3 MCU.

3.1 MEMORY

Figure 3-1 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 modes). 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 the expanded-multiplexed mode, the memory locations are basically the same as in the single chip, 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, except the interrupt vectors are at external memory locations.

3.2 CONTROL AND STATUS REGISTERS

Figure 3-2 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 4K 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.

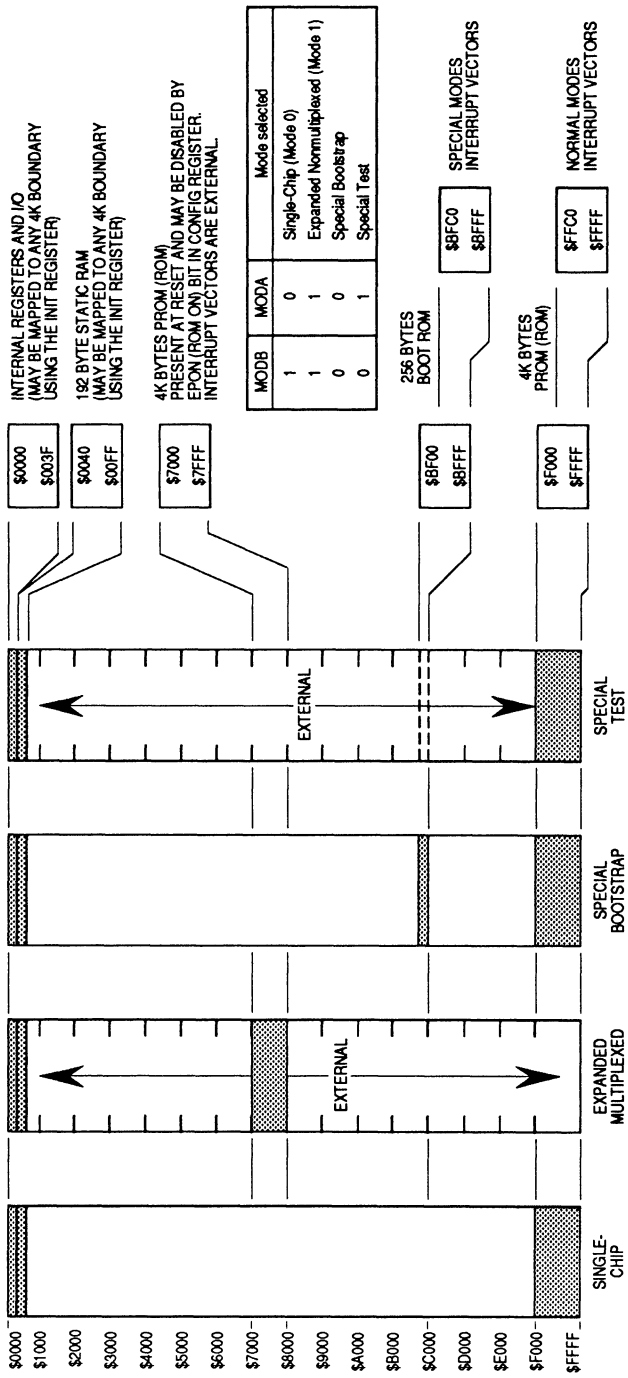


Figure 3-1. Memory Map

3.3 RAM AND I/O MAPPING REGISTER (INIT)

The INIT register 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.

	7	6	5	4	3	2	1	0	
$0**0**3D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT
RESET:	0	0	0	0	0	0	0	0	

RAM3–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 4K 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.

$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA
$0001									Rsvd.
$0002			CWOM						PIOC
$0003	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC
$0004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
$0005									Rsvd.
$0006	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
$0007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
$0008	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
$0009	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
$000A									Rsvd.
$000B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	CFORC
$000C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	OC1M
$000D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	OC1D
$000E	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TCNT
$000F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
$0010	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TIC1
$0011	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
$0012	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TIC2
$0013	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
$0014	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TIC3
$0015	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
$0016	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TOC1
$0017	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
$0018	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TOC2
$0019	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
$001A	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TOC3
$001B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
$001C	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TOC4
$001D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
$001E	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	TI4O5
$001F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
$0020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1
$0021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2
$0022	OC1I	OC2I	OC3I	OC4I	I4O5I	IC1I	IC2I	IC3I	TMSK1
$0023	OC1F	OC2F	OC3F	OC4F	I4O5F	IC1F	IC2F	IC3F	TFLG1

Figure 3-2. Control, Status, and I/O Registers (Sheet 1 of 2)

$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	TMSK2
$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	TFLG2
$0026	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0	PACTL
$0027	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	PACNT
$0028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
$0029	SPIF	WCOL	0	MODF	0	0	0	0	SPSR
$002A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SPDR
$002B	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD
$002C	R8	T8	0	M	WAKE	0	0	0	SCCR1
$002D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2
$002E	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	SCSR
$002F	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	SCDR
$0030									Rsvd.
to									
$0038									Rsvd.
$0039	0	0	IRQE	DLY	CME	0	CR1	CR0	OPTION
$003A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	COPRST
$003B	MBE	0	ELAT	EXCOL	EXROW	0	0	PGM	PPROG
$003C	RBOOT	SMOD	MDA	IRVNE	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO
$003D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT
$003E	TILOP	EPTST	OCCR	CBYP	DISR	FCM	FCOP	0	TEST1
$003F	0	0	0	0	0	NOCOP	EPON	0	CONFIG

Figure 3-2. Control, Status, and I/O Registers (Sheet 2 of 2)

SECTION 4
INPUT/OUTPUT PORTS

The MC68HC711D3 is equipped with four 8-bit I/O ports (A, B, C, and D). In the 40-pin version, ports A bits 4 and 6 are not bonded. Port functions are controlled by the particular mode of operation selected, as shown in Table 2-2 Port Signal Functions. In the single-chip and bootstrap modes, all the ports are configured as parallel I/O data ports. In expanded-multiplexed and test modes, ports B, C, and lines D6 (AS) and D7 (R/$\overline{W}$) are configured as a memory expansion bus, with ports B as the high-order address bus, port C as the multiplexed address and data bus, AS as the demultiplexing signal, and R/$\overline{W}$ 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 the SPI and SCI functions in addition to its bus direction control functions.

4.1 PORT A (PORTA)

In both the normal operating modes, port A can be configured for four timer input capture (IC) functions and three timer output compare (OC) functions, or for four OC and three IC functions, and either a pulse accumulator input (PAI) or a fifth OC function. Pins PA6 and PA4 are not bonded in the 40-pin DIP, and their OC output functions are unavailable, but their software interrupts are available.

	7	6	5	4	3	2	1	0	
$0000	PA7	PA6*	PA5	PA4*	PA3	PA2	PA1	PA0	PORTA
RESET:	0	0	0	0	0	0	0	0	
Alternate Pin Function:	PAI OC1	OC2 OC1	OC3 OC1	OC4 OC1	OC5/IC4 OC1	IC1 —	IC2 —	IC3 —	

*Pin is not bonded in the 40-pin version.

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.

4.2 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.

4.2.1 Port B Data Register (PORTB)

	7	6	5	4	3	2	1	0	
$0004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
RESET:	0	0	0	0	0	0	0	0	
Alternate Pin Function:	A15	A14	A13	A12	A11	A10	A9	A8	

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.

4.2.2 Port B Data Direction Register (DDRB)

	7	6	5	4	3	2	1	0	
$0006	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
RESET:	0	0	0	0	0	0	0	0	

1 = Corresponding Port B pin is configured as output.
0 = Corresponding Port B pin is configured for input only.

4.3 PORT C

Port C is an 8-bit, general-purpose I/O port with a data register (PORTC) and a data direction register (DDRC). In 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.

4.3.1 Port C Control Register (PIOC)

	7	6	5	4	3	2	1	0	
$0002	0	0	CWOM	0	0	0	0	0	PIOC
RESET:	0	0	0	0	0	0	0	0	

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

4.3.2 Port C Data Register (PORTC)

	7	6	5	4	3	2	1	0	
$0003	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC
RESET:	0	0	0	0	0	0	0	0	

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.

4.3.3 Port C Data Direction Register (DDRC)

	7	6	5	4	3	2	1	0	
$0007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
RESET:	0	0	0	0	0	0	0	0	

1 = Corresponding Port C pin is configured as output.
0 = Corresponding Port C pin is configured for input only.

4.4 PORT D

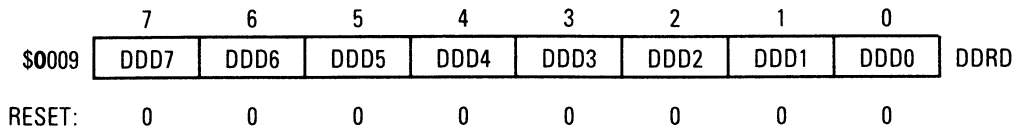
Port D is an 8-bit, general-purpose I/O port with a data register (PORTD) and a data direction register (DDRD). The eight port D bits (D7–D0) can be used for general-purpose I/O, for the SCI and SPI subsystems, or for bus data direction control.

4.4.1 Port D Data Register (PORTD)

	7	6	5	4	3	2	1	0	
$0008	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
RESET:	0	0	0	0	0	0	0	0	
Alternate Pin Function:	R/$\overline{W}$	AS	$\overline{SS}$	SCK	MOSI	MISO	TxD	RxD	
(PROG mode)	O7	O6	O5	O4	O3	O2	O1	O0	

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 for 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. In EPROM programming (PROG) mode this port is the data bus (O7–O0).

4.4.2 Port D Data Direction Register (DDRD)



	7	6	5	4	3	2	1	0	
$0009	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
RESET:	0	0	0	0	0	0	0	0	

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

1 = Configures the corresponding port D pin for output.
0 = Configures the corresponding port D pin for input only.

In expanded and test modes, bits 6 and 7 are dedicated to 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:

1 = Port D bit 5 is configured as a general-purpose output line.
0 = Port D bit 5 is an error-detect input to the SPI.

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

SECTION 5
RESETS, INTERRUPTS, AND LOW-POWER MODES

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

5.1 RESETS

The MCU can be reset in any of the following four ways:

- An active-low input to the $\overline{\text{RESET}}$ pin
- A power-on reset function
- A clock monitor failure
- 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.

5.1.1 $\overline{\text{RESET}}$ 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.

5.1.2 Power-on Reset (POR)

Power-on reset occurs when a positive transition is detected on V_{DD}. This reset is used strictly for power turnon 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.

5.1.3 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 **Section 7.3.1 Configuration Control Register (CONFIG)** for more information). Protected control bits (CR1 and CR0) in the configuration options register (OPTION) allow the user to select one of four COP timeout rates. Table 5-1 shows the relationship between CR1 and CR0 and the COP timeout period for various system clock frequencies.

Table 5-1. COP Timeout Periods

CR1	CR0	E÷2^{15} Divided By	XTAL=2^{23} Timeout −0/ +15.6 ms	XTAL=8.0 MHz Timeout −0/+16.4 ms	XTAL= 4.9152 MHz Timeout −0/+26.7 ms	XTAL=4.0 MHz Timeout −0/+32.8 ms	XTAL= 3.6864 MHz Timeout −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 s	1.049 s	1.707 s	2.1 s	2.276 s
		E=	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	921.6 kHz

The sequence for resetting the watchdog timer is as follows:

1. Write $55 to the COP reset register (COPRST).
2. Write $AA to the COPRST register.

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

5.1.4 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.

5.1.5 Configuration Options Register (OPTION)

The OPTION register 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 only be written once during the first 64 E-clock cycles after reset in normal modes (where the HPRIO register bit 6 (SMOD) is cleared). In special modes (where SMOD = 1), the bits can be written at any time. Bit 3 can be written at any time.

	7	6	5	4	3	2	1	0	
$0039	0	0	IRQE	DLY	CME	0	CR1	CR0	OPTION
RESET:	0	0	0	1	0	0	0	0	

Bits 7, 6, 2

Not used; always read zero.

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 since 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 STOP recovery.

CME — Clock Monitor Enable

1 = Clock monitor circuit is enabled.

0 = Clock monitor circuit is disabled.

CR1, CR0 — COP Timer Rate Selects

The COP system is driven by a constant frequency of E divided by 2 to the 15th power. These two bits specify an additional divide-by value to arrive at the COP timeout rate. These bits are cleared during reset and can only

be written once during the first 64 E-clock cycles after reset in normal modes. The value of these bits is shown in the following table:

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

5.2 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 nonmaskable. Fifteen of the interrupts can be masked using the I bit of the CCR. All the on-chip (hardware) interrupts are individually maskable by local control bits. The software interrupt is nonmaskable. The external input to the $\overline{\text{XIRQ}}$ pin is considered a nonmaskable 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 nonmaskable interrupt.

Table 5-2 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 5-1 shows the interrupt stacking order.

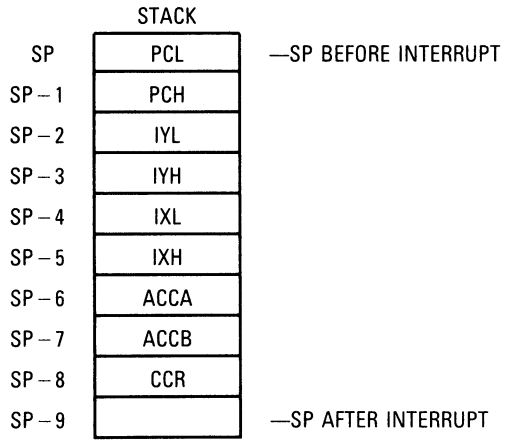


Figure 5-1. Interrupt Stacking Order

Table 5-2. Interrupt Vector Masks and Assignments

Vector Address	Interrupt Source	Condition Code Register Mask	Local Mask
FFC0,FFC1	Reserved	—	—
*	*		
*	*		
FFD4,FFD5	Reserved	—	—
FFD6,FFD7	SCI Serial System	I Bit	—
	Receive Data Register Full	—	RIE
	Receive Overrun	—	RIE
	Idle Line Detect	—	ILIE
	Transmit Data Register Empty	—	TIE
	Transmit Complete	—	TCIE
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
FFDE,FFDF	Timer Overflow	I Bit	TOI
FFE0,FFE1	Timer IC4/OC5	I Bit	I4O5I
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 (Psuedo-Nonmaskable)	X Bit	None
FFF6,FFF7	SWI	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

5.2.1 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 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.

5.2.2 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 requested to the illegal opcode vector. The illegal opcode vector should never be left uninitialized.

5.2.3 Real-Time Interrupt

The real-time interrupt 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 be $E \div 2^{13}$, $E \div 2^{14}$, $E \div 2^{15}$, or $E \div 2^{16}$. See PACTL, TMSK2, and TFLG2 register descriptions in the programamble timer section for control and status bit information.

5.2.4 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 $\overline{\text{XIRQ}}$. After minimum system initialization, software may clear the X bit by a TAP instruction, thus enabling $\overline{\text{XIRQ}}$ interrupts. Thereafter, software cannot set the X bit. So, an $\overline{\text{XIRQ}}$ interrupt is effectively a nonmaskable 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 nonmasked. In the interrupt priority logic, the $\overline{\text{XIRQ}}$ 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.

5.2.5 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 of the CCR and have the following fixed priority relationship:

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 only be written while the I-bit-related interrupts are inhibited (I bit of CCR is logic one). 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.

Figures 5-2, 5-3, and 5-4 illustrate the interrupt process as it relates to normal processing. Figure 5-2 shows how the CPU begins from a reset, and how interrupt detection relates to normal opcode fetches. Figure 5-3 is an expansion of a block in Figure 5-2 and shows how interrupt priority is resolved. Figure 5-4 is an expansion of the SCI interrupt block in Figure 5-3 and shows the resolution of interrupt sources within the SCI subsystem.

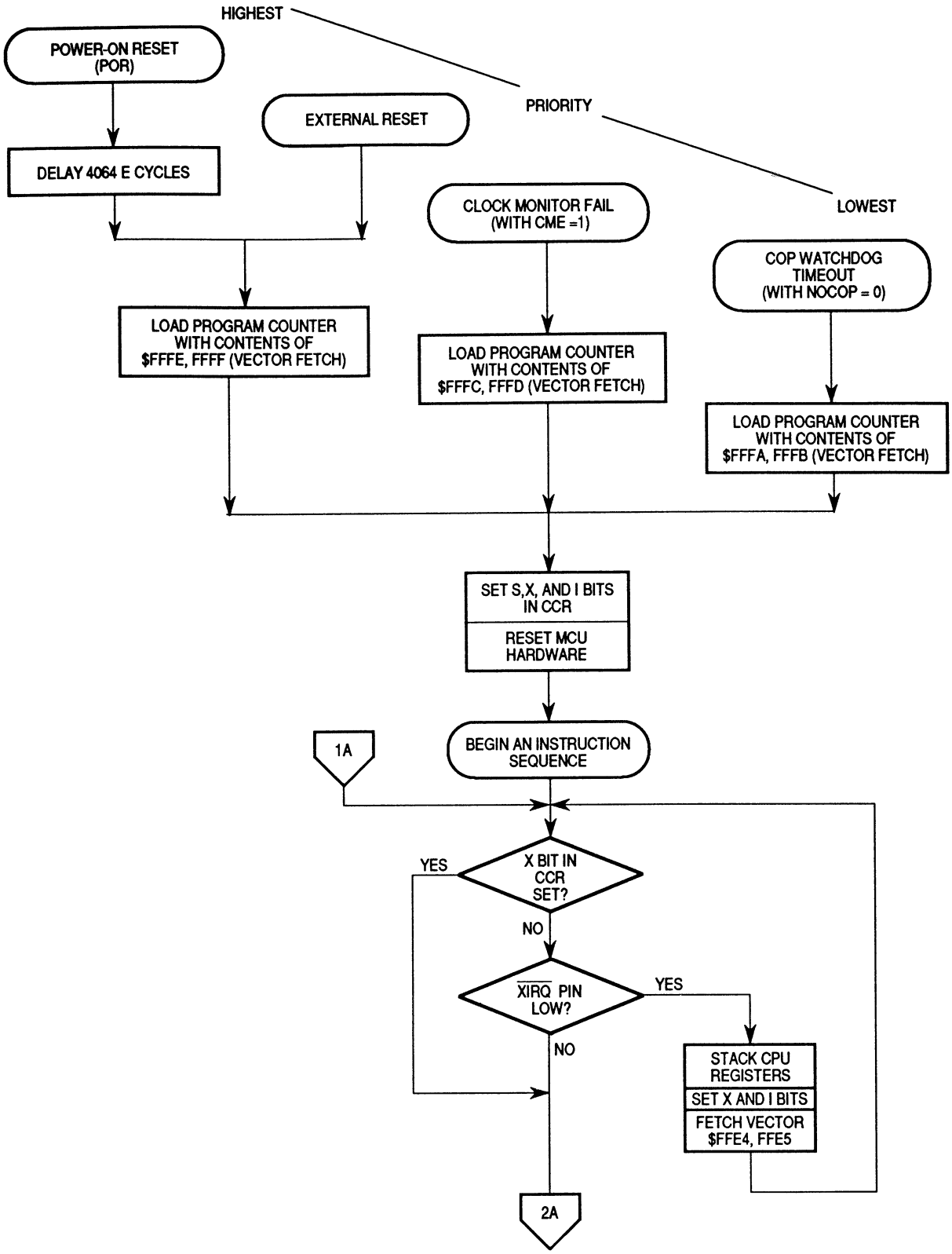


Figure 5-2. Processing Flow Out of Resets (Sheet 1 of 2)

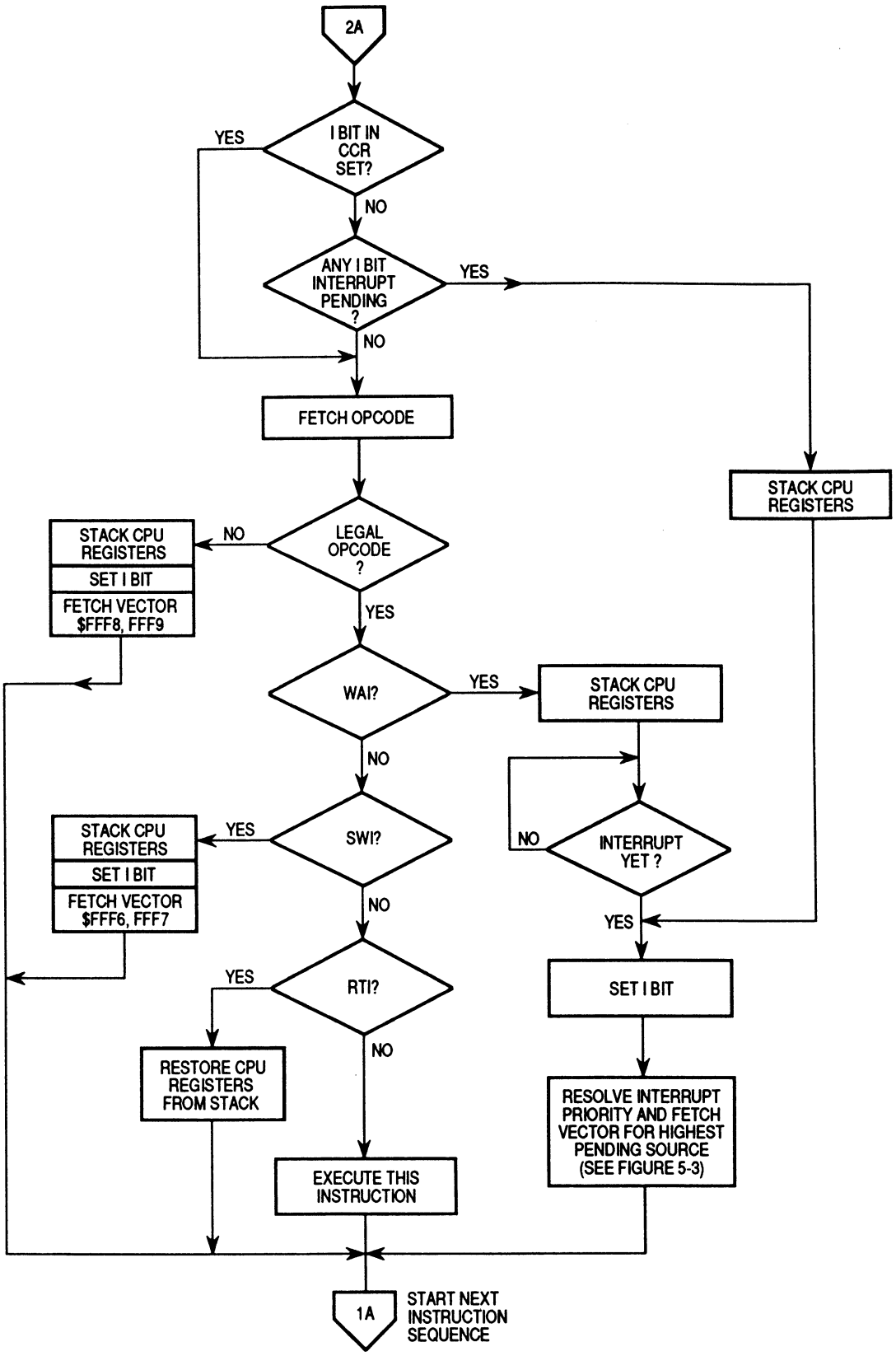


Figure 5-2. Processing Flow Out of Resets (Sheet 2 of 2)

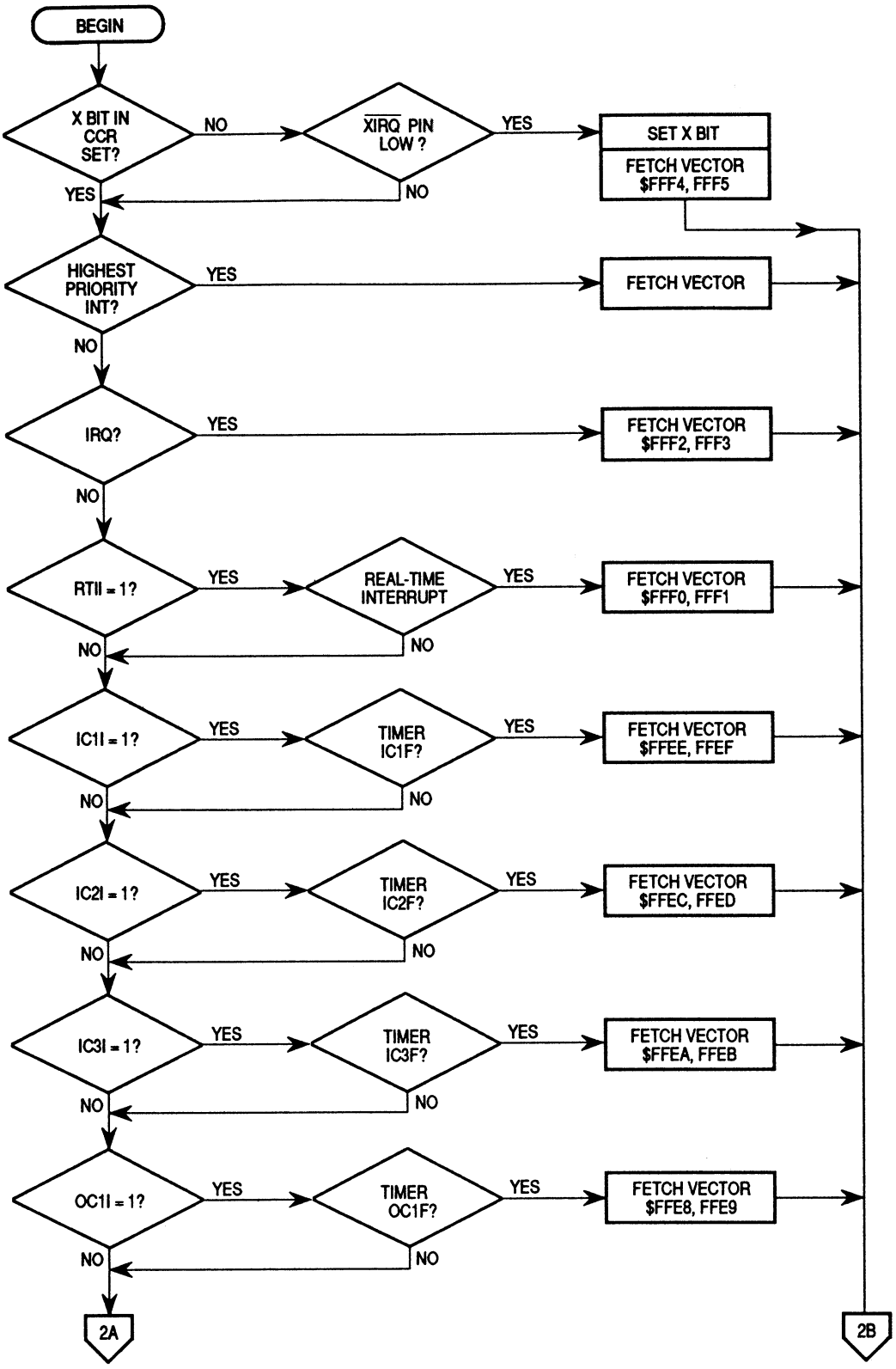


Figure 5-3. Interrupt Priority Resolution (Sheet 1 of 2)

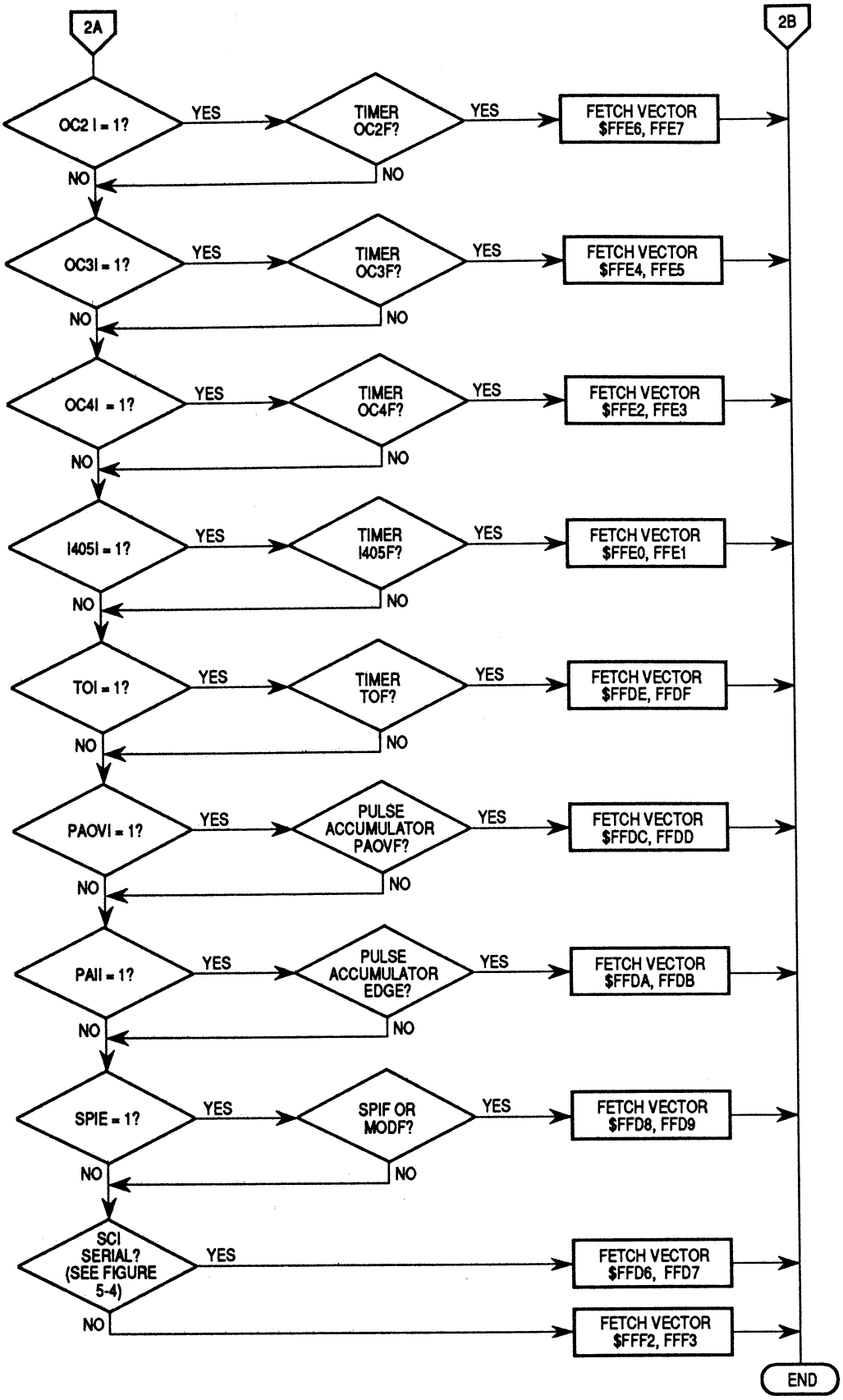


Figure 5-3. Interrupt Priority Resolution (Sheet 2 of 2)

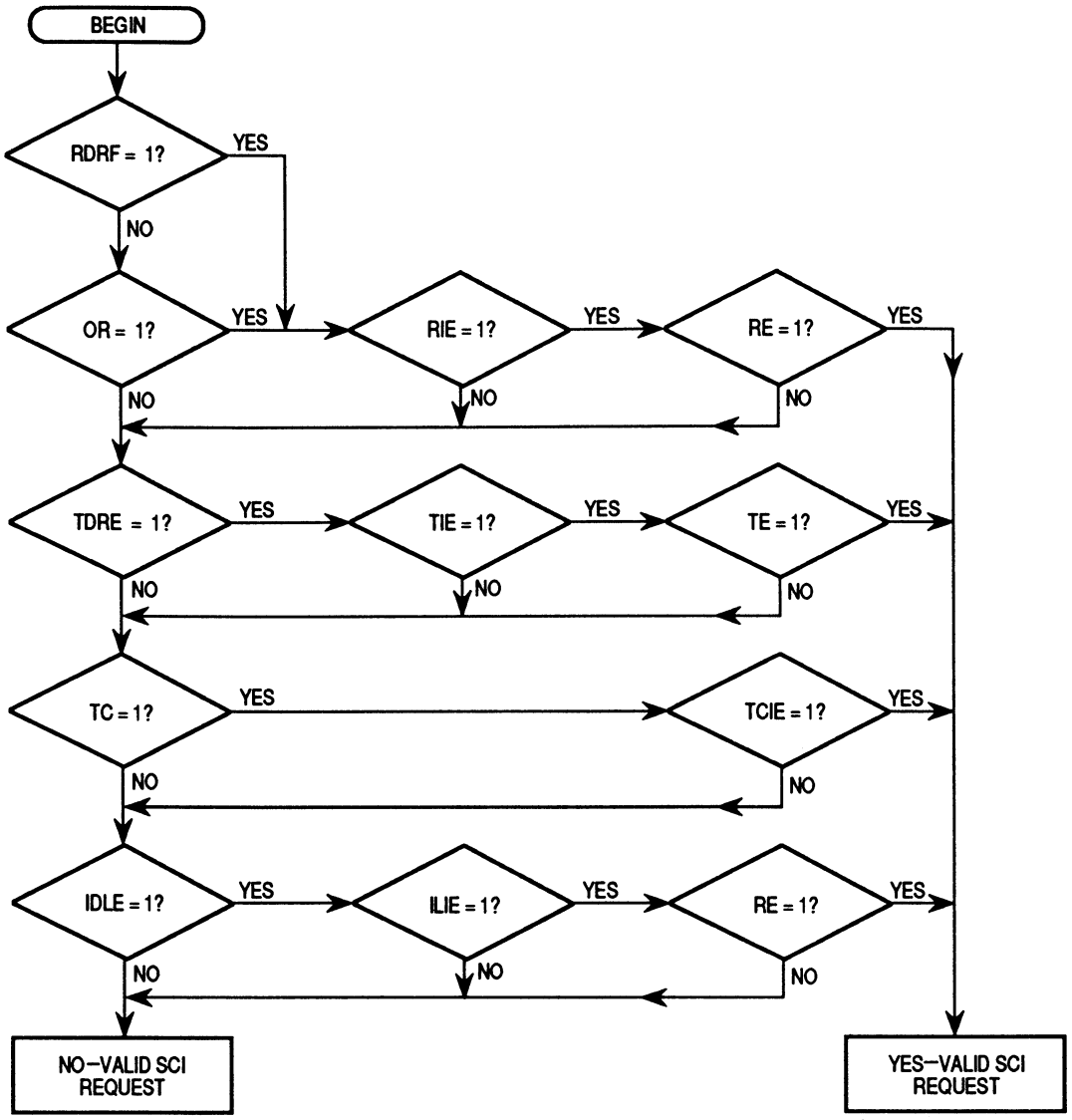


Figure 5-4. Interrupt Source Resolution within SCI

5.2.6 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.

	7	6	5	4	3	2	1	0	
$003C	RBOOT	SMOD	MDA	IRVNE	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO
RESET:	*	*	*	*	0	1	0	1	

* = The reset condition of bits 7, 6, 5, and 4 depends on the mode selected at power-up initialization.

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 to only in special modes. It cannot be written to a one 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 the following table:

Input Pins		Mode Description	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 Enable/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.

In expanded and test modes, this bit determines whether the internal read visibility is on or off:

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.

In single-chip and bootstrap modes, IRVNE determines whether the E clock is driven out or forced low:

1 = E pin is driven low.

0 = E clock is 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 the following table:

PSEL3	PSEL2	PSEL1	PSEL0	Interrupt Source Promoted
0	0	0	0	Timer Overflow
0	0	0	1	Pulse Accumulator Overflow
0	0	1	0	Pulse Accumulator Input Edge
0	0	1	1	SPI Serial Transfer Complete
0	1	0	0	SCI Serial System
0	1	0	1	Reserved (Default to $\overline{\text{IRQ}}$)
0	1	1	0	$\overline{\text{IRQ}}$ (External Pin)
0	1	1	1	Real-Time Interrupt
1	0	0	0	Timer Input Capture 1
1	0	0	1	Timer Input Capture 2
1	0	1	0	Timer Input Capture 3
1	0	1	1	Timer Output Compare 1
1	1	0	0	Timer Output Compare 2
1	1	0	1	Timer Output Compare 3
1	1	1	0	Timer Output Compare 4
1	1	1	1	Timer IC4/OC5

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.

5.3 LOW POWER-CONSUMPTION MODES

The MC68HC11 Family of MCUs 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. The following paragraphs describe these two low power-consumption modes.

5.3.1 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.

5.3.2 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, SPI, or SCI that were or were not active when the wait mode was entered.

SECTION 6
PROGRAMMABLE TIMER

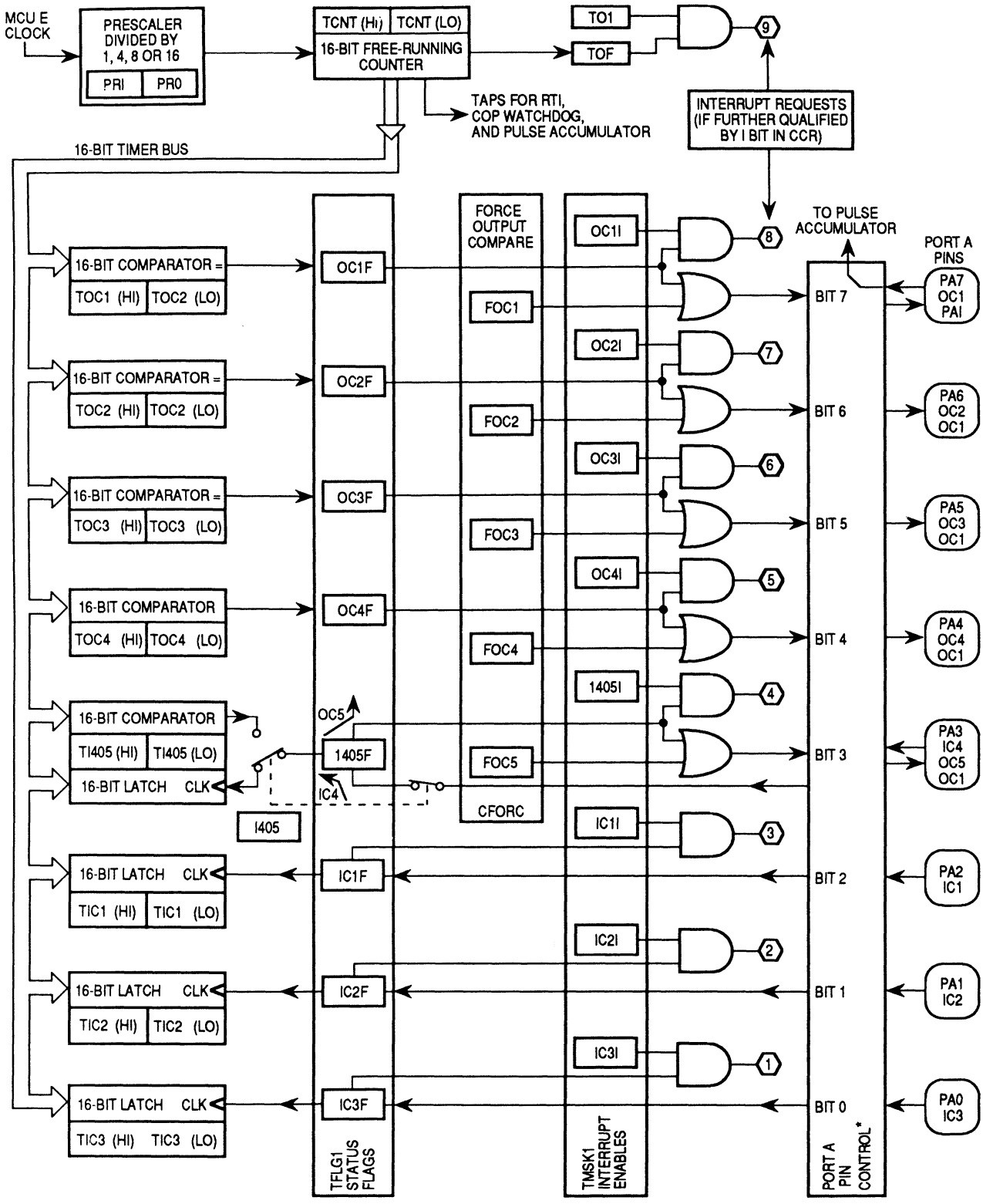
The timer system uses a "time-of-day" approach in that all timing functions are related to a single, 16-bit, free-running counter. The free-running counter is clocked by the output of a programmable prescaler. The prescaler is clocked by the E clock and divides this clock by 1, 4, 8, or 16. The prescaler control bits, found in the TMSK2 register, can only be written once during the first 64 cycles after a reset. The free-running counter (TCNT register) can be read by software at any time without affecting its value since it is clocked and read on the opposite half-cycle of the E clock. The counter is cleared on reset, is a read-only register, and repeats every 65,536 counts. When the count changes from $FFFF to $0000, the timer overflow flag (TOF) bit is set in the timer interrupt flag register 2 (TFLG2). The overflow flag also generates an internal interrupt if the timer overflow interrupt enable (TOI) bit of the timer interrupt mask register 2 (TMSK2) is set. The timer has three input capture and four output compare function registers plus an additional register that can perform either function under software control.

Figure 6-1 is a block diagram of the timer, whose functions and registers of the timer are explained in the following paragraphs.

6.1 INPUT CAPTURE (IC) FUNCTION

There are three, regular, 16-bit, read-only input capture registers (TIC1, TIC2, and TIC3) and a register that can serve as either the fourth IC register or the fifth output compare register (TI4O5). These registers are not initialized by reset. Each IC register is used to latch the value of the free-running counter when a selected transition at an external pin is detected. External devices provide the inputs to IC4–IC1 on the PA3–PA0 pins, and an interrupt can be generated when an input capture edge is detected. The time of detection can be read from the appropriate timer register as part of the interrupt routine.

The control and status bits to 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, the user must clear the DDRA3 bit of the PACTL register. Note that this bit is cleared out of reset. Additionally, to enable PA3 as the fourth input capture, the I4/O5 bit in the PACTL register must be



*Port A pin actions controlled by PACTL, OC1M, OC1D, TCTL1, and TCTL2 registers.

Figure 6-1. Timer Block Diagram

set. Otherwise, PA3 is configured as a fifth output compare out of reset, with bit I4/O5 being cleared. If the DDRA bit DDA3 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. When the TI4O5 register is acting as IC4, it cannot be written to.

6.2 TIMER CONTROL REGISTER 2 (TCTL2)

	7	6	5	4	3	2	1	0	
$0021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2
RESET:	0	0	0	0	0	0	0	0	

EDGxB and EDGxA — Input Capture Edge Control

There are four pairs of these bits. Each pair is cleared to zero by reset and is encoded to configure the input capture edge detector circuit for IC4–IC1. Coding is as follows:

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 Rising or Falling Edge

NOTE

IC4 functions only if the I4/O5 bit of the PACTL register is set.

6.3 OUTPUT COMPARE (OC) FUNCTION

There are four 16-bit read/write output compare registers (TOC1, TOC2, TOC3, and TOC4) and a fifth register (TI4O5) that can function 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 the timer interrupt flag register 1 (TFLG1). An interrupt is then generated, provided that particular interrupt is enabled in the timer interrupt mask register 1 (TMSK1). In addition to the interrupt, a specified action may be initiated at a timer output pin(s). 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 or not the OCxF flag in the TFLG1 register was previously clear.

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 to be taken when a match is found is controlled by two 5-bit registers: 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.

Upon reset, bit I4/O5 of PACTL is configured as OC5. The OC5 function is then enabled when bits OM5 and OL5 of TCTL1 are appropriately set. Note that although the DDRA3 bit of the PACTL register configures port A bit 3 as an input out of reset, the PA3 pin with an enabled output compare is forced to be an output. The DDRA3 bit does not change, however. The DDRA3 bits revert to I/O control once the associated output compare is disabled.

The control and status bits to implement the output compare functions are contained in the PACTL, CFORC, OC1M, OC1D, TCTL1, TMSK1, and TFLG1 registers.

6.3.1 Timer Compare Force Register (CFORC)

This write-only register is used to force early output compare actions. This compare force function is not recommended for use with the output toggle function, because a normal compare occurring immediately before or after the force can result in an undesirable operation.

	7	6	5	4	3	2	1	0	
$000B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	CFORC
RESET:	0	0	0	0	0	0	0	0	

FOC5–FOC1 — Force Output Compare x Action

1 = Causes the action programmed for output compare x to take place, except the output compare x flag (OCxF) of TFLG1 is not set.

0 = Not affected.

Bits 2–0 — Not implemented; always read zero.

6.3.2 Output Compare 1 Mask Register (OC1M)

This register is used with OC1 to specify the bits of port A that are affected as the result of a successful OC1 compare. The bits of the OC1M register correspond bit for bit with lines of port A (lines 7–3, only).

	7	6	5	4	3	2	1	0	
$000C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	OC1M
RESET:	0	0	0	0	0	0	0	0	

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

Bits 2–0 — Not implemented; always read zero.

6.3.3 Output Compare Data Register (OC1D)

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

	7	6	5	4	3	2	1	0	
$000D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	OC1D
RESET:	0	0	0	0	0	0	0	0	

Bits 2–0 — Not implemented; always read zero.

6.3.4 Timer Count Register (TCNT)

	15	14	13	12	11	10	9	8	
$000E									MSB TCNT
$000F									LSB
	7	6	5	4	3	2	1	0	
RESET:	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

This 16-bit read-only register contains the prescaled value of the 16-bit timer. A full counter read should first address the most significant byte (MSB). 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 will return the full 16-bit state of the counter at the time of the MSB read cycle.

6.3.5 Timer Control Register 1 (TCTL1)

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

	7	6	5	4	3	2	1	0	
$0020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1
RESET:	0	0	0	0	0	0	0	0	

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

These control bit pairs are encoded to specify the action taken as the result of a successful OCx compare. OC5 functions only if the I4/O5 bit in the PACTL register is clear. Coding is as follows:

OMx	OLx	Action Taken Upon Successful Compare
0	0	Timer Disconnected from Output Pin Logic
0	1	Toggle OCx Output Line
1	0	Clear OCx Output Line (to Zero)
1	1	Set OCx Output Line (to One)

6.3.6 Timer Interrupt Mask Register 1 (TMSK1)

This 8-bit register is used to enable or inhibit the timer input capture and output compare interrupts.

	7	6	5	4	3	2	1	0	
$0022	OC1I	OC2I	OC3I	OC4I	I4O5I	IC1I	IC2I	IC3I	TMSK1
RESET:	0	0	0	0	0	0	0	0	

OCxI — Output Compare x Interrupt
1 = Interrupt sequence requested if bit OCxF of TFLG1 is set.
0 = Interrupt inhibited.

ICxI — Input Capture x Interrupt
1 = Interrupt sequence requested if bit ICxF of TFLG1 is set.
0 = Interrupt inhibited.

I4O5I — Input Capture 4 or Output Capture 5 Interrupt

When the I4/O5 bit of PACTL is set, the I4O5I bit acts as the IC4 interrupt bit. When I4/O5 is cleared, the I4O5I bit acts as the OC5 interrupt control bit.

1 = Interrupt sequence is requested.
0 = Interrupt is inhibited.

6.3.7 Timer Interrupt Flag Register 1 (TFLG1)

This register is used to indicate the occurrence of timer system events. With the TMSK1 register, TFLG1 allows the timer subsystem to operate in either a polled or interrupt driven system. Each bit of TFLG1 has a corresponding bit in TMSK1 in the same bit position.

	7	6	5	4	3	2	1	0	
$0023	OC1F	OC2F	OC3F	OC4F	I4O5F	IC1F	IC2F	IC3F	TFLG1
RESET:	0	0	0	0	0	0	0	0	

OCxF — Output Compare x Flag

These bits are set each time the timer counter matches the output compare register x value.

I4O5F — Input Capture 4/Output Compare 5 Flag

This bit functions as the flag for either input capture 4 or output compare 5. The flag is set by hardware as IC4 or OC5, depending on which function was enabled by bit I4/O5 of PACTL at the time of interrupt.

ICxF — Input Capture x Flag

These bits are set each time a selected active edge is detected on the ICx input line.

These flag bits are cleared by writing a one to the corresponding bit position.

6.3.8 Timer Interrupt Mask Register 2 (TMSK2)

This register is used to control whether or not a hardware interrupt sequence is requested as the result of a status bit being set in the timer interrupt flag register (TFLG1). The two timer prescaler bits are also included in this register.

	7	6	5	4	3	2	1	0	
$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	TMSK2
RESET:	0	0	0	0	0	0	0	0	

TOI — Timer Overflow Interrupt Enable
1 = Interrupt requested when bit TOF of TFLG2 is set.
0 = Timer overflow interrupt is disabled.

RTII — Real-Time Interrupt Enable
1 = Interrupt requested when bit RTIF of TFLG2 is set.
0 = Real-time interrupt is disabled.

PAOVI — Pulse Accumulator Overflow Interrupt Enable
1 = Interrupt requested when bit PAOVF of TFLG2 is set.
0 = Pulse accumulator overflow interrupt disabled.

PAII — Pulse Accumulator Interrupt Enable
1 = Interrupt requested when bit PAIF of TFLG2 is set.
0 = Pulse accumulator interrupt disabled.

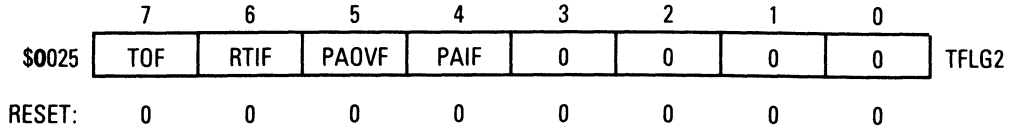
Bits 3–2 — Not implemented; always read zero.

PR1 and PR0 — Timer Prescaler Select
These bits are used to select the prescaler divide-by ratio. They can be written once in the first 64 E-clock cycles out of reset in normal modes. They can be written at any time in special modes. The value of these bits is decoded as follows:

PR1	PR0	Divide By
0	0	1
0	1	4
1	0	8
1	1	16

6.3.9 Timer Interrupt Flag Register 2 (TFLG2)

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



	7	6	5	4	3	2	1	0	
$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	TFLG2
RESET:	0	0	0	0	0	0	0	0	

TOF — Timer Overflow

This bit is set each time the 16-bit free-running counter advances from a value of $FFFF–$0000.

RTIF — Real-Time Interrupt Flag

This bit is set at each rising edge of the selected tap point.

PAOVF — Pulse Accumulator Overflow Interrupt Flag

This bit is set when the count in the pulse accumulator rolls over from $FF–$00.

PAIF — Pulse Accumulator Input-Edge Interrupt Flag

This bit is set when an active edge is detected on the PAI input pin. In the event mode, the event edge triggers PAIF. In gated time accumulator mode, the trailing edge of the gate signal at the PAI input triggers PAIF.

Bits 3–0 — Not implemented; always read zero.

These flag bits are cleared by writing a one to the corresponding bit position.

6.4 PULSE ACCUMULATOR

The pulse accumulator is an 8-bit counter that can operate in either of two modes, depending on the state of a control bit in the PACTL register. These modes are the event counting mode and the gated time accumulation mode. 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 the gated time accumulation mode, a free-running E-clock/64 signal drives the 8-bit counter, but only while the external PAI input pin is activated.

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

6.4.1 Pulse Accumulator Control Register (PACTL)

Four of this register's bits control an 8-bit pulse accumulator system. Another bit enables either the output compare 5 function or the input capture 4 function, while two other bits select the rate for the real-time interrupt system.

	7	6	5	4	3	2	1	0	
$0026	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR2	PACTL
RESET:	0	0	0	0	0	0	0	0	

DDRA7 — Data Direction Control for Port A bit 7
1 = PA7 is an output.
0 = PA7 is an input only.

PAEN — Pulse Accumulator System Enable.
1 = Pulse accumulator is on.
0 = Pulse accumulator is off. (No flags can be set; counter is stopped.)

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

PEDGE — Pulse Accumulator Edge Control
This bit has different meanings, depending on the state of the PAMOD bit, as shown in the following table:

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

DDRA3 — Data Direction Control for Port A bit 3
1 = PA3 is an output.
0 = PA3 is an input only.

I4/O5 — Configure TI4O5 Register for IC or OC

1 = IC4 function is enabled.

0 = OC5 function is enabled.

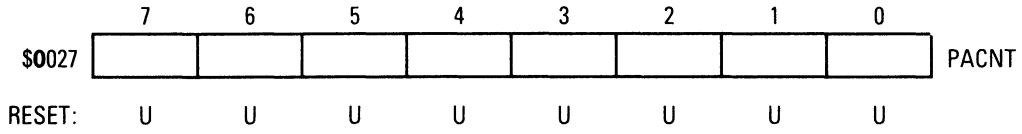
RTR1–RTR0 — Real-Time Interrupt (RTI) Rate

The value of these pins, decoded, selects one of four rates for the real-time periodic interrupt circuits. After reset, a full RTI period elapses before the first RTI interrupt. RTI rates are shown in Table 6-1.

Table 6-1. RTI Rate at Various Crystal Frequencies

RTR1	RTR0	Divide E By	XTAL = 2^{23}	XTAL = 8.0 MHz	XTAL = 4.9152 MHz	XTAL = 4.0 MHz	XTAL = 3.6864 MHz
0	0	2^{13}	3.91 ms	4.10 ms	6.67 ms	8.19 ms	8.89 ms
0	1	2^{14}	7.81 ms	8.19 ms	13.33 ms	16.38 ms	17.78 ms
1	0	2^{15}	15.62 ms	16.38 ms	26.67 ms	32.77 ms	35.56 ms
1	1	2^{16}	31.25 ms	32.77 ms	53.33 ms	65.54 ms	71.11 ms
		E =	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	921.6 kHz

6.4.2 Pulse Accumulator Count Register (PACNT)



This 8-bit read/write register contains the count of external input events at the PAI input or the accumulated count while the PAI input is active in gated time accumulation mode.

SECTION 7
PROGRAMMABLE READ-ONLY MEMORY (PROM)

The MC68HC711D3 has 4K bytes of PROM, either EPROM or 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.

7.1 PROM PROGRAMMING

There are two separate methods of programming the on-chip PROM of an MC68HC711D3. In the first method (PROG mode), certain pins of the MCU are made to emulate the pins of a 27256-type EPROM, and may be programmed as a standard EPROM. In the second method (MCU mode), the PROM is programmed through the MCU in the bootstrap or test modes.

7.2 PROGRAMMING PROM USING PROG MODE

In order to make the MC68HC711D3 emulate a standard 27256-type EPROM, the MCU must be shifted into a new programming mode called PROG, and a footprint conversion must be made using an adapter. The MCU enters PROG when pins MODA, MODB, and $\overline{\text{RESET}}$ are held low. No clocks are necessary. Table 7-1 shows which MCU pins emulate EPROM pins in this mode:

Table 7-1. 27256 Emulation

MCU Pins	EPROM Pins
PC7–PC0	A7–A0
PB4–PB0	A12–A8
PB6–PB5	GND
NC	A14–A13
PB7	$\overline{OE}$
PD7–PD0	07–O0
$\overline{XIRQ}$	V_{PP}
$\overline{IRQ}$	$\overline{CE}$
V_{DD}	V_{CC}
V_{SS}	GND
MODA	GND
MODB	GND
$\overline{RESET}$	GND
PA3–PA0	GND
PA6–PA4	NC
PA7	GND
EXTAL	GND
XTAL	NC
E	NC

NOTE: MCU addresses A14–A13 and EPROM addresses A14–A13 are not connected. A12 is connected, but treated as a "don't care" to allow a potential upgrade to 8K EPROM. The 4K PROM repeats eight times in the 32K programming space of the 27256.

7.2.1 External Read and Verify (PROG Mode)

To read and verify the OTPROM or EPROM, the MCU is placed in PROG mode. $\overline{OE}$ (MCU pin PB7) is held high to deselect the PROM. $\overline{CE}$ may be high or low at this point. An address is presented. After a setup time, $\overline{OE}$ and $\overline{CE}$ (if not already low) are brought low to allow the address to decode the EPROM location. After access time is completed, data becomes valid on the data bus and is read. $\overline{OE}$ is brought high before presenting the next address.

When verifying only, it is acceptable to leave both $\overline{OE}$ and $\overline{CE}$ low and present addresses. The data is presented out of the data bus after a setup time. In PROG mode, reads must be sequential. That is, A0 must toggle between reads.

7.2.2 External Programming (PROG Mode)

To program the OTPROM or EPROM, the MCU is placed in PROG mode. $\overline{CE}$ is held high while the V_{PP} ($\overline{XIRQ}$) pin is brought to programming voltage level. $\overline{OE}$ is held high throughout. The address and data to be programmed are presented. After a setup time, $\overline{CE}$ is brought low. This state is held for the duration of programming time. $\overline{CE}$ must be brought high before the address and data are changed to program the next location. The correlation between MCU and PROG mode pins is shown in Figure 7-1.

The erased state of each byte of PROM is $FF. Each bit of the EPROM or OTPROM is programmable to 0 on a per byte basis. Once an EPROM bit is programmed to 0, it may not be reprogrammed to 1 without ultraviolet (UV) erasure. OTPROM cannot be erased or reprogrammed. Bytes may be programmed in any order.

7.3 PROM PROGRAMMING USING THE MCU

Programming the OTPROM or EPROM through the MCU is allowed only in special test and bootstrap modes. With the EPON bit of the CONFIG register set, the various control bits of the PPROG register are manipulated to program the PROM locations. The erased state of a PROM byte is $FF.

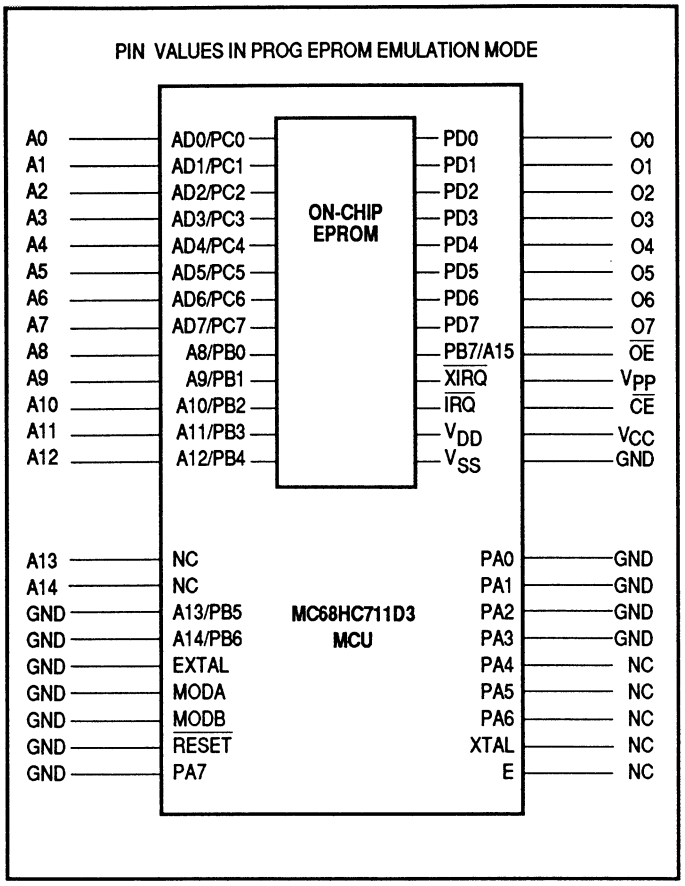


Figure 7-1. MC68HC711D3 Block Diagram in PROG Mode

7.3.1 Configuration Control Register (CONFIG)

The configuration control register controls the presence of OTPROM or EPROM in the memory map and enables the COP watchdog system.

This register is writable only once in expanded and single-chip modes (SMOD = 0). In these modes, 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, 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 one to the MDA bit in the HPRIO register.

	7	6	5	4	3	2	1	0	
$003F	0	0	0	0	0	NOCOP	EPON	0	CONFIG
RESET:	0	0	0	0	0	*	*	0	

Bits 7–3 and 0 — Not implemented; always read zero.

NOCOP — Computer Operating Properly System Disable

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.

EPON — PROM Enable

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.

7.3.2 PROM Programming Control Register (PPROG)

This register 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.

	7	6	5	4	3	2	1	0	
$003B	MBE	0	ELAT	EXCOL	EXROW	0	0	PGM	PPROG
RESET:	0	0	0	0	0	0	0	0	

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

Bit 6 — Not implemented; always reads zero.

ELAT — EPROM (OTPROM) Latch Control
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
This bit is reserved for testing.

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

Bits 2, 1 — Not implemented; always read zero.

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

7.3.3 PROM Programming Sequence

Before programming, ensure that V_{PP} voltage is available on the $\overline{XIRQ}$ pin.

Note that $\overline{XIRQ}$ must not become active (be brought low) during programming. This state would disrupt or corrupt the programming of the OTPROM or EPROM. The proper sequence for programming the PROM through the MCU (with EPON bit set in CONFIG) is as follows:

Step	Comments
1. Write $20 to PPROG. 2. Write data to PROM. 3. Write $21 to PPROG. 4. Delay 2 to 4 ms. 5. Write $20 to PPROG.	Set ELAT bit (PGM = 0) to enable PROM latches. Set PGM bit (ELAT = 1) to enable PROM high voltage. Turn off high voltage to PROM array.
6. Repeat steps 2–5, as needed.	
7. Write $00 to PPROG.	Return to read mode.

7.3.4 Protecting the PROM

Some precautions are necessary in order to protect the OTPROM or EPROM. For systems which do not make use of the on-chip programming capability, but program the MC68HC711D3 as a standard EPROM, voltage on the $\overline{XIRQ}$/V_{PP} pin should never be permitted to be greater than V_{DD}. The PROM cannot be programmed or corrupted without high voltage on the $\overline{XIRQ}$ pin.

Systems that use the on-chip programming feature require that V_{PP} voltage be available. Regardless of which mode the MCU is in, if the part is operating with V_{PP} voltage present on the $\overline{XIRQ}$/V_{PP} pin, then the $\overline{IRQ}$/$\overline{CE}$ pin must be pulled to a high level in reset.

The pin configuration necessary to place the MCU in bootstrap mode (MODA, MODB, and $\overline{RESET}$ pins being low) is also the configuration that places the MCU in the PROG state. If, at the same time, V_{PP} is present and $\overline{IRQ}$ is low, then the PROM is being programmed. A pullup resistor on the $\overline{IRQ}$ line prevents this situation by ensuring the $\overline{IRQ}$ pin is high leaving reset.

PROM should be programmed at room temperature only. An opaque label should be placed over the quartz window of EPROM (windowed) packages after programming.

7.3.5 Erasing the PROM

OTPROM MCU devices are shipped in an erased state. Once programmed, they cannot be erased. Electrical erasing procedures cannot be performed on either OTPROM (nonwindowed packages) or EPROM devices (windowed packages).

EPROM devices can be erased by exposure to a high-intensity UV light with a wavelength of 2537 Å. The recommended integrated dosage (UV intensity $\times$ exposure time) is 15 Ws/cm^2. UV lamps should be used without shortwave filters, and the EPROM device should be positioned about one inch from the UV lamp. The erased state of an EPROM location is $FF.

SECTION 8
SERIAL COMMUNICATIONS INTERFACE

The serial communications interface (SCI) allows the MCU to be efficiently interfaced with peripheral devices that require an asynchronous serial data format. The SCI uses a standard nonreturn-to-zero (NRZ) format with a variety of baud rates derived from the crystal clock circuit. Interfacing is accomplished using port D pins. PD0 is used for receive data (RxD), and PD1 for transmit data (TxD). The baud-rate generation circuit contains a programmable prescaler and divider clocked by the E clock. Figure 8-1 shows a block diagram of the SCI.

8.1 DATA FORMAT

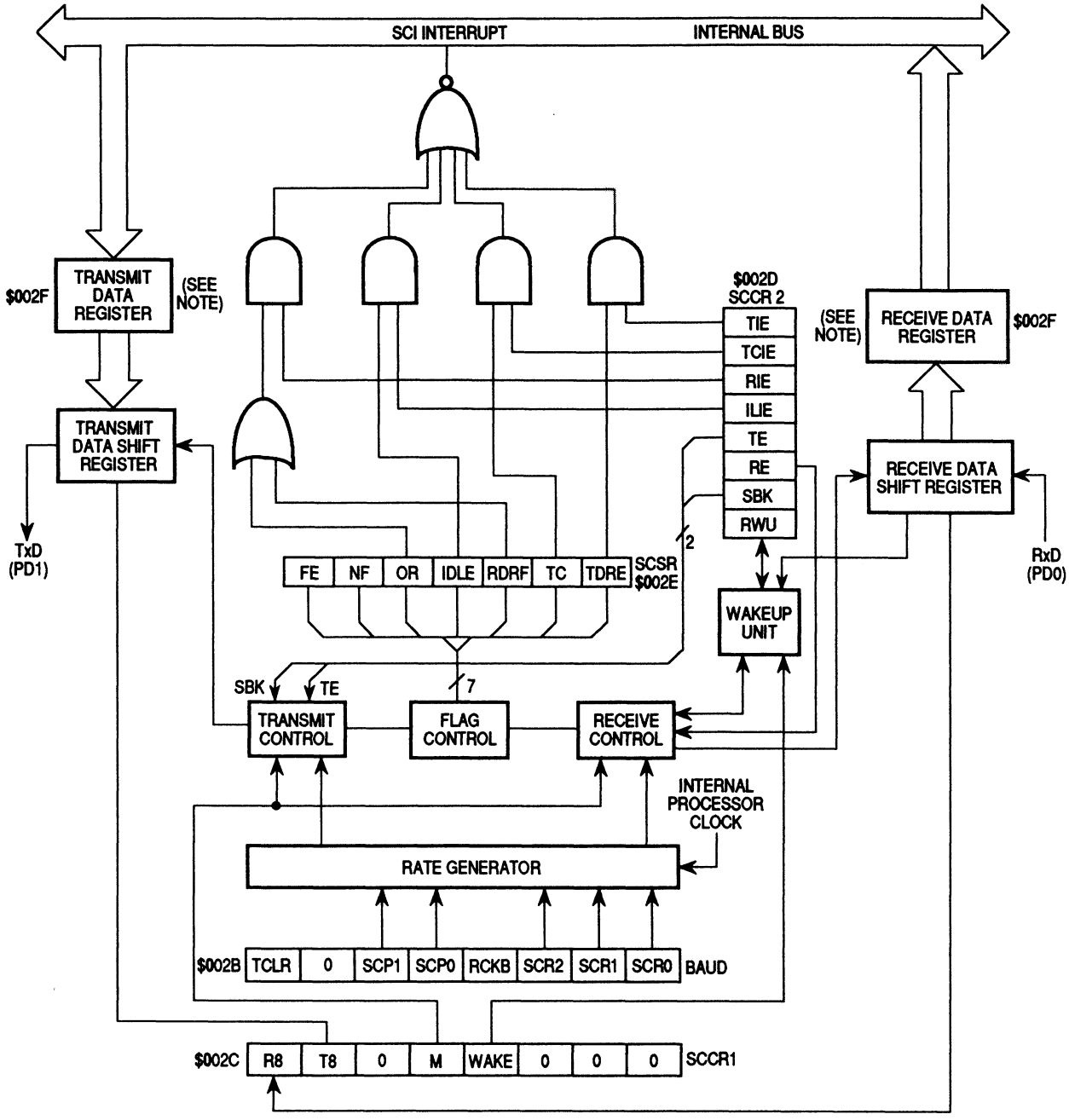
The serial data format requires the following:

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

Selection of the word length is controlled by the M bit of the serial communications interface control register 1 (SCCR1).

8.2 TRANSMIT OPERATION

The SCI transmitter includes a parallel transmit data register, called the SCDR, and a serial shift register that puts data from the SCDR into serial form. The contents of the serial shift register can be written only through the SCDR. This double-buffered system allows a character to be shifted out serially while another character is waiting in the SCDR to be transferred into the serial shift



NOTE: The serial communications data register (SCDR) is controlled by the internal R/$\overline{W}$ signal. It is the transmit data register when written and received data register when read.

Figure 8-1. SCI Block Diagram

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 the serial communication control register (SCCR2) is set.

8.3 RECEIVE OPERATION

In receive operations, the transmit sequence is reversed. Data is received in the serial shift register and is transferred to a parallel receive data register (the SCDR) as a complete word. This double-buffered system allows a character to be shifted in serially while another character is already in the SCDR. An advanced data recovery scheme is used to distinguish 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.

8.4 WAKEUP FEATURE

The wakeup feature reduces SCI service overhead in multiple receiver systems. Software for each receiver evaluates the first character(s) of each message. If the message is intended for a different receiver, the SCI can be placed in a sleep mode, stopping the rest of the message from generating requests for service. Whenever a new message begins, logic causes the sleeping receivers to awaken and evaluate the initial character(s) of the new message.

Two methods of wakeup are available: idle-line wakeup or address-mark wakeup. In idle-line wakeup, a sleeping receiver awakens as soon as the RxD line becomes idle. In the address-mark wakeup, a one in the most significant bit (MSB) of a character is used to indicate that the message is an address that wakes up all sleeping receivers.

8.5 SCI REGISTERS

The following paragraphs describe the operations of the five addressable registers used in the SCI.

8.5.1 Serial Communications Data Register (SCDR)

The SCDR register 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. Figure 8-1 shows the SCDR as two separate registers whose single address is $**0**02F.

8.5.2 Serial Communications Control Register 1 (SCCR1)

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

	7	6	5	4	3	2	1	0	
$002C	R8	T8	0	M	WAKE	0	0	0	SCCR1
RESET:	U	U	0	0	0	0	0	0	

R8 — Receive Data Bit 8

This bit functions as the ninth serial data bit received when the SCI system is configured for nine data bit operation (the M bit is set).

T8 — Transmit Data Bit 8

This bit functions as the ninth data bit to be transmitted when the SCI system is configured for nine data bit operation (the M bit is set).

Bits 5, 2–0 — Not implemented; always read zero.

M — SCI Character Length

1 = One start bit, nine data bits, one stop bit — system now configured for nine data bit operation.

0 = One start bit, eight data bits, one stop bit — system now configured for eight data bit operation.

WAKE — Wakeup Method Select

1 = Address Mark

0 = Idle Line

8.5.3 Serial Communications Control Register 2 (SCCR2)

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

	7	6	5	4	3	2	1	0	
$002D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2
RESET:	0	0	0	0	0	0	0	0	

TIE — Transmit Interrupt Enable
- 1 = SCI interrupt requested (if TDRE bit of SCSR is set)
- 0 = TDRE interrupts disabled

TCIE — Transmit Complete Interrupt Enable
- 1 = SCI interrupt requested (if TC bit of SCSR is set)
- 0 = TC interrupts disabled

RIE — Receive Interrupt Enable
- 1 = SCI interrupt requested (if RDRF or OR bit of SCSR is set)
- 0 = Receive data register full (RDRF) or overrun error (OR) interrupts disabled

ILIE — Idle-Line Interrupt Enable
- 1 = SCI interrupt requested (if IDLE bit of SCSR is set)
- 0 = Idle-line detect (IDLE) interrupts disabled

TE — Transmit Enable
- 1 = Transmit shift register output applied to the TxD line
- 0 = PD1 pin reverts to general-purpose I/O as soon as current transmitter activity finishes.

RE — Receive Enable
- 1 = Receiver enabled
- 0 = Receiver disabled and RDRF, IDLE, OR, NF, and FE interrupts inhibited

RWU — Receiver Wakeup

When set by the user's software, this bit puts the receiver to sleep and enables the wakeup function. If the WAKE bit of SCCR1 is logic zero, RWU is cleared by the SCI logic after receiving 10 (M bit of SCCR1 = 0) or 11 (M = 1) consecutive ones. If WAKE is logic one, RWU is cleared by the SCI logic after receiving a data word whose MSB is set.

SBK — Send Break

If this bit is toggled (set and cleared), the transmitter sends 10 or 11 consecutive zeros, depending on the setting of the M bit of SCCR1, then reverts to idle or sending data. If SBK remains set, the transmitter continually sends whole frames of zeros in groups of 10 or 11 until SBK is cleared.

8.5.4 Serial Communications Status Register (SCSR)

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

	7	6	5	4	3	2	1	0	
$002E	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	SCSR
RESET:	1	1	0	0	0	0	0	0	

TDRE — Transmit Data Register Empty

This bit is set automatically as the contents of SCDR are transferred to the serial shift register in a transmit operation. New data is not transmitted unless the SCSR register is read before the data is writing to SCDR. TDRE is cleared by a read of SCSR followed by a write to SCDR.

TC — Transmit Complete

This bit is set automatically when all data frame, preamble, or break condition transmissions are complete. When TC is set, the serial line goes idle (continuous mark). TC is cleared by a read of SCSR followed by a write to SCDR.

RDRF — Receive Data Register Full

This bit is set automatically when a character is transferred from the serial shift register to SCDR in a receive operation. RDRF is cleared by a read of SCSR followed by a read of SCDR.

IDLE — Idle-Line Detect

This bit is inhibited while the RWU bit of SCCR2 is set. IDLE is set automatically when the receiver serial input detects an idle line. This bit is cleared by a read of SCSR followed by a read of SCDR.

OR — Overrun Error

This bit is automatically set when a new character is ready to transfer from the receiver shift register to the SCDR and the SCDR is already full (RDRF bit set). Data transfer is inhibited until this bit is cleared. OR is cleared by a read of SCSR followed by a read of SCDR.

NF — Noise Flag

This bit is automatically set if there is noise on any of the received bits, including the start and stop bits. This bit is not set until the RDRF flag is set. It is not set in the case of an overrun. NF is cleared by a read of SCSR followed by a write to SCDR.

FE — Framing Error

This bit is automatically set when no stop bit is detected in the received data character. The FE bit is set at the same time as the RDRF flag is set. If the byte received causes both framing and overrun errors, only the overrun error is recognized. This flag inhibits further transfers until it is cleared. FE is cleared by a read of SCSR followed by a read of SCDR.

Bit 0 — Not implemented; always reads zero.

8.5.5 Baud Rate Register (BAUD)

This register is used to select different baud rates that can be used as the rate control for the receiver and transmitter.

	7	6	5	4	3	2	1	0	
$002B	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD
RESET:	0	0	0	0	0	U	U	U	

TCLR — Clear Baud Rate Counters (Test)

This bit is used to clear the baud rate counter chain during factory testing. TCLR is zero and cannot be set while the MCU is in normal operating modes.

Bit 6 — Not implemented; always reads zero.

SCP1 and SCP0 — SCI Baud Rate Prescaler Selects

These bits control a prescaler whose output is the input for a second divider that is controlled by the SCR2–SCR0 bits. Refer to Table 8-1.

RCKB — SCI Baud Rate Clock Check (Test)

This bit is used during factory testing to enable the exclusive OR of the receiver clock and transmitter clock to be driven out of the TxD pin. RCKB is zero and cannot be set while in normal operating modes.

SCR2–SCR0 — SCI Baud Rate Selects

These bits select the baud rate for both the transmitter and the receiver. The prescaler output selected by SCP1 and SCP0 is further divided by the setting of these bits. Refer to Table 8-2.

Table 8-1. Prescaler Highest Baud Rate Frequency Output

SCP Bit		Clock* Divided By	Crystal Frequency (MHz)				
1	0		8.3886	8.0	4.9152	4.0	3.6864
0	0	1	131.072 K Baud	125.000 K Baud	76.80 K Baud	62.50 K Baud	57.60 K Baud
0	1	3	43.690 K Baud	41.666 K Baud	25.60 K Baud	20.833 K Baud	19.20 K Baud
1	0	4	32.768 K Baud	31.250 K Baud	19.20 K Baud	15.625 K Baud	14.40 K Baud
1	1	13	10.082 K Baud	9600 Baud	5.907 K Baud	4800 Baud	4430 Baud

*The internal processor clock

Table 8-2. Transmit Baud Rate Output for a Given Prescaler Output

SCR Bit			Divided By	Representative Highest Prescaler Baud Rate Output				
2	1	0		131.072 K Baud	32.768 K Baud	76.80 K Baud	19.20 K Baud	9600 Baud
0	0	0	1	131.072 K Baud	32.768 K Baud	76.80 K Baud	19.20 K Baud	9600 Baud
0	0	1	2	65.536 K Baud	16.384 K Baud	38.40 K Baud	9600 Baud	4800 Baud
0	1	0	4	32.768 K Baud	8.192 K Baud	19.20 K Buad	4800 Baud	2400 Baud
0	1	1	8	16.384 K Baud	4.096 K Baud	9600 Baud	2400 Baud	1200 Baud
1	0	0	16	8.192 K Baud	2.048 K Baud	4800 Baud	1200 Baud	600 Baud
1	0	1	32	4.096 K Baud	1.024 K Baud	2400 Baud	600 Baud	300 Baud
1	1	0	64	2.048 K Baud	512 Baud	1200 Baud	300 Baud	150 Baud
1	1	1	128	1.024 K Baud	256 Baud	600 Baud	150 Baud	75 Baud

SECTION 9
SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is a high-speed synchronous serial I/O system. The SPI can be used for simple I/O expansion or for allowing several MCUs to be interconnected in a multimaster configuration. Clock phase and polarity are software programmable to allow direct compatibility with a large number of peripheral devices. The SPI system can be configured as either a master or a slave.

Four basic signal lines are associated with the SPI system: the master-out slave-in (MOSI), the master-in slave-out (MISO), the serial clock (SCK), and the slave select ($\overline{SS}$) pins. The SPI signals are assigned to port D bits 5–2 as shown in Figure 9-1. Note that SPI outputs must have the corresponding bits set in the port D data direction register (DDRD). However, any SPI input line is forced to be an input, regardless of the DDRD bit settings.

A series of eight SCK clock cycles are generated to synchronize data transfer. When a master device transmits data to a slave device via the MOSI line, the slave responds by sending data to the master device via the MISO line. This exchange implies full-duplex transmission with both data out and data in synchronized to the same clock signal. The byte transmitted is replaced by the byte received, thereby eliminating the need for separate transmit-empty and receiver-full status bits. A single status bit (SPIF) is used to signify that the I/O operation has been completed. Figure 9-1 is a block diagram of the SPI.

The SPI is double buffered on read, but not on write. If a write is attempted during data transfer, the transfer is uninterrupted, and the write is unsuccessful. This condition will cause the write collision (WCOL) bit of the SPSR register to be set. After a data byte is shifted, the SPIF flag of SPSR is set.

In the master mode, the SCK pin is an output. Depending on the CPOL bit of the SPCR register, SCK idles high or low until data is written to the shift register. Once data is written, eight clocks are generated to shift the eight bits of data. Then SCK goes idle again.

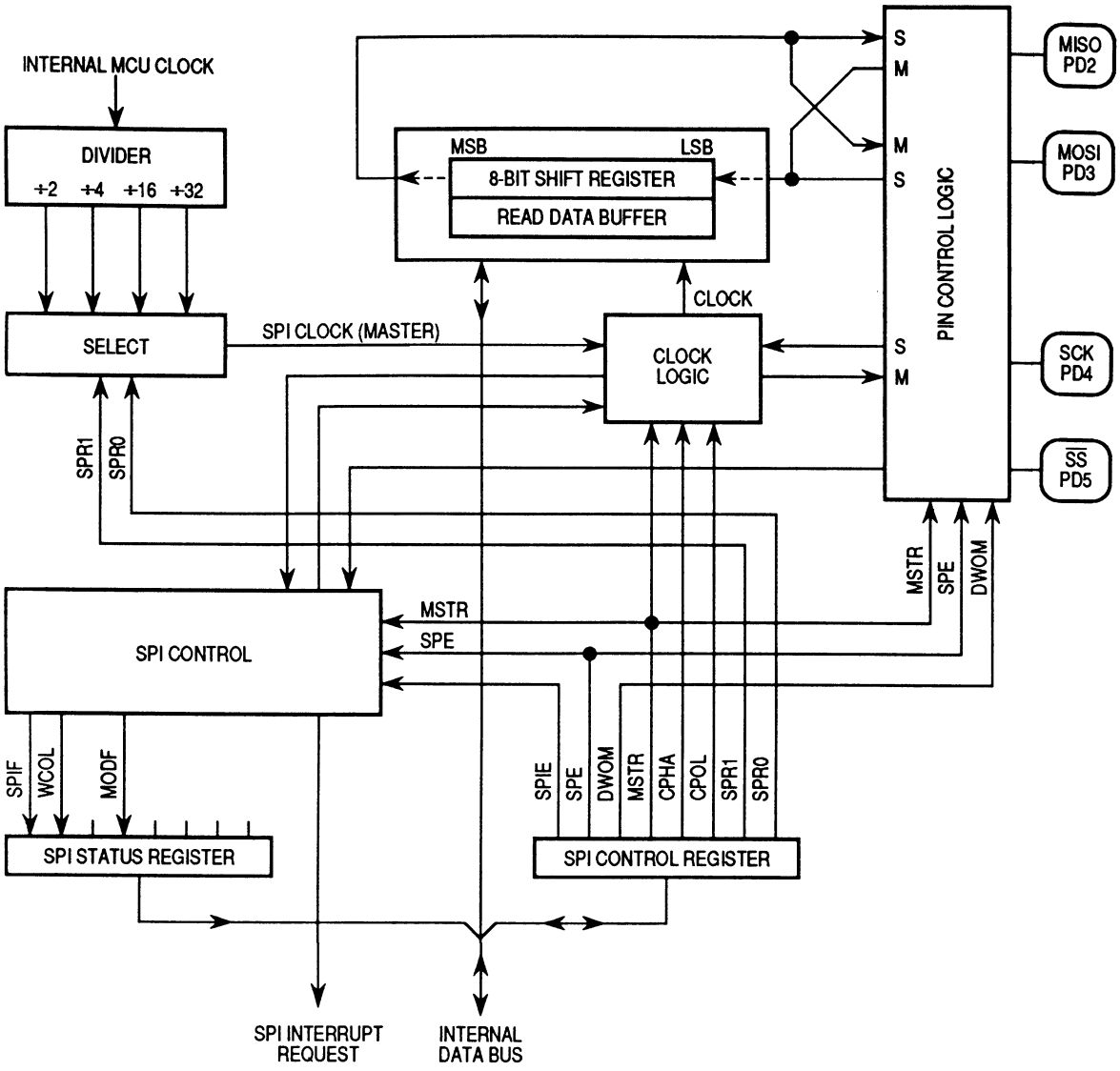


Figure 9-1. SPI Block Diagram

In the slave mode, the slave start logic receives a logic low at the $\overline{SS}$ pin and a clock input at the SCK pin. Thus, the slave is synchronized with the master. Data from the master is received serially at the slave MOSI line and loads the 8-bit shift register. Once the 8-bit shift register is loaded, its data is parallel transferred to the read buffer. During a write cycle, data is written into the shift register. The slave then waits for a clock train from the master to shift the data out on the slave's MISO line. Figure 9-2 illustrates the MOSI, MISO, SCK, and $\overline{SS}$ master-slave interconnections.

NOTE

After a byte transfer in slave mode, SCK must be in its inactive state for at least two E-clock cycles before the next byte transfer.

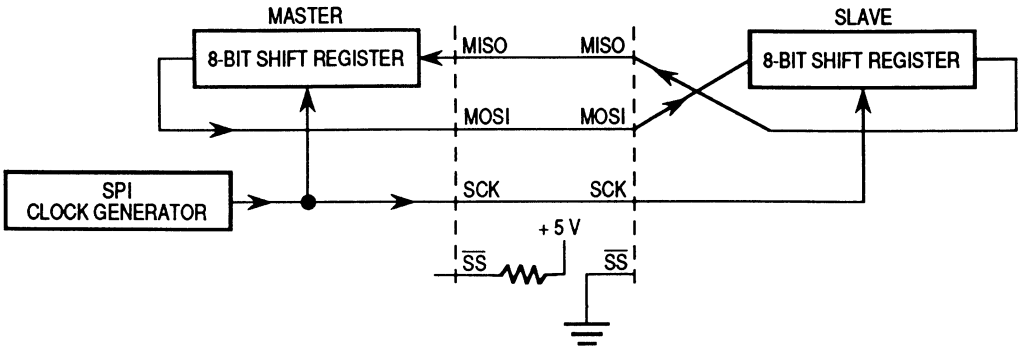


Figure 9-2. SPI Master-Slave Interconnections

9.1 SPI REGISTERS

There are three registers in the SPI that provide control, status, and data storage functions. These registers are described in the following paragraphs.

9.1.1 Serial Peripheral Control Register (SPCR)

	7	6	5	4	3	2	1	0	
$0028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
RESET:	0	0	0	0	0	0	U	U	

SPIE — Serial Peripheral Interrupt Enable
1 = SPI interrupt is requested (if bit SPIF or MODF of SPSR = 1).
0 = SPI (SPIF) interrupts are disabled.

SPE — Serial Peripheral System Enable
1 = SPI system is on.
0 = SPI system is off.

DWOM — Port D Wired-OR Mode Option
This bit affects port D bits 5–0 together.
1 = Port D bits 5–0 outputs act as open-drain outputs.
0 = Port D outputs are normal CMOS outputs.

MSTR — Master Mode Select
1 = Master mode is selected.
0 = Slave mode is selected.

CPOL — Clock Polarity

This bit selects the polarity of the SCK clock. It is used with the CPHA control bit to produce the desired clock-data relationship between master and slave. (Refer to Figure 9-3.)

1 = SCK line idles high.
0 = SCK line idles low.

CPHA — Clock Phase

This bit selects one of two fundamentally different clock protocols when considered with the CPOL setting. (Refer to Figure 9-3.)

1 = Data is sampled midway through the SCK period. $\overline{SS}$ to the slave can be left low between transfers.
0 = Transfer begins when $\overline{SS}$ goes low. Data is sampled on the initial edge of SCK. $\overline{SS}$ to the slave must be negated and then asserted between bytes.

SPR1 and SPR0 — SPI Clock Rate Selects

These two bits select one of four baud rates to be used as SCK if the SPI is set as a master. They have no effect in the slave mode. Their value decodes as follows:

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

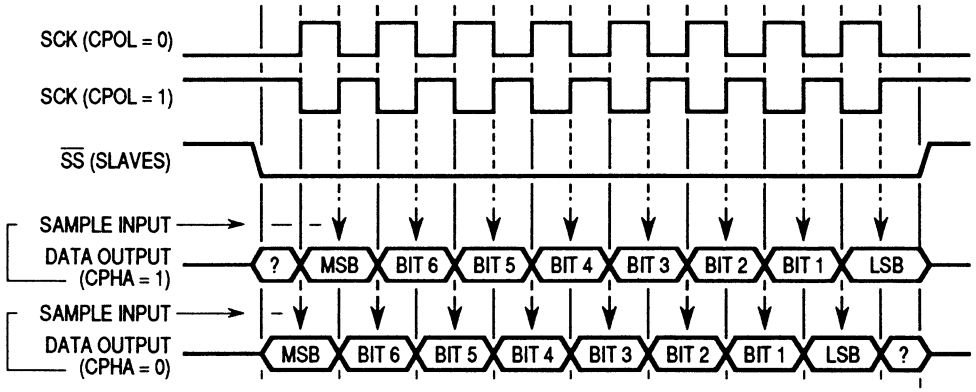
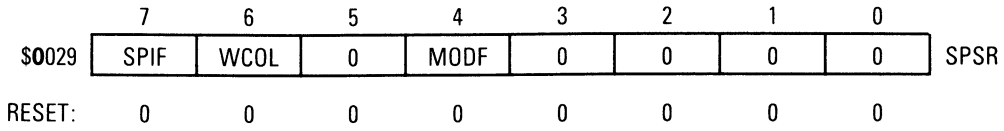


Figure 9-3. SPI Data Clock Timing Diagram

9.1.2 Serial Peripheral Status Register (SPSR)



	7	6	5	4	3	2	1	0	
$0029	SPIF	WCOL	0	MODF	0	0	0	0	SPSR
RESET:	0	0	0	0	0	0	0	0	

SPIF — SPI Transfer Complete Flag

This flag is automatically set when data transfer is complete between processor and external device. The flag is cleared by a read of SPSR followed by a read or write of SPDR.

WCOL — Write Collision

This bit is set automatically when an attempt is made to write to the SPI data register while data is being transferred. The bit is cleared by a read of SPSR followed by a read or write of SPDR.

Bits 5, 3–0 — Not implemented; always read zero.

MODF — Mode Fault

This bit indicates the possibility of a multimaster conflict for system control and allows a proper exit from system operation to a reset or default system state. MODF is only set when a master device has its $\overline{SS}$ pin pulled low. The SPI system is then effectively terminated by the following actions:

1. The DDRD bits are cleared to disable SPI outputs.
2. The MSTR bit is cleared to reconfigure the SPI as slave.
3. The SPE bit is cleared to disable the SPI.
4. An SPI interrupt is generated if SPIF is set.

This bit is cleared by a read of SPSR followed by a write to SPCR. The SPE and MSTR control bits can be restored to their original state by user software once the MODF bit has been cleared. The DDRD bits must be restored after a mode fault.

9.1.3 Serial Peripheral Data I/O Register (SPDR)

Located at address $**0**02A,this register is used to transmit and receive data on the serial bus. A write to this register by a master initiates transmission/reception of another byte. A slave writes data to this register for later transmission to a master. When transmission is complete, the SPIF status bit is set in the SPSR register of both the master and slave device. When a read is performed on the SPDR, a buffer is actually being read. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated, or an overrun condition will exist. In case of an overrun, the byte causing the overrun is lost.

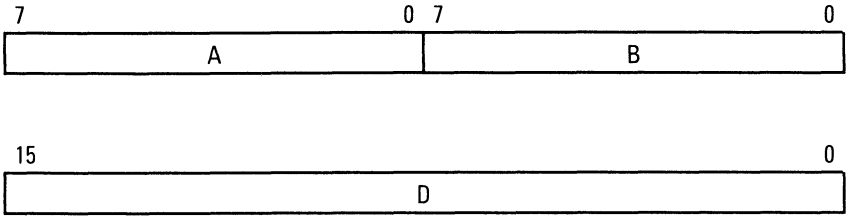
SECTION 10
PROGRAMMING INFORMATION

10.1 PROGRAMMING MODEL

The M68HC11 Family of MCUs has eight central processing unit (CPU) registers available to the programmer. Each register is explained in the following paragraphs.

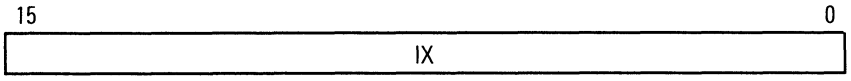
10.1.1 Accumulators (A, B, and D)

Accumulators A and B are general-purpose 8-bit registers used to hold operands and the results of arithmetic calculations or data manipulations. These two accumulators are treated as a single double-byte accumulator (D accumulator) for some instructions.



10.1.2 Index Register X (IX)

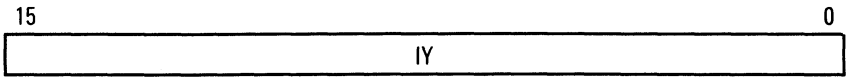
The IX register is a 16-bit register used for the indexed addressing mode. It provides a 16-bit value that can be added to an 8-bit offset provided in an instruction to create an effective address. The IX register can also be used as either a counter or temporary storage area.



10.1.3 Index Register Y (IY)

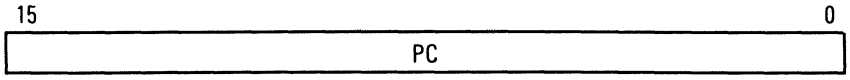
The IY register is a 16-bit register used for the indexed addressing mode, similar to the IX register. However, most instructions using the IY register

are two-byte opcodes and require an extra byte of machine code and an extra cycle of execution time. The IY register can also be used as a counter or temporary storage area.



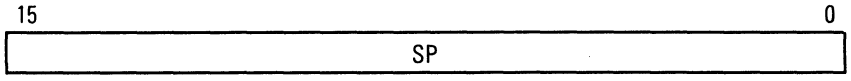
10.1.4 Program Counter (PC)

The PC is a 16-bit register that contains the address of the next instruction to be executed.



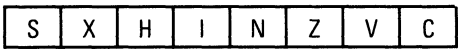
10.1.5 Stack Pointer (SP)

The SP is a 16-bit register that contains the address of the next free location on the stack. The stack is configured as a sequence of last-in first-out (LIFO) read/write registers that allow important data to be stored during interrupts and subroutine calls. Each time a new byte is added to the stack, the SP is decremented. Each time a byte is removed, the SP is incremented.



10.1.6 Condition Code Register (CCR)

The CCR is an 8-bit register in which five bits are used to indicate the results of the instruction just executed and three bits are mask bits for interrupt and stop. These bits can be individually tested by a program, and specific action can be taken as a result of their state. Each bit is explained in the following paragraphs.



10.1.6.1 CARRY/BORROW (C). When set, the C bit indicates that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during shift and rotate instructions.

10.1.6.2 OVERFLOW (V). The V bit is set if an arithmetic overflow occurred as a result of the operation. Otherwise, the V bit is cleared.

10.1.6.3 ZERO (Z). When set, the Z bit indicates the result of the last arithmetic, logical, or data manipulation operation was zero.

10.1.6.4 NEGATIVE (N). When set, the N bit indicates that the result of the last arithmetic, logical, or data manipulation operation was negative.

10.1.6.5 INTERRUPT MASK (I). The I bit is set by either hardware or program instruction to disable (mask) all maskable interrupt sources, both external or internal.

10.1.6.6 HALF-CARRY (H). The H bit is set during ADD, ABA, and ADC operations to indicate that a carry occurred between bits 3 and 4. This bit is mainly useful in binary-coded decimal (BCD) calculations.

10.1.6.7 X INTERRUPT MASK (X). The X bit is set only by hardware ($\overline{\text{RESET}}$ or $\overline{\text{XIRQ}}$) and is cleared only by the program instructions, such as transfer A to CC register (TAP), or return from interrupt (RTI).

10.1.6.8 STOP DISABLE (S). Under program control, the S bit is set to disable the STOP instruction. It is cleared to enable the STOP instruction. The STOP instruction is treated as no operation (NOP) if the S bit is set.

10.2 INSTRUCTION SET

The central processing unit (CPU) of the MC68HC11 is basically a proper extension of the MC6801 CPU. In addition to its ability to execute all M6800 and M6801 instructions, 91 new opcodes are provided by the paged opcode map. Major functional additions include a second 16-bit index register (the Y register), two types of 16×16 divide instructions, STOP and WAIT instructions, and bit manipulation instructions.

Table 10-1 shows all the MC68HC711D3 instructions in all possible addressing modes. For each instruction, the operand construction is shown as well as the number of machine code bytes and the execution time in CPU E-clock cycles. Notes provided at the end of the table explain the letters in the Operand and Execution Time columns for some instructions. Definitions of "Special Ops" found in the Boolean Expression column are found in Figure 10-1.

Table 10-1. Instructions, Addressing Modes, and Execution Times (Sheet 1 of 7)

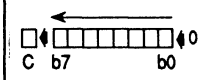
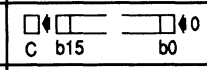
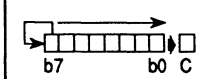
Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycles	Condition Codes
				Opcode	Operand(s)			S X H I N Z V C
ABA	Add Accumulators	A + B ⇨ A	INH	1B		1	2	— — ↕ — ↕ ↕ ↕ ↕
ABX	Add B to X	IX + 00:B ⇨ IX	INH	3A		1	3	— — — — — — — —
ABY	Add B to Y	IY + 00:B ⇨ IY	INH	18 3A		2	4	— — — — — — — —
ADCA (opr)	Add with Carry to A	A + M + C ⇨ A	A IMM	89	ii	2	2	— — ↕ — ↕ ↕ ↕ ↕
			A DIR	99	dd	2	3	
			A EXT	B9	hh ll	3	4	
			A IND,X	A9	ff	2	4	
			A IND,Y	18 A9	ff	3	5	
ADCB (opr)	Add with Carry to B	B + M + C ⇨ B	B IMM	C9	ii	2	2	— — ↕ — ↕ ↕ ↕ ↕
			B DIR	D9	dd	2	3	
			B EXT	F9	hh ll	3	4	
			B IND,X	E9	ff	2	4	
			B IND,Y	18 E9	ff	3	5	
ADDA (opr)	Add Memory to A	A + M ⇨ A	A IMM	8B	ii	2	2	— — ↕ — ↕ ↕ ↕ ↕
			A DIR	9B	dd	2	3	
			A EXT	BB	hh ll	3	4	
			A IND,X	AB	ff	2	4	
			A IND,Y	18 AB	ff	3	5	
ADDB (opr)	Add Memory to B	B + M ⇨ B	B IMM	CB	ii	2	2	— — ↕ — ↕ ↕ ↕ ↕
			B DIR	DB	dd	2	3	
			B EXT	FB	hh ll	3	4	
			B IND,X	EB	ff	2	4	
			B IND,Y	18 EB	ff	3	5	
ADDD (opr)	Add 16-Bit to D	D + M:M + 1 ⇨ D	IMM	C3	jj kk	3	4	— — — — ↕ ↕ ↕ ↕
			DIR	D3	dd	2	5	
			EXT	F3	hh ll	3	6	
			IND,X	E3	ff	2	6	
			IND,Y	18 E3	ff	3	7	
ANDA (opr)	AND A with Memory	A•M ⇨ A	A IMM	84	ii	2	2	— — — — ↕ ↕ 0 —
			A DIR	94	dd	2	3	
			A EXT	B4	hh ll	3	4	
			A IND,X	A4	ff	2	4	
			A IND,Y	18 A4	ff	3	5	
ANDB (opr)	AND B with Memory	B•M ⇨ B	B IMM	C4	ii	2	2	— — — — ↕ ↕ 0 —
			B DIR	D4	dd	2	3	
			B EXT	F4	hh ll	3	4	
			B IND,X	E4	ff	2	4	
			B IND,Y	18 E4	ff	3	5	
ASL (opr)	Arithmetic Shift Left		EXT	78	hh ll	3	6	— — — — ↕ ↕ ↕ ↕
			IND,X	68	ff	2	6	
			IND,Y	18 68	ff	3	7	
ASLA			A INH	48		1	2	
ASLB			B INH	58		1	2	
ASLD	Arithmetic Shift Left Double		INH	05		1	3	— — — — ↕ ↕ ↕ ↕
ASR (opr)	Arithmetic Shift Right		EXT	77	hh ll	3	6	— — — — ↕ ↕ ↕ ↕
			IND,X	67	ff	2	6	
			IND,Y	18 67	ff	3	7	
ASRA			A INH	47		1	2	
ASRB			B INH	57		1	2	
BCC (rel)	Branch if Carry Clear	? C = 0	REL	24	rr	2	3	— — — — — — — —
BCLR (opr) (msk)	Clear Bit(s)	M•($\overline{mm}$) ⇨ M	DIR	15	dd mm	3	6	— — — — ↕ ↕ 0 —
			IND,X	1D	ff mm	3	7	
			IND,Y	18 1D	ff mm	4	8	
BCS (rel)	Branch if Carry Clear	? C = 1	REL	25	rr	2	3	— — — — — — — —
BEQ (rel)	Branch if = Zero	? Z = 1	REL	27	rr	2	3	— — — — — — — —

Table 10-1. Instructions, Addressing Modes, and Execution Times (Sheet 2 of 7)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycles	Condition Codes
				Opcode	Operand(s)			S X H I N Z V C
BGE (rel)	Branch if ≥ Zero	? N ⊕ V = 0	REL	2C	rr	2	3	— — — — — — — —
BGT (rel)	Branch if > Zero	? Z + (N ⊕ V) = 0	REL	2E	rr	2	3	— — — — — — — —
BHI (rel)	Branch if Higher	? C + Z = 0	REL	22	rr	2	3	— — — — — — — —
BHS (rel)	Branch if Higher or Same	? C = 0	REL	24	rr	2	3	— — — — — — — —
BITA (opr)	Bit(s) Test A with Memory	A•M	A IMM	85	ii	2	2	— — — — ↕ ↕ 0 —
			A DIR	95	dd	2	3	
			A EXT	B5	hh ll	3	4	
			A IND,X	A5	ff	2	4	
			A IND,Y	18 A5	ff	3	5	
BITB (opr)	Bit(s) Test B with Memory	B•M	B IMM	C5	ii	2	2	— — — — ↕ ↕ 0 —
			B DIR	D5	dd	2	3	
			B EXT	F5	hh ll	3	4	
			B IND,X	E5	ff	2	4	
			B IND,Y	18 E5	ff	3	5	
BLE (rel)	Branch if ≤ Zero	? Z + (N ⊕ V) = 1	REL	2F	rr	2	3	— — — — — — — —
BLO (rel)	Branch if Lower	? C = 1	REL	25	rr	2	3	— — — — — — — —
BLS (rel)	Branch if Lower or Same	? C + Z = 1	REL	23	rr	2	3	— — — — — — — —
BLT (rel)	Branch If < Zero	? N ⊕ V = 1	REL	2D	rr	2	3	— — — — — — — —
BMI (rel)	Branch if Minus	? N = 1	REL	2B	rr	2	3	— — — — — — — —
BNE (rel)	Branch if Not = Zero	? Z = 0	REL	26	rr	2	3	— — — — — — — —
BPL (rel)	Branch if Plus	? N = 0	REL	2A	rr	2	3	— — — — — — — —
BRA (rel)	Branch Always	? 1 = 1	REL	20	rr	2	3	— — — — — — — —
BRCLR(opr) (msk) (rel)	Branch if Bit(s) Clear	? M•mm = 0	DIR	13	dd mm rr	4	6	— — — — — — — —
			IND,X	1F	ff mm rr	4	7	
			IND,Y	18 1F	ff mm rr	5	8	
BRN (rel)	Branch Never	? 1 = 0	REL	21	rr	2	3	— — — — — — — —
BRSET(opr) (msk) (rel)	Branch if Bit(s) Set	? ($\overline{M}$)•mm = 0	DIR	12	dd mm rr	4	6	— — — — — — — —
			IND,X	1E	ff mm rr	4	7	
			IND,Y	18 1E	ff mm rr	5	8	
BSET (opr) (msk)	Set Bit(s)	M + mm ⇩ M	DIR	14	dd mm	3	6	— — — — ↕ ↕ 0 —
			IND,X	1C	ff mm	3	7	
			IND,Y	18 1C	ff mm	4	8	
BSR (rel)	Branch to Subroutine	See Special Ops	REL	8D	rr	2	6	— — — — — — — —
BVC (rel)	Branch if Overflow Clear	? V = 0	REL	28	rr	2	3	— — — — — — — —
BVS (rel)	Branch if Overflow Set	? V = 1	REL	29	rr	2	3	— — — — — — — —
CBA	Compare A to B	A − B	INH	11		1	2	— — — — ↕ ↕ ↕ ↕
CLC	Clear Carry Bit	0 ⇩ C	INH	0C		1	2	— — — — — — — 0
CLI	Clear Interrupt Mask	0 ⇩ I	INH	0E		1	2	— — — 0 — — — —
CLR (opr)	Clear Memory Byte	0 ⇩ M	EXT	7F	hh ll	3	6	— — — — 0 1 0 0
			IND,X	6F	ff	2	6	
			IND,Y	18 6F	ff	3	7	
CLRA	Clear Accumulator A	0 ⇩ A	A INH	4F		1	2	— — — — 0 1 0 0
CLRB	Clear Accumulator B	0 ⇩ B	B INH	5F		1	2	— — — — 0 1 0 0
CLV	Clear Overflow Flag	0 ⇩ V	INH	0A		1	2	— — — — — — 0 —
CMPA (opr)	Compare A to Memory	A − M	A IMM	81	ii	2	2	— — — — ↕ ↕ ↕ ↕
			A DIR	91	dd	2	3	
			A EXT	B1	hh ll	3	4	
			A IND,X	A1	ff	2	4	
			A IND,Y	18 A1	ff	3	5	

Table 10-1. Instructions, Addressing Modes, and Execution Times (Sheet 3 of 7)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)				Condition Codes
				Opcode	Operand(s)			S X H I N Z V C
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 2 3 2 3	2 3 4 4 5	— — — — ↕ ↕ ↕ ↕
COM (opr)	Ones Complement Memory Byte	$FF–M ⇩ M	EXT IND,X IND,Y	73 63 18 63	hh ll ff ff	3 2 3	6 6 7	— — — — ↕ ↕ 0 1
COMA	Ones Complement A	$FF–A ⇩ A	A INH	43		1	2	— — — — ↕ ↕ 0 1
COMB	Ones Complement B	$FF–B ⇩ B	B INH	53		1	2	— — — — ↕ ↕ 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	4 3 4 3 3	5 6 7 7 7	— — — — ↕ ↕ ↕ ↕
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	3 2 3 2 3	4 5 6 6 7	— — — — ↕ ↕ ↕ ↕
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	4 3 4 3 3	5 6 7 7 7	— — — — ↕ ↕ ↕ ↕
DAA	Decimal Adjust A	Adjust Sum to BCD	INH	19		1	2	— — — — ↕ ↕ ↕ ↕
DEC (opr)	Decrement Memory Byte	M – 1 ⇩ M	EXT IND,X IND,Y	7A 6A 18 6A	hh ll ff ff	3 2 3	6 6 7	— — — — ↕ ↕ ↕ —
DECA	Decrement Accumulator A	A – 1 ⇩ A	A INH	4A		1	2	— — — — ↕ ↕ ↕ —
DECB	Decrement Accumulator B	B – 1 ⇩ B	B INH	5A		1	2	— — — — ↕ ↕ ↕ —
DES	Decrement Stack Pointer	SP – 1 ⇩ SP	INH	34		1	3	— — — — — — — —
DEX	Decrement Index Register X	IX – 1 ⇩ IX	INH	09		1	3	— — — — — ↕ — —
DEY	Decrement Index Register Y	IY – 1 ⇩ IY	INH	18 09		2	4	— — — — — ↕ — —
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 2 3 2 3	2 3 4 4 5	— — — — ↕ ↕ 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 2 3 2 3	2 3 4 4 5	— — — — ↕ ↕ 0 —
FDIV	Fractional Divide 16 by 16	D/IX ⇩ IX; r ⇩ D	INH	03		1	41	— — — — — ↕ ↕ ↕
IDIV	Integer Divide 16 by 16	D/IX ⇩ IX; r ⇩ D	INH	02		1	41	— — — — — ↕ 0 ↕
INC (opr)	Increment Memory Byte	M + 1 ⇩ M	EXT IND,X IND,Y	7C 6C 18 6C	hh ll ff ff	3 2 3	6 6 7	— — — — ↕ ↕ ↕ —
INCA	Increment Accumulator A	A + 1 ⇩ A	A INH	4C		1	2	— — — — ↕ ↕ ↕ —
INCB	Increment Accumulator B	B + 1 ⇩ B	B INH	5C		1	2	— — — — ↕ ↕ ↕ —
INS	Increment Stack Pointer	SP + 1 ⇩ SP	INH	31		1	3	— — — — — — — —

Table 10-1. Instructions, Addressing Modes, and Execution Times (Sheet 4 of 7)

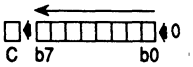
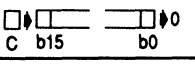
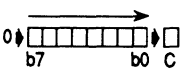
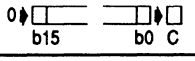
Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycles	Condition Codes
				Opcode	Operand(s)			S X H I N Z V C
INX	Increment Index Register X	IX + 1 ⇾ IX	INH	08		1	3	— — — — — ↕ — —
INY	Increment Index Register Y	IY + 1 ⇾ IY	INH	18 08		2	4	— — — — — ↕ — —
JMP (opr)	Jump	See Special Ops	EXT	7E	hh ll	3	3	— — — — — — — —
			IND,X	6E	ff	2	3	
			IND,Y	18 6E	ff	3	4	
JSR (opr)	Jump to Subroutine	See Special Ops	DIR	9D	dd	2	5	— — — — — — — —
			EXT	BD	hh ll	3	6	
			IND,X	AD	ff	2	6	
			IND,Y	18 AD	ff	3	7	
LDAA (opr)	Load Accumulator A	M ⇾ A	A IMM	86	ii	2	2	— — — — ↕ ↕ 0 —
			A DIR	96	dd	2	3	
			A EXT	B6	hh ll	3	4	
			A IND,X	A6	ff	2	4	
			A IND,Y	18 A6	ff	3	5	
LDAB (opr)	Load Accumulator B	M ⇾ B	B IMM	C6	ii	2	2	— — — — ↕ ↕ 0 —
			B DIR	D6	dd	2	3	
			B EXT	F6	hh ll	3	4	
			B IND,X	E6	ff	2	4	
			B IND,Y	18 E6	ff	3	5	
LDD (opr)	Load Double Accumulator D	M ⇾ A,M + 1 ⇾ B	IMM	CC	jj kk	3	3	— — — — ↕ ↕ 0 —
			DIR	DC	dd	2	4	
			EXT	FC	hh ll	3	5	
			IND,X	EC	ff	2	5	
			IND,Y	18 EC	ff	3	6	
LDS (opr)	Load Stack Pointer	M:M + 1 ⇾ SP	IMM	8E	jj kk	3	3	— — — — ↕ ↕ 0 —
			DIR	9E	dd	2	4	
			EXT	BE	hh ll	3	5	
			IND,X	AE	ff	2	5	
			IND,Y	18 AE	ff	3	6	
LDX (opr)	Load Index Register X	M:M + 1 ⇾ IX	IMM	CE	jj kk	3	3	— — — — ↕ ↕ 0 —
			DIR	DE	dd	2	4	
			EXT	FE	hh ll	3	5	
			IND,X	EE	ff	2	5	
			IND,Y	CD EE	ff	3	6	
LDY (opr)	Load Index Register Y	M:M + 1 ⇾ IY	IMM	18 CE	jj kk	4	4	— — — — ↕ ↕ 0 —
			DIR	18 DE	dd	3	5	
			EXT	18 FE	hh ll	4	6	
			IND,X	1A EE	ff	3	6	
			IND,Y	18 EE	ff	3	6	
LSL (opr)	Logical Shift Left		EXT	78	hh ll	3	6	— — — — ↕ ↕ ↕ ↕
			IND,X	68	ff	2	6	
			IND,Y	18 68	ff	3	7	
LSLA			A INH	48		1	2	
LSLB			B INH	58		1	2	
LSLD	Logical Shift Left Double		INH	05		1	3	— — — — ↕ ↕ ↕ ↕
LSR (opr)	Logical Shift Right		EXT	74	hh ll	3	6	— — — — 0 ↕ ↕ ↕
			IND,X	64	ff	2	6	
			IND,Y	18 64	ff	3	7	
LSRA			A INH	44		1	2	
LSRB			B INH	54		1	2	
LSRD	Logical Shift Right Double		INH	04		1	3	— — — — 0 ↕ ↕ ↕
MUL	Multiply 8 by 8	A × B ⇾ D	INH	3D		1	10	— — — — — — — ↕

Table 10-1. Instructions, Addressing Modes, and Execution Times (Sheet 5 of 7)

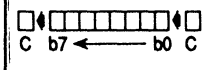
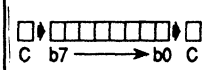
Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycles	Condition Codes
				Opcode	Operand(s)			S X H I N Z V C
NEG (opr)	Twos Complement Memory Byte	0 – M ⇩ M	EXT IND,X IND,Y	70 60 18 60	hh ll ff ff	3 2 3	6 6 7	— — — — ↕ ↕ ↕ ↕
NEGA	Twos Complement A	0 – A ⇩ A	A INH	40		1	2	— — — — ↕ ↕ ↕ ↕
NEGB	Twos Complement B	0 – B ⇩ B	B INH	50		1	2	— — — — ↕ ↕ ↕ ↕
NOP	No Operation	No Operation	INH	01		1	2	— — — — — — — —
ORAA (opr)	OR Accumulator A (Inclusive)	A + M ⇩ A	A IMM A DIR A EXT A IND,X A IND,Y	8A 9A BA AA 18 AA	ii dd hh ll ff ff	2 2 3 2 3	2 3 4 4 5	— — — — ↕ ↕ 0 —
ORAB (opr)	OR Accumulator B (Inclusive)	B + M ⇩ B	B IMM B DIR B EXT B IND,X B IND,Y	CA DA FA EA 18 EA	ii dd hh ll ff ff	2 2 3 2 3	2 3 4 4 5	— — — — ↕ ↕ 0 —
PSHA	Push A onto Stack	A ⇩ Stk,SP = SP – 1	A INH	36		1	3	— — — — — — — —
PSHB	Push B onto Stack	B ⇩ Stk,SP = SP – 1	B INH	37		1	3	— — — — — — — —
PSHX	Push X onto Stack (Lo First)	IX ⇩ Stk,SP = SP – 2	INH	3C		1	4	— — — — — — — —
PSHY	Push Y onto Stack (Lo First)	IY ⇩ Stk,SP = SP – 2	INH	18 3C		2	5	— — — — — — — —
PULA	Pull A from Stack	SP = SP + 1,A ⇧ Stk	A INH	32		1	4	— — — — — — — —
PULB	Pull B from Stack	SP = SP + 1,B ⇧ Stk	B INH	33		1	4	— — — — — — — —
PULX	Pull X from Stack (Hi First)	SP = SP + 2,IX ⇧ Stk	INH	38		1	5	— — — — — — — —
PULY	Pull Y from Stack (Hi First)	SP = SP + 2,IY ⇧ Stk	INH	18 38		2	6	— — — — — — — —
ROL (opr) ROLA ROLB	Rotate Left		EXT IND,X IND,Y A INH B INH	79 69 18 69 49 59	hh ll ff ff	3 2 3 1 1	6 6 7 2 2	— — — — ↕ ↕ ↕ ↕
ROR (opr) RORA RORB	Rotate Right		EXT IND,X IND,Y A INH B INH	76 66 18 66 46 56	hh ll ff ff	3 2 3 1 1	6 6 7 2 2	— — — — ↕ ↕ ↕ ↕
RTI	Return from Interrupt	See Special Ops	INH	3B		1	12	↕ ↕ ↕ ↕ ↕ ↕ ↕ ↕
RTS	Return from Subroutine	See Special Ops	INH	39		1	5	— — — — — — — —
SBA	Subtract B from A	A – B ⇩ A	INH	10		1	2	— — — — ↕ ↕ ↕ ↕
SBCA (opr)	Subtract with Carry from A	A – M – C ⇩ A	A IMM A DIR A EXT A IND,X A IND,Y	82 92 B2 A2 18 A2	ii dd hh ll ff ff	2 2 3 2 3	2 3 4 4 5	— — — — ↕ ↕ ↕ ↕
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 2 3 2 3	2 3 4 4 5	— — — — ↕ ↕ ↕ ↕
SEC	Set Carry	1 ⇩ C	INH	OD		1	2	— — — — — — — 1
SEI	Set Interrupt Mask	1 ⇩ I	INH	OF		1	2	— — — 1 — — — —
SEV	Set Overflow Flag	1 ⇩ V	INH	OB		1	2	— — — — — — 1 —

Table 10-1. Instructions, Addressing Modes, and Execution Times (Sheet 6 of 7)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycles	Condition Codes
				Opcode	Operand(s)			S X H I N Z V C
STAA (opr)	Store Accumulator A	A ⇨ M	A DIR	97	dd	2	3	— — — — ↕ ↕ 0 —
			A EXT	B7	hh ll	3	4	
			A IND,X	A7	ff	2	4	
			A IND,Y	18 A7	ff	3	5	
STAB (opr)	Store Accumulator B	B ⇨ M	B DIR	D7	dd	2	3	— — — — ↕ ↕ 0 —
			B EXT	F7	hh ll	3	4	
			B IND,X	E7	ff	2	4	
			B IND,Y	18 E7	ff	3	5	
STD (opr)	Store Accumulator D	A ⇨ M,B ⇨ M + 1	DIR	DD	dd	2	4	— — — — ↕ ↕ 0 —
			EXT	FD	hh ll	3	5	
			IND,X	ED	ff	2	5	
			IND,Y	18 ED	ff	3	6	
STOP	Stop Internal Clocks		INH	CF		1	2	— — — — — — — —
STS (opr)	Store Stack Pointer	SP ⇨ M:M + 1	DIR	9F	dd	2	4	— — — — ↕ ↕ 0 —
			EXT	BF	hh ll	3	5	
			IND,X	AF	ff	2	5	
			IND,Y	18 AF	ff	3	6	
STX (opr)	Store Index Register X	IX ⇨ M:M + 1	DIR	DF	dd	2	4	— — — — ↕ ↕ 0 —
			EXT	FF	hh ll	3	5	
			IND,X	EF	ff	2	5	
			IND,Y	CD EF	ff	3	6	
STY (opr)	Store Index Register Y	IY ⇨ M:M + 1	DIR	18 DF	dd	3	5	— — — — ↕ ↕ 0 —
			EXT	18 FF	hh ll	4	6	
			IND,X	1A EF	ff	3	6	
			IND,Y	18 EF	ff	3	6	
SUBA (opr)	Subtract Memory from A	A − M ⇨ A	A IMM	80	ii	2	2	— — — — ↕ ↕ ↕ ↕
			A DIR	90	dd	2	3	
			A EXT	B0	hh ll	3	4	
			A IND,X	A0	ff	2	4	
			A IND,Y	18 A0	ff	3	5	
SUBB (opr)	Subtract Memory from B	B − M ⇨ B	B IMM	C0	ii	2	2	— — — — ↕ ↕ ↕ ↕
			B DIR	D0	dd	2	3	
			B EXT	F0	hh ll	3	4	
			B IND,X	E0	ff	2	4	
			B IND,Y	18 E0	ff	3	5	
SUBD (opr)	Subtract Memory from D	D − M:M + 1 ⇨ D	IMM	83	jj kk	3	4	— — — — ↕ ↕ ↕ ↕
			DIR	93	dd	2	5	
			EXT	B3	hh ll	3	6	
			IND,X	A3	ff	2	6	
			IND,Y	18 A3	ff	3	7	
SWI	Software Interrupt	See Special Ops	INH	3F		1	14	— — — 1 — — — —
TAB	Transfer A to B	A ⇨ B	INH	16		1	2	— — — — ↕ ↕ 0 —
TAP	Transfer A to CC Register	A ⇨ CCR	INH	06		1	2	↕ ↕ ↕ ↕ ↕ ↕ ↕ ↕
TBA	Transfer B to A	B ⇨ A	INH	17		1	2	— — — — ↕ ↕ 0 —
TEST	TEST (Only in Test Modes)	Address Bus Counts	INH	00		I	*	— — — — — — — —
TPA	Transfer CC Register to A	CCR ⇨ A	INH	07		1	2	— — — — — — — —
TST (opr)	Test for Zero or Minus	M − 0	EXT	7D	hh ll	3	6	— — — — ↕ ↕ 0 0
			IND,X	6D	ff	2	6	
			IND,Y	18 6D	ff	3	7	
TSTA		A − 0	A INH	4D		1	2	— — — — ↕ ↕ 0 0
TSTB		B − 0	B INH	5D		1	2	— — — — ↕ ↕ 0 0
TSX	Transfer Stack Pointer to X	SP + 1 ⇨ IX	INH	30		1	3	— — — — — — — —
TSY	Transfer Stack Pointer to Y	SP + 1 ⇨ IY	INH	18 30		2	4	— — — — — — — —

Table 10-1. Instructions, Addressing Modes, and Execution Times (Sheet 7 of 7)

Source Form(s)	Operation	Boolean Expression	Addressing Mode for Operand	Machine Coding (Hexadecimal)		Bytes	Cycles	Condition Codes
				Opcode	Operand(s)			S X H I N Z V C
TXS	Transfer X to Stack Pointer	IX − 1 ⇨ SP	INH	35		1	3	— — — — — — — —
TYS	Transfer Y to Stack Pointer	IY − 1 ⇨ SP	INH	18 35		2	4	— — — — — — — —
WAI	Wait for Interrupt	Stack Regs & WAIT	INH	3E		2	**	— — — — — — — —
XGDX	Exchange D with X	IX ⇨ D, D ⇨ IX	INH	8F		1	3	— — — — — — — —
XGDY	Exchange D with Y	IY ⇨ D, D ⇨ IY	INH	18 8F		2	4	— — — — — — — —

Cycle

*Infinity or Until Reset Occurs

**Twelve 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 the Address Following the Machine Code Offset Byte)

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

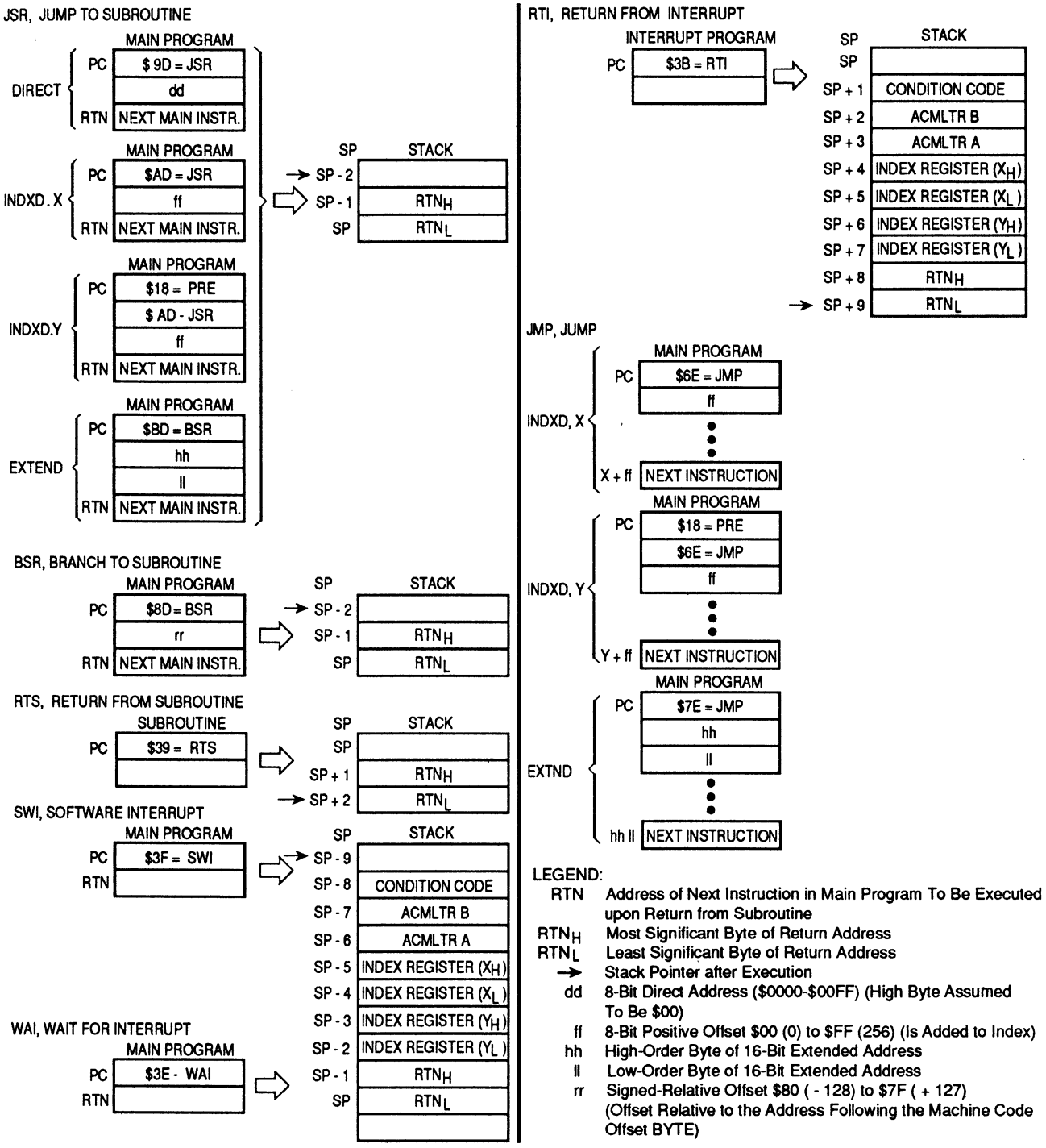


Figure 10-1. Special Operations

10.3 ADDRESSING MODES

The MC68HC11 MCU uses six different addressing modes that provide the programmer with an opportunity to optimize code for all situations. Some instructions require an additional byte before the opcode to accommodate a multipage opcode map. This byte is called the prebyte.

The term effective address (EA) is used in describing the various addressing modes. The effective address is the address from which the argument for an instruction is fetched or stored.

10.3.1 Immediate

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. These are two- or three-byte instructions, (four-byte if a prebyte is required).

10.3.2 Direct

In the direct addressing mode, the low-order address of the operand address is contained in a single byte following the opcode, and the high-order byte of an address is assumed to be $00. Direct addressing allows the user to directly address $0000–$00FF using two-byte instructions. Execution time is reduced by eliminating the additional memory access. In most applications, this 256-byte area is reserved for frequently referenced data. In this MCU, software can configure the memory map so that internal registers and RAM or external memory space can occupy these addresses.

10.3.3 Extended

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

10.3.4 Indexed

In the indexed addressing mode, one of the index registers, either X or Y, is used in calculating the effective address. In this case, the effective address is variable and depends on two factors, the current contents of the index

register (X or Y) being used, and the 8-bit unsigned offset contained in the instruction. This addressing mode allows the programmer to reference any memory location in the 64K-byte address space. These are usually two- or three-byte instructions, depending on whether a prebyte is required. They consist of the opcode plus the 8-bit offset and perhaps a prebyte.

10.3.5 Relative

The relative addressing mode is used only in branch instructions. In relative addressing, the contents of the 8-bit signed byte following the opcode, the offset, is added to the PC. This addition is done if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. Relative addressing instructions are usually two-byte instructions.

10.3.6 Inherent

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator, as well as the control instruction with no other arguments, are included in this addressing mode. These are one- or two-byte instructions.

10.3.7 Prebyte

To expand the number of instructions used in this MCU, a prebyte has been added to certain instructions. The instructions affected are usually associated with index register Y. Accessing opcodes from page 2, 3, or 4 would require a prebyte.

10.4 OPCODE MAP SUMMARY

Table 10-2 is an opcode map for the instructions and addressing modes used by the MC68HC711D3.

Table 10-2. Opcode Map

OPCODE MAP PAGE 1

		INH	INH	REL	INH	ACCA	ACCB	IND,X	EXT	IMM	DIR	IND,X	EXT	IMM	DIR	IND,X	EXT	
MSB		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
LSB		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0	TEST*	SBA	BRA	TSX	NEG				SUB								0
0001	1	NOP	CBA	BRN	INS					CMP								1
0010	2	IDIV	BRSET	BHI	PULA					SBC								2
0011	3	FDIV	BRCLR	BLS	PULB	COM				SUBD				ADDD				3
0100	4	LSRD	BSET	BCC	DES	LSR				AND								4
0101	5	ASLD	BCLR	BCS	TXS					BIT								5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA				STA			7
1000	8	INX	PG 2	BVC	PULX	ASL				EOR								8
1001	9	DEX	DAA	BVS	RTS	ROL				ADC								9
1010	A	CLV	PG 3	BPL	ABX	DEC				ORA								A
1011	B	SEV	ABA	BMI	RTI					ADD								B
1100	C	CLC	BSET	BGE	PSHX	INC				CPX				LDD				C
1101	D	SEC	BCLR	BLT	MUL	TST				BSR	JSR			PG 4	STD			D
1110	E	CLI	BRSET	BGT	WAI			JMP		LDS				LDX				E
1111	F	SEI	BRCLR	BLE	SWI	CLR				XGDX	STS			STOP	STX			F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

DIR ↑ IND,X ↓

OPCODE MAP PAGE 2 (18xx)

		INH			INH			IND,Y		IMM	DIR	IND,Y	EXT	IMM	DIR	IND,Y	EXT	
MSB		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
LSB		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0				TSY			NEG				SUB				SUB		0
0001	1											CMP				CMP		1
0010	2											SBC				SBC		2
0011	3							COM				SUBD				ADDD		3
0100	4							LSR				AND				AND		4
0101	5				TYS							BIT				BIT		5
0110	6							ROR				LDA				LDA		6
0111	7							ASR				STA				STA		7
1000	8	INY			PULY			ASL				EOR				EOR		8
1001	9	DEY						ROL				ADC				ADC		9
1010	A				ABY			DEC				ORA				ORA		A
1011	B											ADD				ADD		B
1100	C		BSET		PSHY			INC		CPY						LDD		C
1101	D		BCLR					TST				JSR				STD		D
1110	E		BRSET					JMP				LDS			LDY			E
1111	F		BRCLR					CLR		XGDY		STS			STY			F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

IND,Y ↓

Table 10-2. Opcode Map (Continued)

OPCODE MAP PAGE 3 (1Axx)									ACCA				ACCB			
									IMM	DIR	IND,X	EXT			IND,X	
MSB		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
LSB		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0																0
0001	1																1
0010	2																2
0011	3									CPD							3
0100	4																4
0101	5																5
0110	6																6
0111	7																7
1000	8																8
1001	9																9
1010	A																A
1011	B																B
1100	C											CPY					C
1101	D																D
1110	E															LDY	E
1111	F															STY	F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

OPCODE MAP PAGE 4 (CDxx)									ACCA				ACCB			
											IND,Y				IND,Y	
MSB		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
LSB		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0																0
0001	1																1
0010	2																2
0011	3											CPD					3
0100	4																4
0101	5																5
0110	6																6
0111	7																7
1000	8																8
1001	9																9
1010	A																A
1011	B																B
1100	C											CPX					C
1101	D																D
1110	E															LDX	E
1111	F															STX	F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

SECTION 11
ELECTRICAL SPECIFICATIONS

This section contains the electrical specifications and associated timing information for the MC68HC711D3 MCU.

11.1 MAXIMUM RATINGS

The following ratings define a range of operating conditions which will not permanently damage the device. However, sections of the device may not operate normally while being exposed to the extremes of this range.

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC68HC711D3	T_A	T_L to T_H −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C
Current Drain per Pin* Excluding V_{DD}, V_{SS}, V_{RH}, and V_{RL}	I_D	25	mA

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

This device contains protective circuitry against damage due to high static voltages or electrical fields; however, it is advised that normal precautions be taken to avoid application of any voltages higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or V_{DD}).

11.2 THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance	θ_{JA}		°C/W
Plastic 44-Pin Quad Pack (PLCC)		50	
40-Pin Dual-in-line Pkg (DIP)		50	
Ceramic 40-Pin DIP (Cerdip)		60	

11.3 POWER CONSIDERATIONS

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

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{I/O}$
P_{INT} = $I_{DD} \times V_{DD}$, Watts — Chip Internal Power
$P_{I/O}$ = Power Dissipation on Input and Output Pins — User Determined

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

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

$$P_D = K \div (T_J + 273°C) \quad (2)$$

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

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

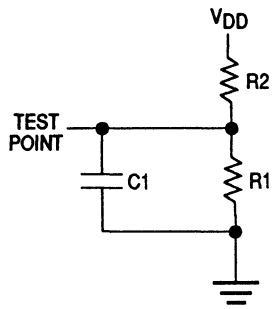
11.4 DC ELECTRICAL CHARACTERISTICS

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

Characteristics		Symbol	Min	Max	Unit
Output Voltage I_{Load} = ±10.0 μA (see Note 1)	All Outputs All Outputs Except RESET and MODA	V_{OL} V_{OH}	— V_{DD} − 0.1	0.1 —	V
Output High Voltage I_{Load} = − 0.8 mA, V_{DD} = 4.5 V (see Note 1)	All Outputs Except RESET, XTAL, and MODA	V_{OH}	V_{DD} − 0.8	—	V
Output Low Voltage I_{Load} = 1.6 mA	All Outputs Except XTAL	V_{OL}	—	0.4	V
Input High Voltage	All Inputs Except RESET RESET	V_{IH}	0.7 × V_{DD} 0.8 × V_{DD}	V_{DD} + 0.3 V_{DD} + 0.3	V
Input Low Voltage	All Inputs	V_{IL}	V_{SS} − 0.3	0.2 × V_{DD}	V
I/O Ports, Three-State Leakage $V_{in} = V_{IH}$ or V_{IL}	PA7, PA3, PC7–PC0, PD7–PD0 MODA/LIR, RESET	I_{OZ}	—	±10	μA
Input Current (see Note 2) $V_{in} = V_{DD}$ or V_{SS} $V_{in} = V_{DD}$ or V_{SS}	 IRQ, XIRQ MODB/V_{stby}	I_{in}	 — —	 ±1 ±10	μA
RAM Standby Voltage	Powerdown	V_{SB}	4.0	V_{DD}	V
RAM Standby Current	Powerdown	I_{SB}	—	20	μA
Total Supply Current (see Note 2) RUN: Single-Chip Mode Expanded-Nonmultiplexed Mode WAIT: (All Peripheral Functions Shut Down) Single-Chip Mode Expanded-Nonmultiplexed Mode STOP: (No Clocks) Single-Chip Mode		 I_{DD} W_{IDD} S_{IDD} 	 — — — — —	 15 27 6 10 100	 mA mA mA mA μA
Input Capacitance	PA3–PA0, IRQ, XIRQ, EXTAL PA7, PC7–PC0, PD7–PD0, MODA/LIR, RESET	C_{in}	— —	8 12	pF
Power Dissipation	Single-Chip Mode Expanded-Nonmultiplexed Mode	P_D	— —	85 150	mW
EPROM Programming Voltage		V_{PP}	11.75	12.75	V
EPROM Programming Time		t_{PP}	2	4	ms

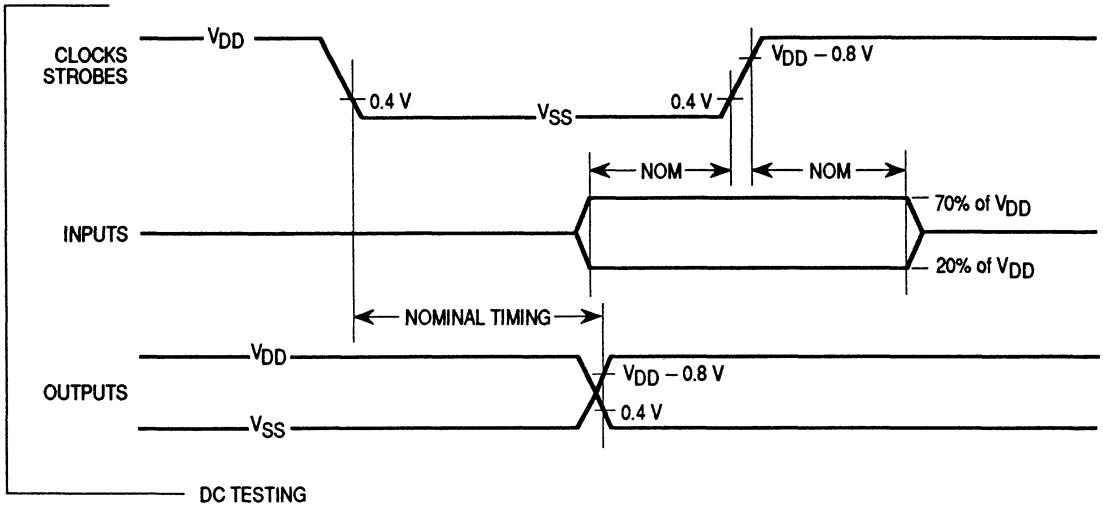
NOTES:

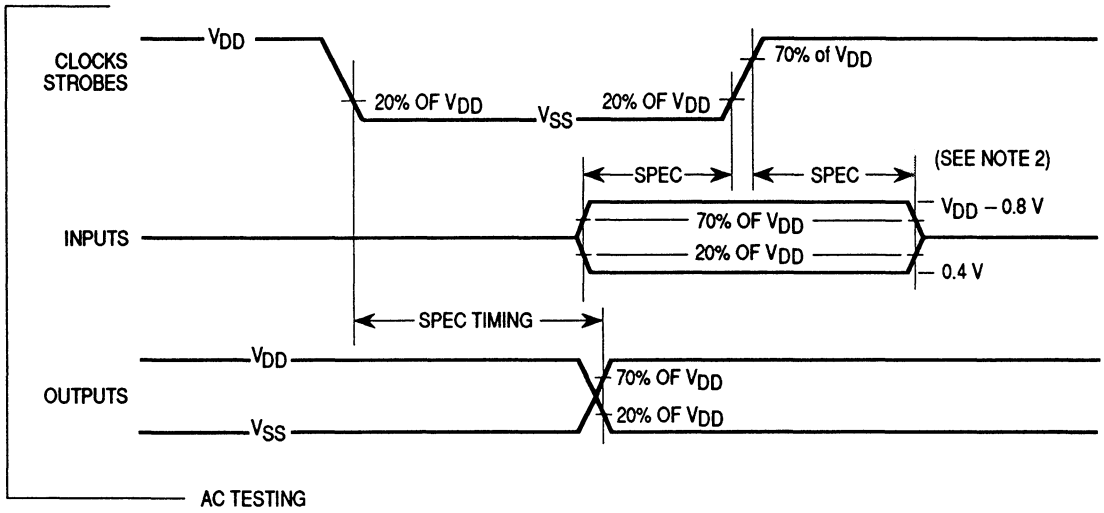
1. V_{OH} specification for RESET and MODA is not applicable because they are open-drain pins. V_{OH} specification is not applicable to ports C and D in wired-OR mode.
2. All ports configured as inputs: $V_{IL} \leq 0.2$ V, $V_{IH} \geq V_{DD} - 0.2$ V; no dc loads; EXTAL is driven with a square wave; t_{cyc} = 476.5 ns.



Equivalent test load [1]

Pins	R1	R2	C1
PA7-PA3 PB7-PB0 PC7-PC0 PD7,6,5,0 E	3.26 kΩ	2.38 kΩ	90 pF
PD4-PD1	3.26 kΩ	2.38 kΩ	200 pF





NOTES:

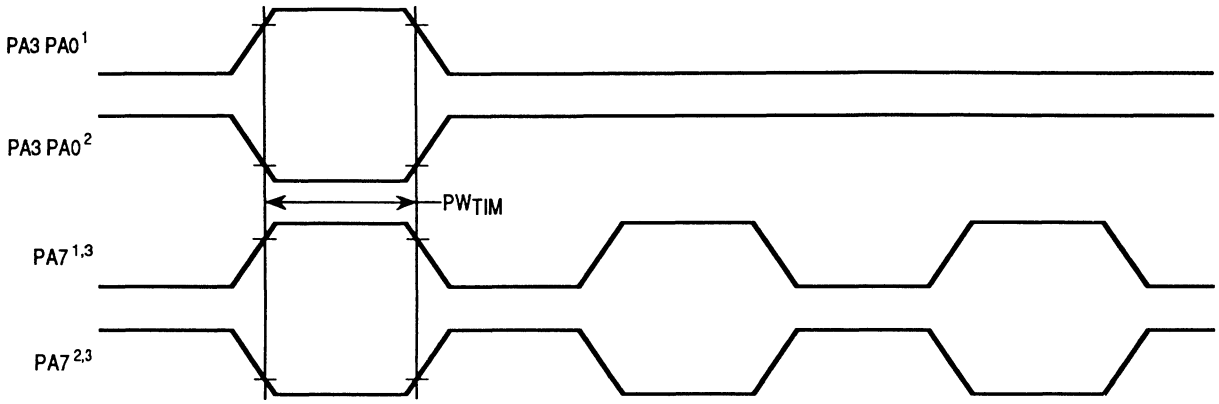
1. Full test loads are applied during all AC electrical timing measurements.
2. During AC timing measurements, inputs are driven to 0.4 V and V_{DD} – 0.8 V while timing measurements are taken at the 20% and 70% of V_{DD} points.

Figure 11-1. Test Methods

11.5 CONTROL TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_o	dc	1.0	dc	2.0	dc	2.1	MHz
E-Clock Period	t_{cyc}	1000	—	500	—	476	—	ns
Crystal Frequency	f_{XTAL}	—	4.0	—	8.0	—	8.4	MHz
External Oscillator Frequency	4 f_o	dc	4.0	dc	8.0	dc	8.4	MHz
Processor Control Setup Time ($t_{PCS} = 1/4\ t_{cyc} - 50$ ns) (See Figures 11-3, 11-5, and 11-7)	t_{PCS}	—	200	—	75	—	69	ns
Reset Input Pulse Width (To Guarantee External Reset Vector) (see Note 1 and Figure 11-3)	PW_{RSTL}	8	—	8	—	8	—	t_{cyc}
(Minimum Input Time May Be Pre-empted by Internal Reset)		1	—	1	—	1	—	
Mode Programming Setup Time (See Figure 11-3)	t_{MPS}	2	—	2	—	2	—	t_{cyc}
Mode Programming Hold Time (See Figure 11-3)	t_{MPH}	0	—	0	—	0	—	ns
Interrupt Pulse Width, $\overline{IRQ}$ Edge Sensitive Mode ($PW_{IRQ} = t_{cyc} + 20$ ns) (See Figure 11-4 and 11-6)	PW_{IRQ}	1020	—	520	—	496	—	ns
Wait Recovery Startup Time (See Figure 11-5)	t_{WRS}	—	4	—	4	—	4	t_{cyc}
Timer Pulse Width Input Capture Pulse Accumulator Input ($PW_{TIM} = t_{cyc} + 20$ ns) (See Figure 11-2)	PW_{TIM}	1020	—	520	—	496	—	ns

NOTES:
1. $\overline{RESET}$ will be recognized during the first clock cycle in which 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.
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.



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

Figure 11-2. Timer Inputs Timing Diagram

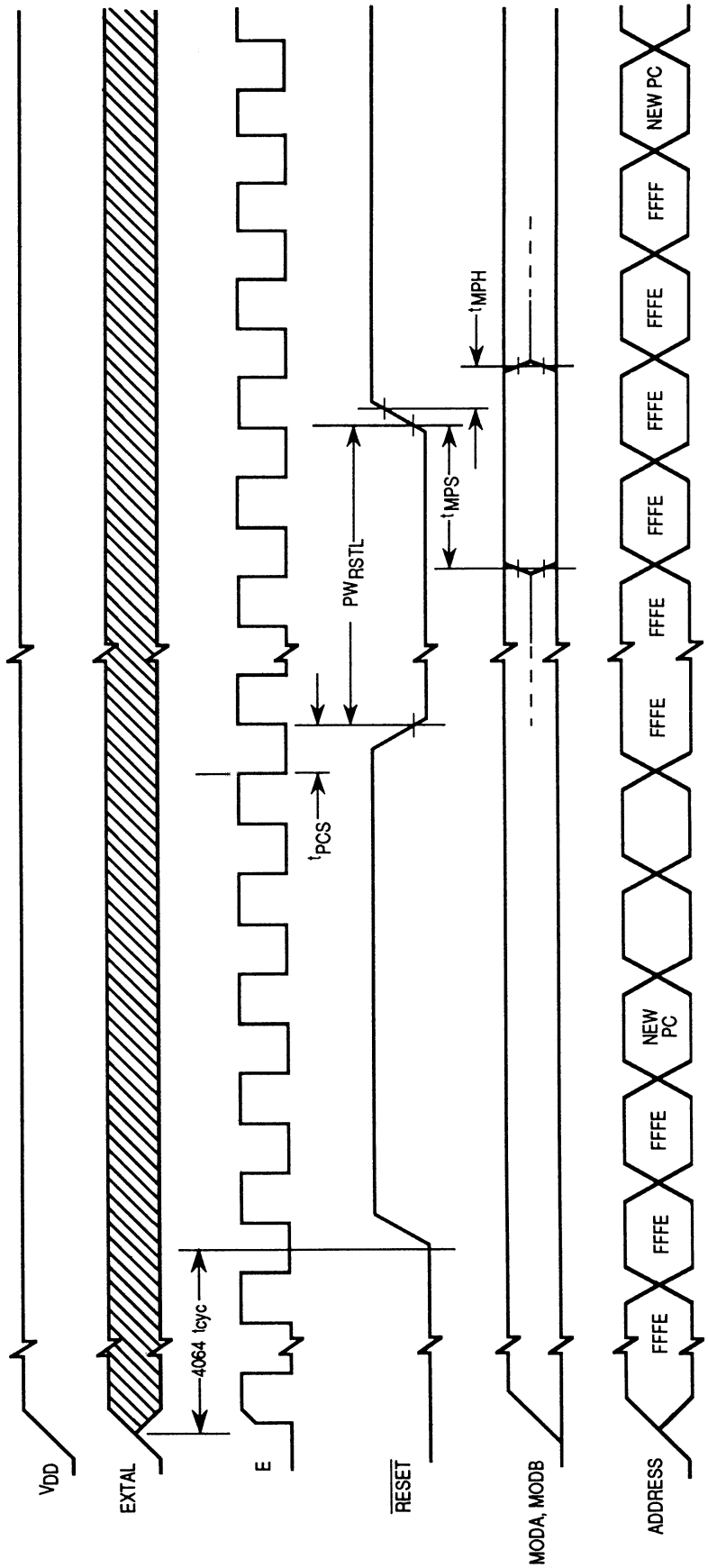
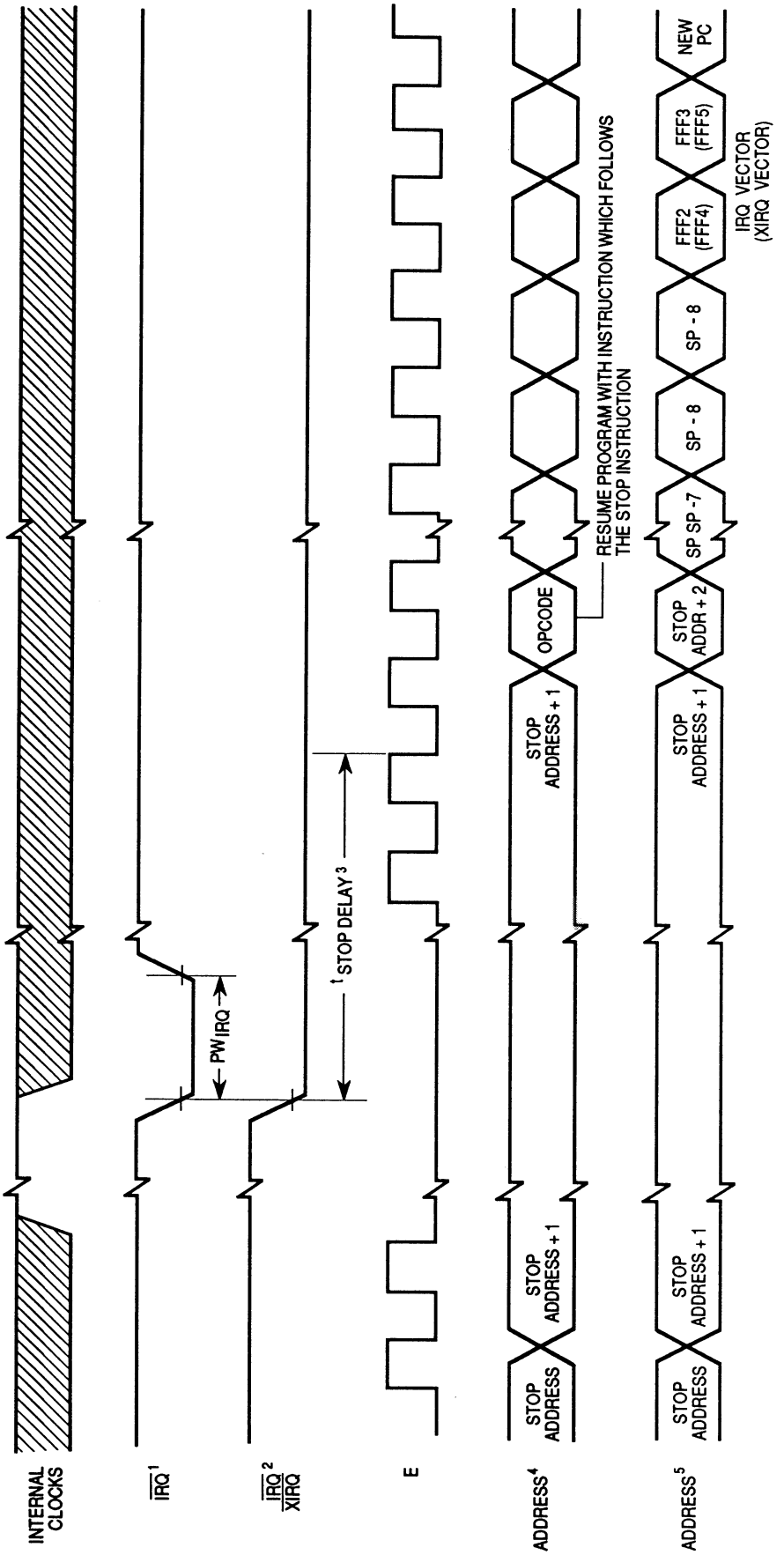
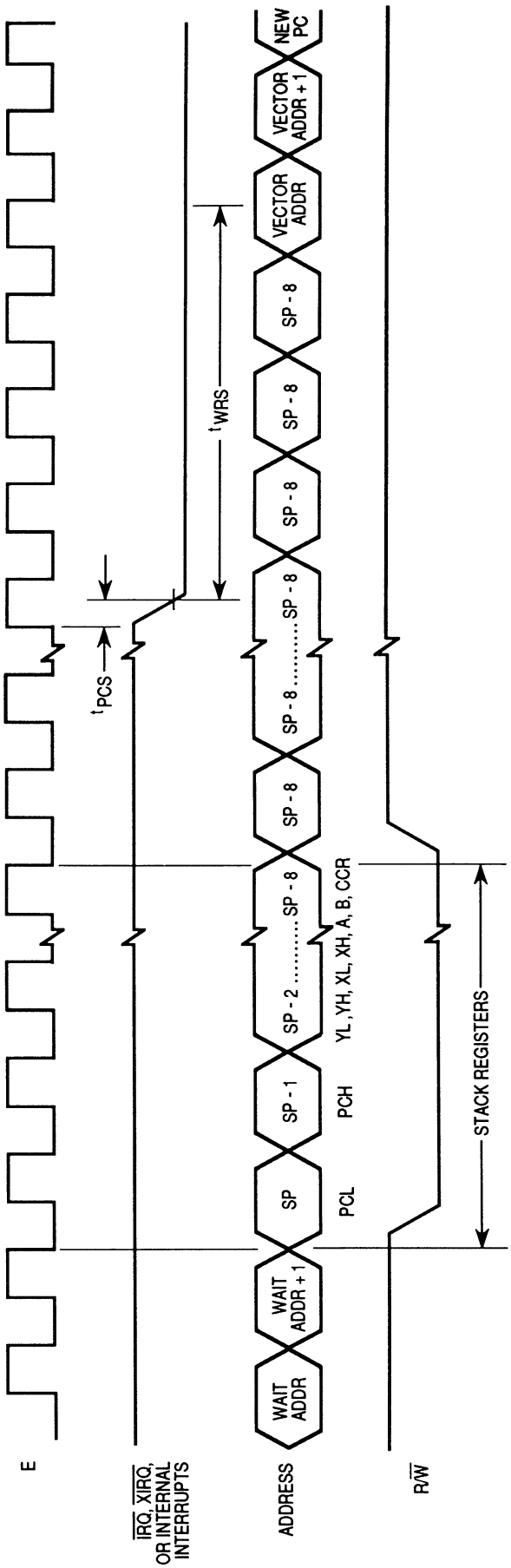


Figure 11-3. POR External Reset Timing Diagram



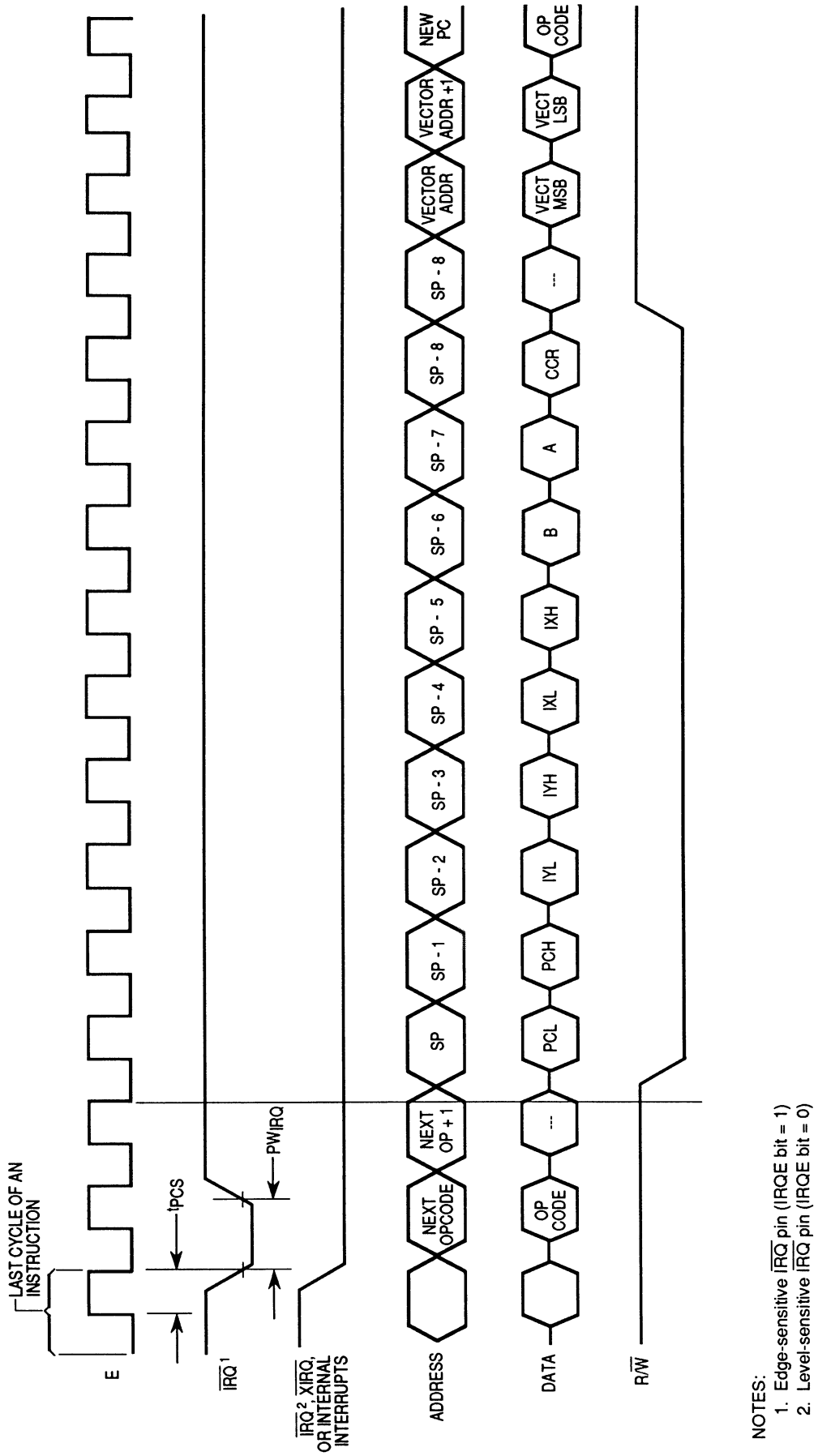
NOTES:
1. Edge-sensitive IRQ pin (IRQE bit = 1)
2. Level-sensitive IRQ pin (IRQE bit = 0)
3. tSTOPDELAY = 4064 t cyc if DLY bit = 1 or 4 t cyc if DLY = 0
4. XIRQ with X bit in CCR = 1
5. IRQ or (XIRQ with X bit in CCR = 0)

Figure 11-4. STOP Recovery Timing Diagram



NOTE: $\overline{\text{RESET}}$ will also cause recovery from WAIT.

Figure 11-5. WAIT Recovery from Interrupt Timing Diagram



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

Figure 11-6. Interrupt Timing Diagram

11.6 PERIPHERAL PORT TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation (E-Clock Frequency)	f_o	1.0	1.0	2.0	2.0	2.1	2.1	MHz
E-Clock Period	t_{cyc}	1000	—	500	—	476	—	ns
Peripheral Data Setup Time (MCU Read of Ports A, B, C, and D)	t_{PDSU}	100	—	100	—	100	—	ns
Peripheral Data Hold Time (MCU Read of Ports A, B, C, and D)	t_{PDH}	50	—	50	—	50	—	ns
Delay Time, Peripheral Data Write (MCU Writes to Port A)	t_{PWD}	—	200	—	200	—	200	ns
(MCU Writes to Ports B, C, and D) $t_{PWD} = 1/4\ t_{cyc} + 100$ ns		—	350	—	225	—	219	

NOTES:
1. Port C and D timing is valid for active drive (DWOM bit not set in SPCR register, and CWOM bit not set in PIOC register).
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

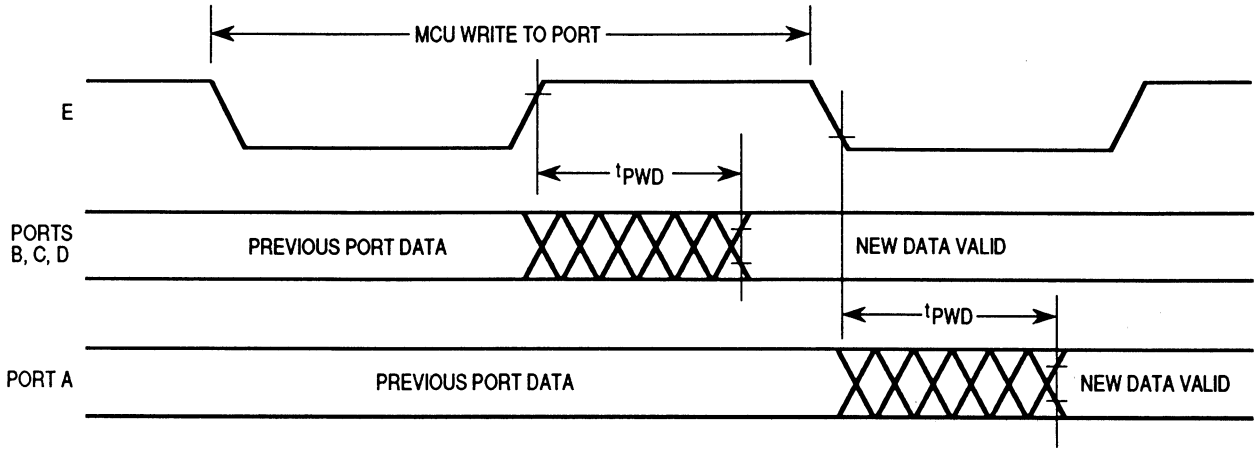


Figure 11-7. Port Write Timing Diagram

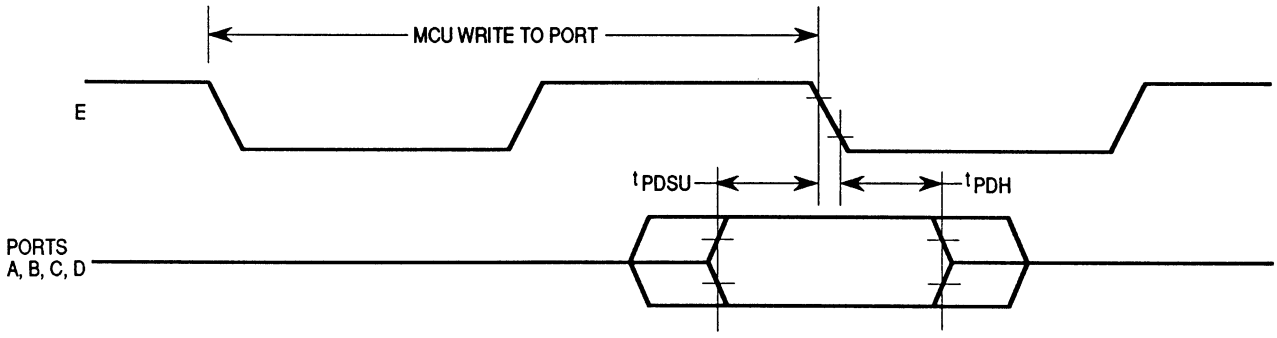


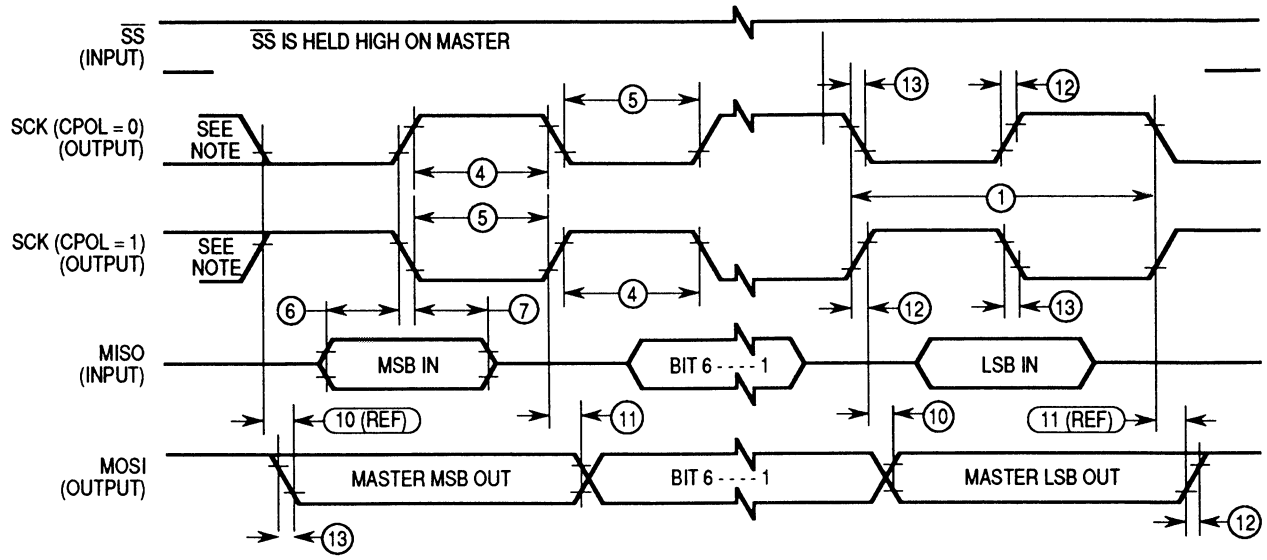
Figure 11-8. Port Read Timing Diagram

11.7 SERIAL PERIPHERAL INTERFACE TIMING

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

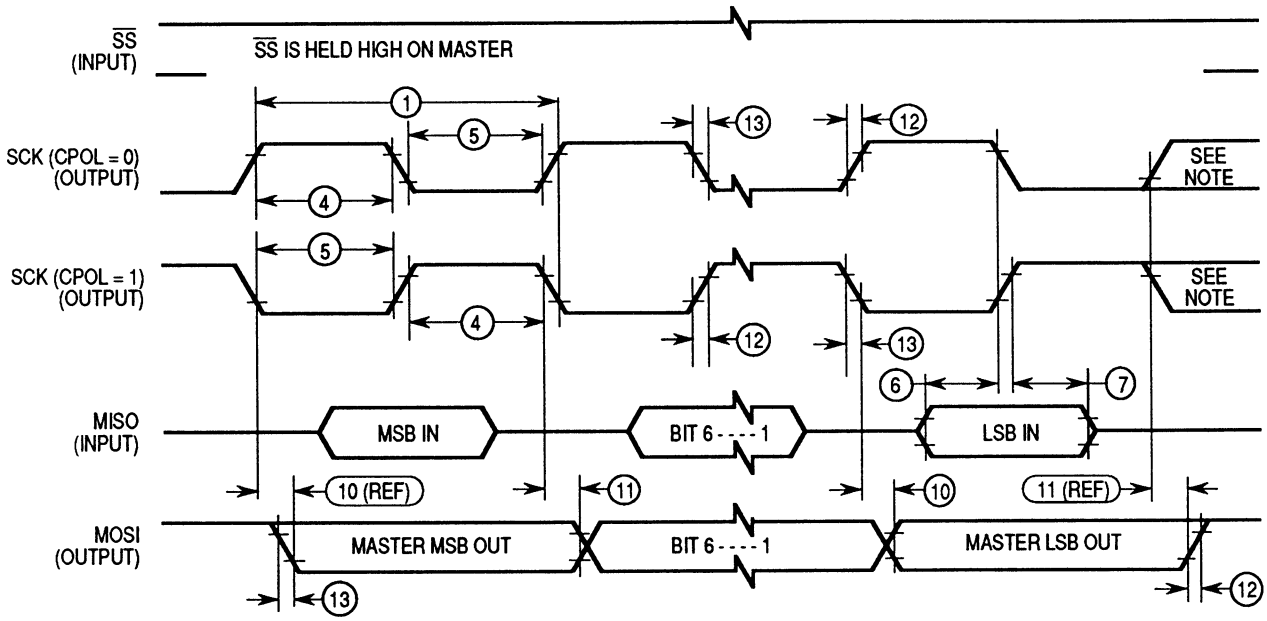
Num	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	 $f_{op(m)}$ $f_{op(s)}$	 dc dc	 0.5 2.1	 f_{op} MHz
1	Cycle Time Master Slave	 $t_{cyc(m)}$ $t_{cyc(s)}$	 2.0 480	 — —	 t_{cyc} ns
2	Enable Lead Time Master Slave	 $t_{lead(m)}$ $t_{lead(s)}$	 * 240	 — —	 ns ns
3	Enable Lag Time Master Slave	 $t_{lag(m)}$ $t_{lag(s)}$	 * 240	 — —	 ns ns
4	Clock (SCK) High Time Master Slave	 $t_{w(SCKH)m}$ $t_{w(SCKH)s}$	 340 190	 — —	 ns ns
5	Clock (SCK) Low Time Master Slave	 $t_{w(SCKL)m}$ $t_{w(SCKL)s}$	 340 190	 — —	 ns ns
6	Data Setup Time (Inputs) Master Slave	 $t_{su(m)}$ $t_{su(s)}$	 100 100	 — —	 ns ns
7	Data Hold Time (Inputs) Master Slave	 $t_{h(m)}$ $t_{h(s)}$	 100 100	 — —	 ns ns
8	Access Time (Time to Data Active from High-Impedance State) Slave	 t_a	 0	 120	 ns
9	Disable Time (Hold Time to High-Impedance State) Slave	 t_{dis}	 —	 240	 ns
10	Data Valid (After Enable Edge)**	$t_{v(s)}$	—	240	ns
11	Data Hold Time (Outputs) (After Enable Edge)	t_{ho}	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 SS)	 t_{rm} t_{rs}	 — —	 100 2.0	 ns µs
13	Fall Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{fm} t_{fs}	 — —	 100 2.0	 ns µs

*Signal product depends on software.
**Assumes 200 pF load on all SPI pins.
NOTE: All timing is shown with respect 20% V_{DD} and 70% V_{DD} unless otherwise noted.



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

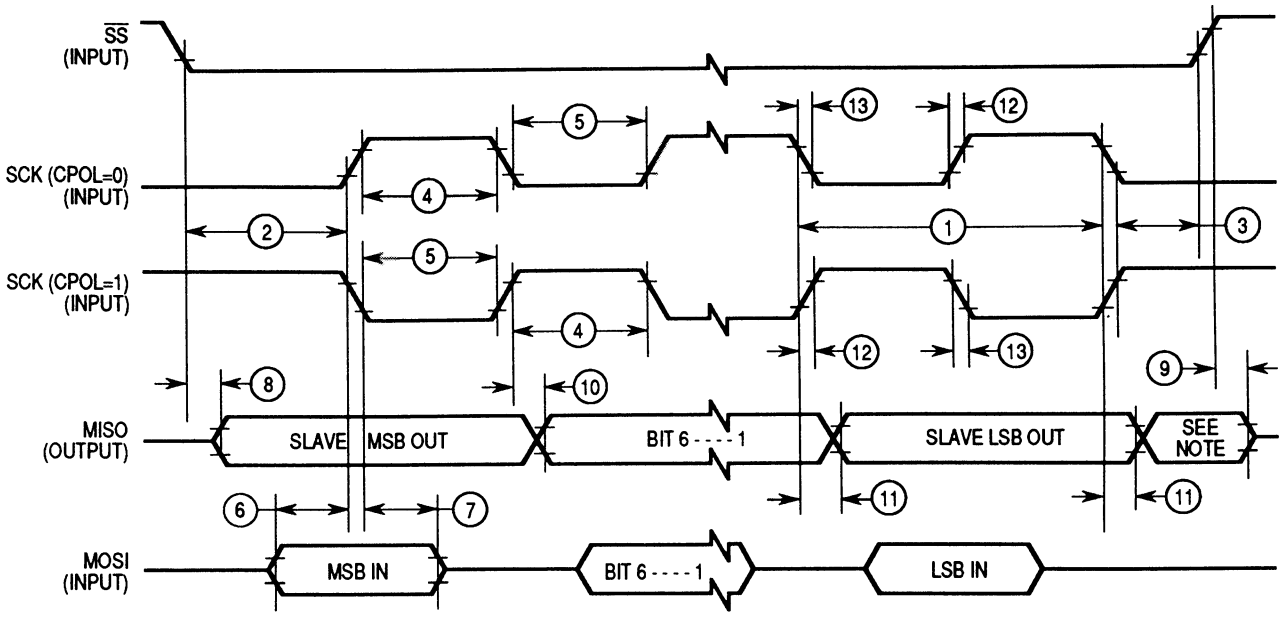
(a) SPI Master Timing (CPHA = 0)



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

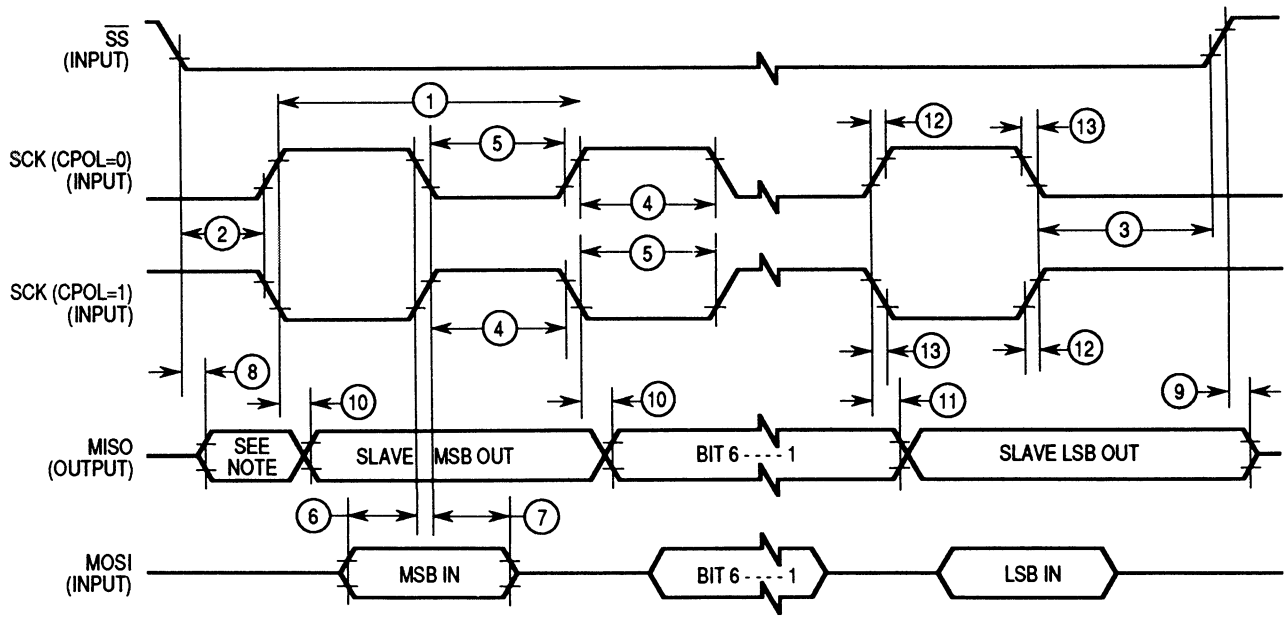
(b) SPI Master Timing (CPHA = 1)

Figure 11-9. SPI Timing Diagrams (Sheet 1 of 2)



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

(c) SPI Slave Timing (CPHA = 0)



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

(d) SPI Slave Timing (CPHA = 1)

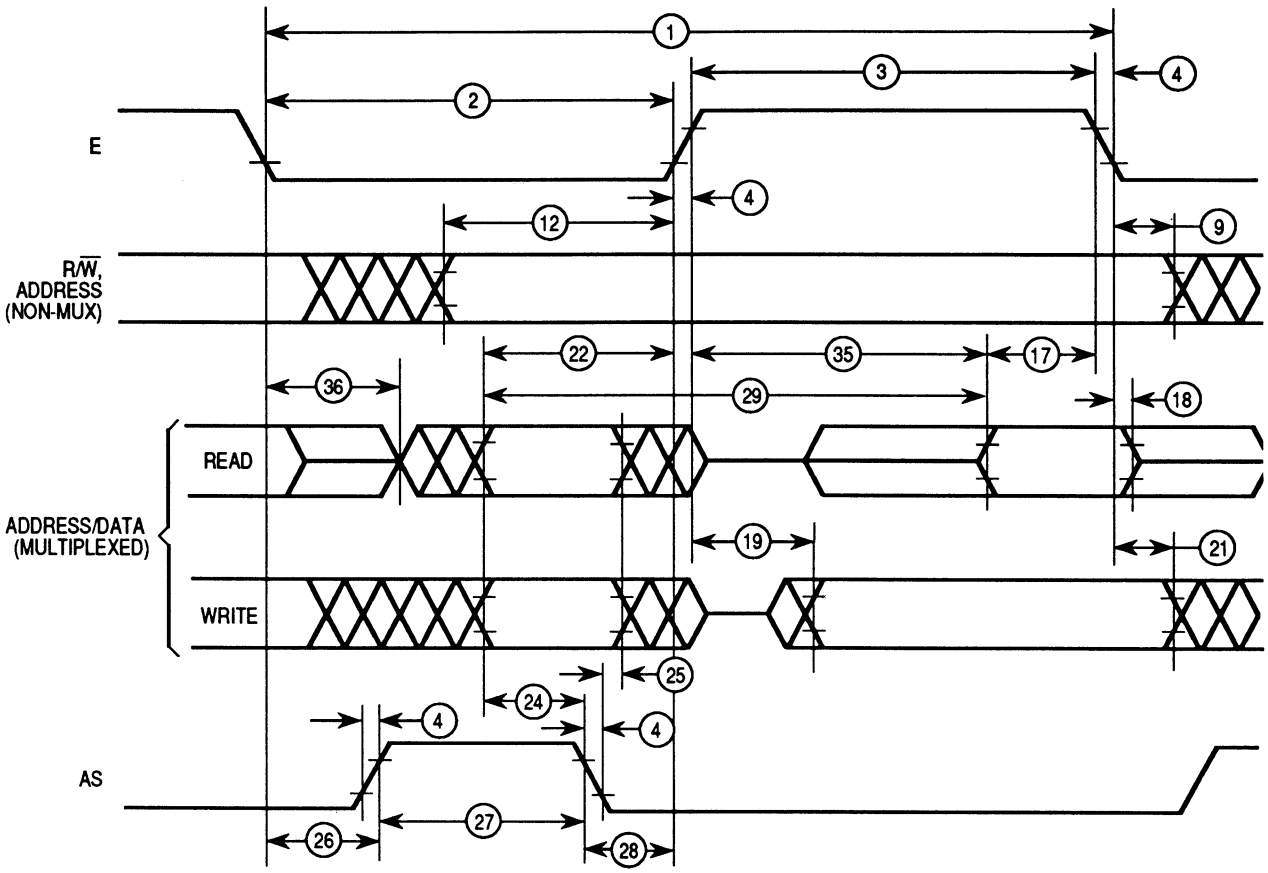
Figure 11-9. SPI Timing Diagrams (Sheet 2 of 2)

11.8 EXPANSION BUS TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
			Min	Max	Min	Max	Min	Max	
	Frequency of Operation (E Clock Frequency)	f_o	1.0	1.0	2.0	2.0	2.1	2.1	MHz
1	Cycle Time	t_{cyc}	1000	—	500	—	476	—	ns
2	Pulse Width, E Low (PW_{EL} = 1/2 t_{cyc} − 23 ns)	PW_{EL}	477	—	227	—	215	—	ns
3	Pulse Width, E High (PW_{EH} = 1/2 t_{cyc} − 28 ns)	PW_{EH}	472	—	222	—	210	—	ns
4	E and AS Rise and Fall Time	t_r, t_f	—	20	—	20	—	20	ns
9	Address Hold Time (t_{AH} = 1/8 t_{cyc} − 29.5 ns) (see Note 1a)	t_{AH}	95.5	—	33	—	30	—	ns
12	Non-Muxed Address Valid Time to E Rise ($t_{AV} = PW_{EL} - (t_{ASD} + 80$ ns) (see Note 1a)	t_{AV}	281.5	—	94	—	85	—	ns
17	Read Data Setup Time	t_{DSR}	30	—	30	—	30	—	ns
18	Read Data Hold Time (Max = t_{MAD})	t_{DHR}	10	145.5	10	83	10	80	ns
19	Write Data Delay Time (t_{DDW} = 1/8 t_{cyc} + 65.5 ns) (see Note 1a)	t_{DDW}	—	190.5	—	128	—	125	ns
21	Write Data Hold Time (t_{DHW} = 1/8 t_{cyc} − 29.5 ns) (see Note 1a)	t_{DHW}	95.5	—	33	—	30	—	ns
22	Muxed Address Valid Time to E Rise ($t_{AVM} = PW_{EL} - (t_{ASD} + 90$ ns) (see Note 1a)	t_{AVM}	271.5	—	84	—	75	—	ns
24	Muxed Address Valid Time to AS Fall ($t_{ASL} = PW_{ASH} - 70$ ns)	t_{ASL}	151	—	26	—	20	—	ns
25	Muxed Address Hold Time (t_{AHL} = 1/8 t_{cyc} − 29.5 ns) (see Note 1b)	t_{AHL}	95.5	—	33	—	30	—	ns
26	Delay Time, E to AS Rise (t_{ASD} = 1/8 t_{cyc} − 9.5 ns) (see Note 1a)	t_{ASD}	115.5	—	53	—	50	—	ns
27	Pulse Width, AS High (PW_{ASH} = 1/4 t_{cyc} − 29 ns)	PW_{ASH}	221	—	96	—	90	—	ns
28	Delay Time, AS to E Rise (t_{ASED} = 1/8 t_{cyc} − 9.5 ns) (see Note 1b)	t_{ASED}	115.5	—	53	—	50	—	ns
29	MPU Address Access Time ($t_{ACCA} = t_{AVM} + t_r + PW_{EH} - t_{DSR}$) (see Note 1a)	t_{ACCA}	733.5	—	296	—	275	—	ns
35	MPU Access Time ($t_{ACCE} = PW_{EH} - t_{DSR}$)	t_{ACCE}	—	442	—	192	—	180	ns
36	Muxed Address Delay (Previous Cycle MPU Read) ($t_{MAD} = t_{ASD} + 30$ ns) (see Note 1a)	t_{MAD}	145.5	—	83	—	80	—	ns

NOTES:

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



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

Figure 11-10. Expansion Bus Timing Diagram

SECTION 12
MECHANICAL DATA

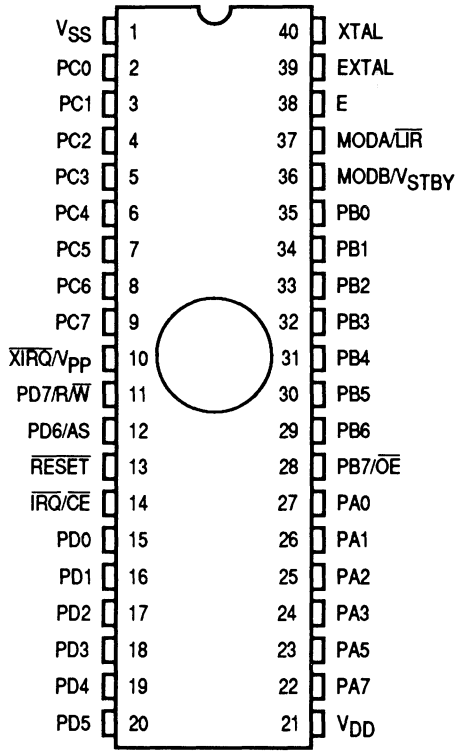
The following section contains the pin assignments, package dimensions, and ordering information for the MC68HC711D3 MCU.

12.1 ORDERING INFORMATION

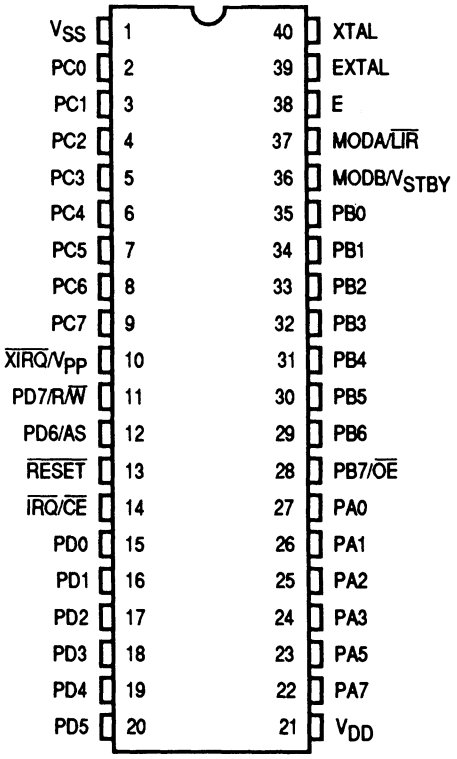
Package Type	Temperature	MC Order Number	
		Ceramic (EPROM)	Plastic (OTPROM)
40-Pin DIP	−40°C to +85°C	MC68HC711D3S	MC68HC711D3P
44-Pin Quad	−40°C to +85°C		MC68HC711D3FN

12.2 PIN ASSIGNMENTS

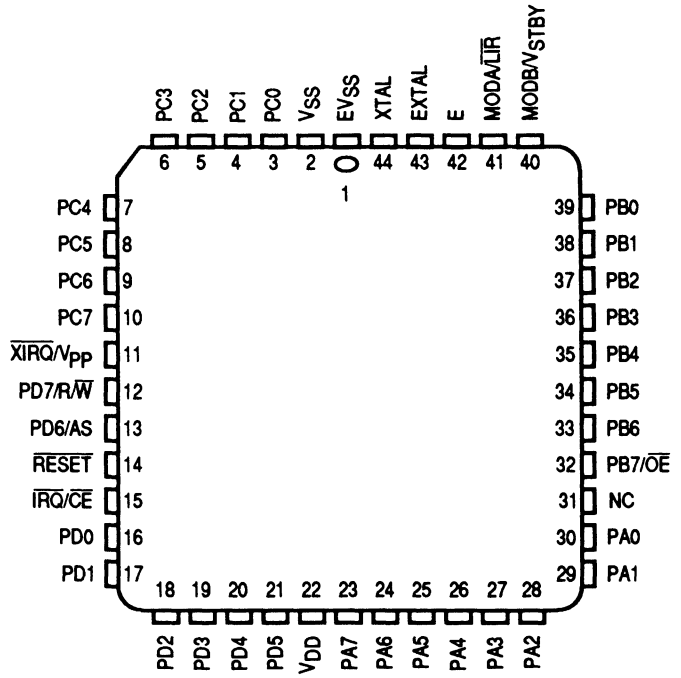
12.2.1 40-Pin Cerdip



12.2.2 40-Pin Plastic DIP

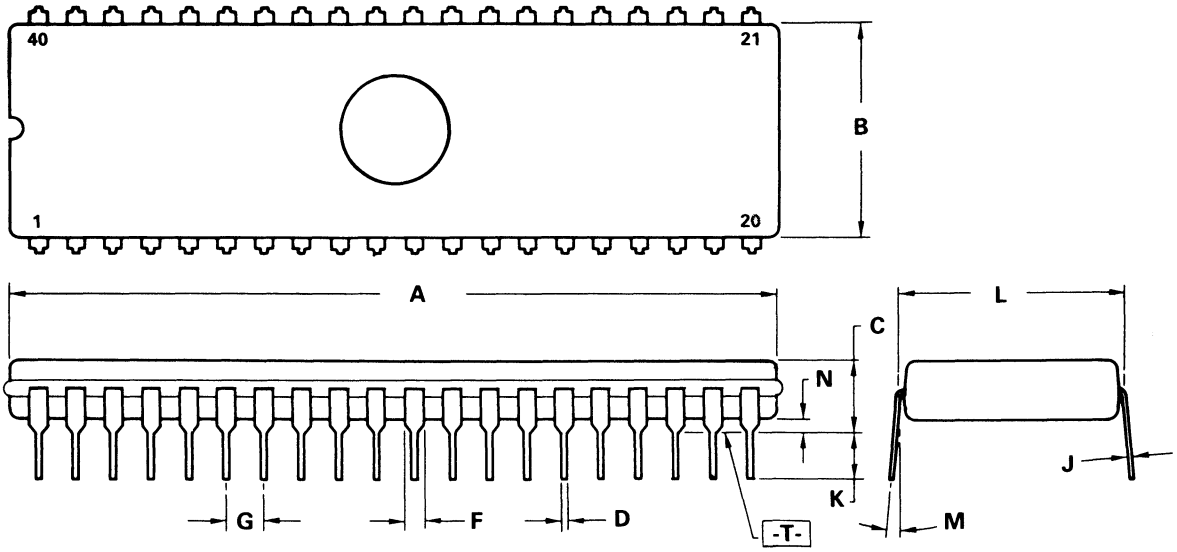


12.2.3 44-Pin Plastic PLCC



12.3 PACKAGE DIMENSIONS

S SUFFIX
CERDIP PACKAGE
CASE 734A-01



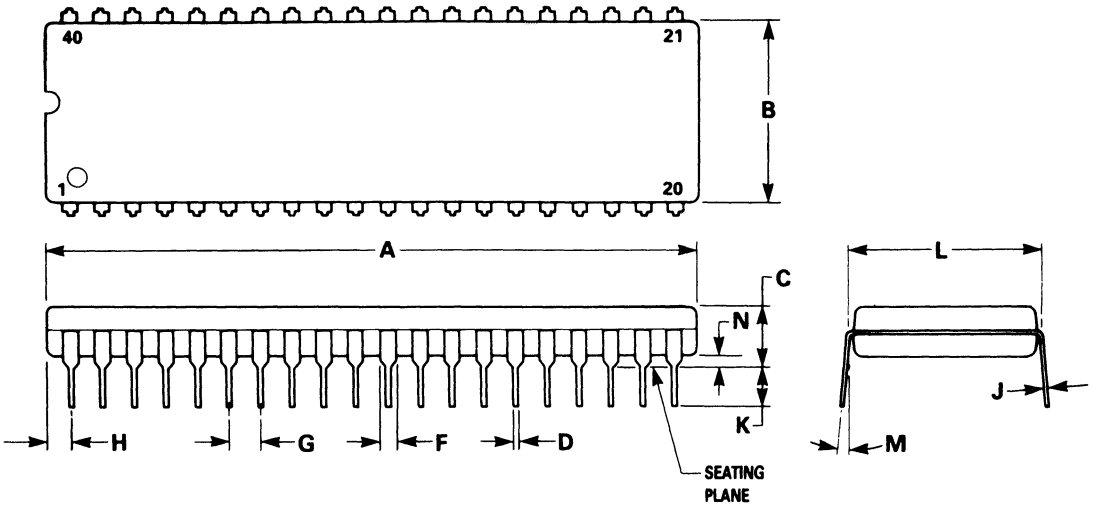
NOTES:
1. DIMENSION "A" IS A DATUM. T IS BOTH A DATUM AND A SEATING PLANE.
2. POSITIONAL TOLERANCE FOR LEADS: (40 PLACES)

⌖	ϕ 0.25 (0.010) Ⓜ	T	A Ⓜ

3. DIMENSIONS A & B INCLUDE MENISCUS.
4. DIMENSIONS L TO CENTER OF LEADS WHEN FORMED PARALLEL.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 982.
6. CONTROLLING DIMENSION: INCH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	51.31	53.23	2.020	2.096
B	12.70	15.94	0.500	0.610
C	4.06	6.09	0.160	0.240
D	0.38	0.55	0.015	0.022
F	1.27	1.65	0.050	0.065
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	3.17	4.06	0.125	0.160
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.27	0.020	0.050

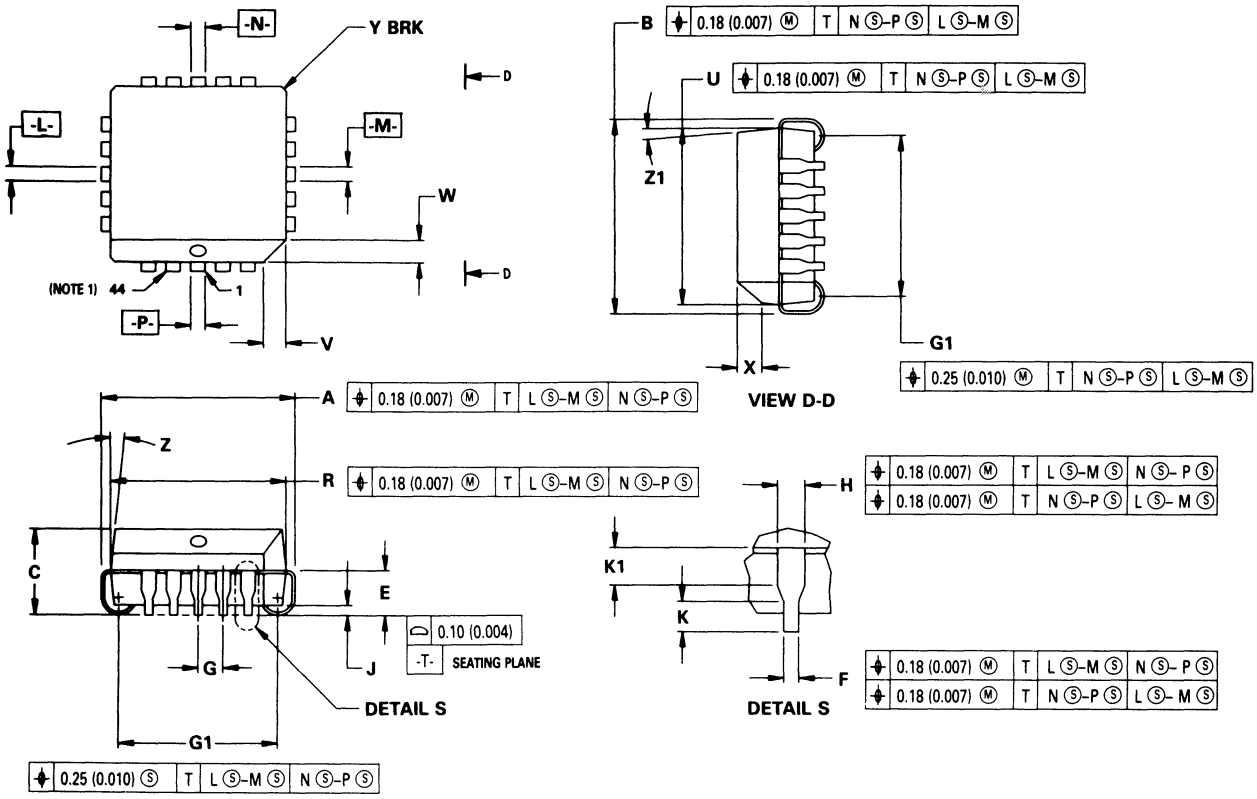
P SUFFIX
PLASTIC PACKAGE
CASE 711-03



NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 mm (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

FN SUFFIX
PLASTIC LEADED CHIP CARRIER
CASE 777-02



DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	17.40	17.65	0.685	0.695
B	17.40	17.65	0.685	0.695
C	4.20	4.57	0.165	0.180
E	2.29	2.79	0.090	0.110
F	0.33	0.48	0.013	0.019
G	1.27 BSC		0.050 BSC	
H	0.66	0.81	0.026	0.032
J	0.51	—	0.020	—
K	0.64	—	0.025	—
R	16.51	16.66	0.650	0.656
U	16.51	16.66	0.650	0.656
V	1.07	1.21	0.042	0.048
W	1.07	1.21	0.042	0.048
X	1.07	1.42	0.042	0.056
Y	—	0.50	—	0.020
Z	2°	10°	2°	10°
G1	15.50	16.00	0.610	0.630
K1	1.02	—	0.040	—
Z1	2°	10°	2°	10°

NOTES:
1. DUE TO SPACE LIMITATION, CASE 777-02 SHALL BE REPRESENTED BY A GENERAL (SMALLER) CASE OUTLINE DRAWING RATHER THAN SHOWING ALL 44 LEADS.
2. DATUMS -L-, -M-, -N-, AND -P- DETERMINED WHERE TOP OF LEAD SHOULDER EXIT PLASTIC BODY AT MOLD PARTING LINE.
3. DIM G1, TRUE POSITION TO BE MEASURED AT DATUM -T-, SEATING PLANE.
4. DIM R AND U DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE MOLD PROTRUSION IS 0.25 (0.010) PER SIDE.
5. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
6. CONTROLLING DIMENSION: INCH.

NOTES

NOTES

Mfax is a trademark of Motorola, Inc.

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

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

INTERNET: http://motorola.com/sps

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

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

MOTOROLA

MC68HC711D3/D

1ATX22547-4 PRINTED IN USA 7/97 IMPERIAL LITHO 30834 3,500 LITAMCU