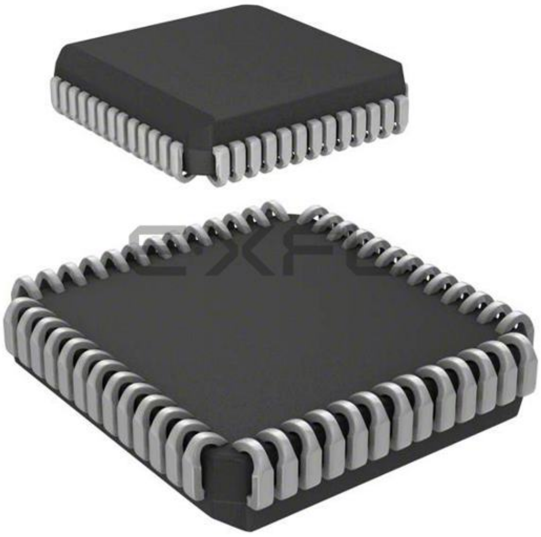
NXP USA Inc. - MC68HC11A1CFNE3R Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	HC11
Core Size	8-Bit
Speed	3MHz
Connectivity	SCI, SPI
Peripherals	POR, WDT
Number of I/O	38
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	512 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x8b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	52-LCC (J-Lead)
Supplier Device Package	52-PLCC (19.1x19.1)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc11a1cfne3r

4 PARALLEL I/O

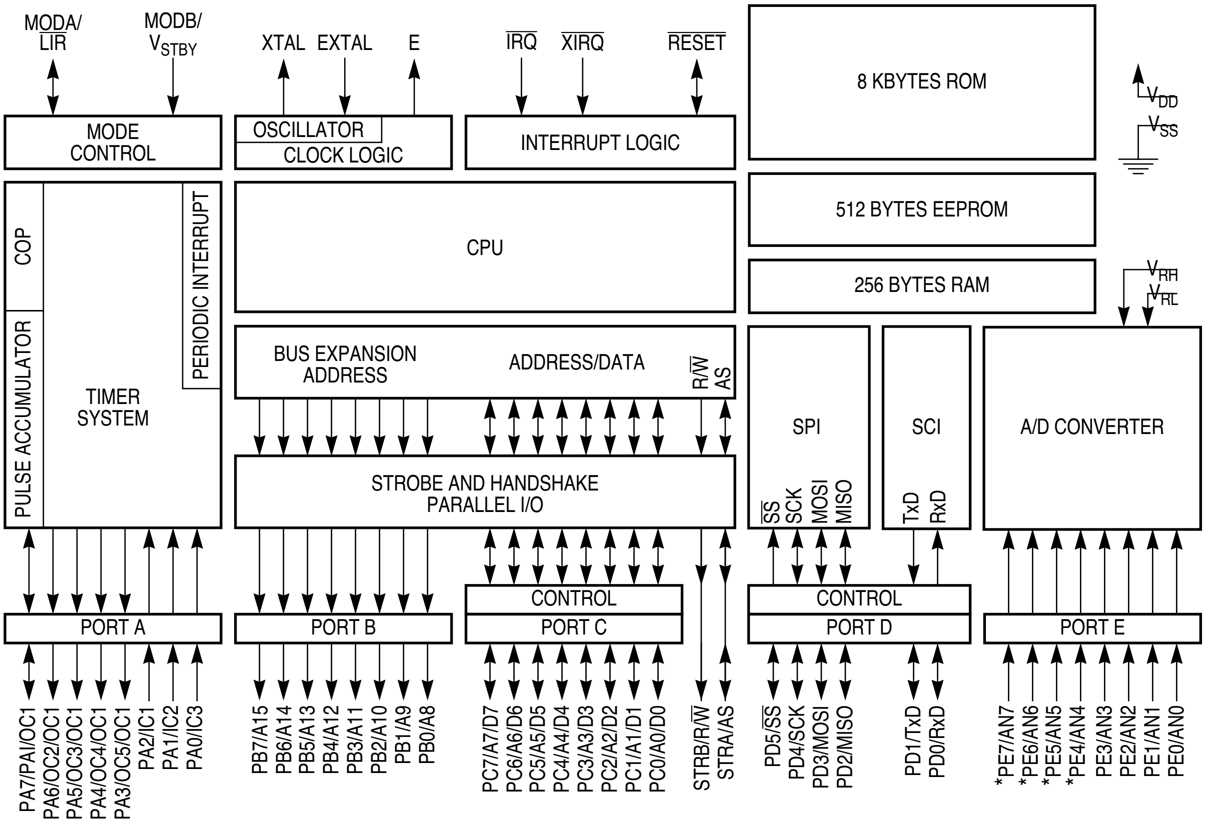
5 SERIAL COMMUNICATIONS INTERFACE

6 SERIAL PERIPHERAL INTERFACE

(SCI) and a separate synchronous serial peripheral interface (SPI) are included. The main 16-bit free-running timer system has three input capture lines, five output compare lines, and a real-time interrupt function. An 8-bit pulse accumulator subsystem can count external events or measure external periods.

Self monitoring circuitry is included on-chip to protect against system errors. A computer operating properly (COP) watchdog system protects against software failures. A clock monitor system generates a system reset in case the clock is lost or runs too slow. An illegal opcode detection circuit provides a non-maskable interrupt if an illegal opcode is detected.

Two software controlled operating modes, WAIT and STOP, are available to conserve additional power.



* NOT BONDED ON 48-PIN VERSION.

A8 BLOCK

Figure 1-1 Block Diagram

1.3 Programmer’s Model

In addition to being able to execute all M6800 and M6801 instructions, the MC68HC11A8 allows execution of 91 new opcodes. **Figure 1-2** shows the seven CPU registers which are available to the programmer.

2 SIGNAL DESCRIPTIONS AND OPERATING MODES

The signal descriptions and operating modes are presented in this section. When the microcontroller is in an expanded multiplexed operating mode, 18 pins change function to support a multiplexed address/data bus.

2.1 Signal Pin Descriptions

The following paragraphs provide a description of the input/output signals. Reference is made, where applicable, to other sections that contain more detail about the function being performed.

2.1.1 Input Power (V_{DD}) and Ground (V_{SS})

Power is supplied to the microcontroller using these pins. V_{DD} is the positive power input and V_{SS} is ground. Although the MC68HC11A8 is a CMOS device, very fast signal transitions are present on many of its pins. Short rise and fall times are present even when the microcontroller is operating at slow clock rates. Special care must be taken to provide good power supply bypassing at the MCU. Recommended bypassing would include a 0.1 µF ceramic capacitor between the V_{DD} and V_{SS} pins and physically adjacent to one of the two pins. A bulk capacitance, whose size depends on the other circuitry in the system, should also be present on the circuit board.

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

This active low bidirectional control signal is used as an input to initialize the MC68HC11A8 to a known start-up state, and as an open-drain output to indicate that an internal failure has been detected in either the clock monitor or computer operating properly (COP) watchdog circuit. This reset signal is significantly different from the reset signal used on other Motorola MCUs. Please refer to **9 RESETS, INTERRUPTS, AND LOW POWER MODES** before designing circuitry to generate or monitor this signal.

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

These two pins provide the interface for either a crystal or a CMOS compatible clock to control the internal clock generator circuitry. The frequency applied to these pins shall be four times higher than the desired E clock rate. The XTAL pin is normally left unterminated when using an external CMOS compatible clock input to the EXTAL pin. However, a 10K to 100K load resistor to ground may be used to reduce RFI noise emission. The XTAL output is normally intended to drive only a crystal.

The XTAL output may be buffered with a high-input-impedance buffer such as the 74HC04, or it may be used to drive the EXTAL input of another M68HC11.

In all cases take extra care in the circuit board layout around the oscillator pins. Load capacitances shown in the oscillator circuits include all stray layout capacitances. Refer to **Figure 2-1**, **Figure 2-2**, and **Figure 2-3** for diagrams of oscillator circuits.

Table 3-1 Register and Control Bit Assignments (Sheet 1 of 2)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$1000	Bit 7	–	–	–	–	–	–	Bit 0	PORTA	I/O Port A
$1001									Reserved	
$1002	STAF	STAI	CWOM	HNDS	OIN	PLS	EGA	INVB	PIOC	Parallel I/O Control Register
$1003	Bit 7	–	–	–	–	–	–	Bit 0	PORTC	I/O Port C
$1004	Bit 7	–	–	–	–	–	–	Bit 0	PORTB	Output Port B
$1005	Bit 7	–	–	–	–	–	–	Bit 0	PORTCL	Alternate Latched Port C
$1006									Reserved	
$1007	Bit 7	–	–	–	–	–	–	Bit 0	DDRC	Data Direction for Port C
$1008			Bit 5	–	–	–	–	Bit 0	PORTD	I/O Port D
$1009			Bit 5	–	–	–	–	Bit 0	DDRD	Data Direction for Port D
$100A	Bit 7	–	–	–	–	–	–	Bit 0	PORTE	Input Port E
$100B	FOC1	FOC2	FOC3	FOC4	FOC5				CFORC	Compare Force Register
$100C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3				OC1M	OC1 Action Mask Register
$100D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3				OC1D	OC1 Action Data Register
$100E	Bit 15	–	–	–	–	–	–	Bit 8	TCNT	Timer Counter Register
$100F	Bit 7	–	–	–	–	–	–	Bit 0		
$1010	Bit 15	–	–	–	–	–	–	Bit 8	TIC1	Input Capture 1 Register
$1011	Bit 7	–	–	–	–	–	–	Bit 0		
$1012	Bit 15	–	–	–	–	–	–	Bit 8	TIC2	Input Capture 2 Register
$1013	Bit 7	–	–	–	–	–	–	Bit 0		
$1014	Bit 15	–	–	–	–	–	–	Bit 8	TIC3	Input Capture 3 Register
$1015	Bit 7	–	–	–	–	–	–	Bit 0		
$1016	Bit 15	–	–	–	–	–	–	Bit 8	TOC1	Output Compare 1 Register
$1017	Bit 7	–	–	–	–	–	–	Bit 0		
$1018	Bit 15	–	–	–	–	–	–	Bit 8	TOC2	Output Compare 2 Register
$1019	Bit 7	–	–	–	–	–	–	Bit 0		
$101A	Bit 15	–	–	–	–	–	–	Bit 8	TOC3	Output Compare 3 Register
$101B	Bit 7	–	–	–	–	–	–	Bit 0		
$101C	Bit 15	–	–	–	–	–	–	Bit 8	TOC4	Output Compare 4 Register
$101D	Bit 7	–	–	–	–	–	–	Bit 0		
$101E	Bit 15	–	–	–	–	–	–	Bit 8	TCO5	Output Compare 5 Register
$101F	Bit 7	–	–	–	–	–	–	Bit 0		

4.4.1 Input Handshake Protocol

In the input handshake protocol, port C is a latching input port, STRA is an edge-sensitive latch command from the external system that is driving port C, and STRB is a “ready” output line controlled by logic in the MCU.

When a “ready” condition is recognized, the external device places data on the port C lines, then pulses the STRA line. The active edge on the STRA line latches the port C data into the PORTCL register, sets the STAF flag (optionally causing an interrupt), and deasserts the STRB line. Deassertion of the STRB line automatically inhibits the external device from strobing new data into port C. Reading the PORTCL latch register (independent of clearing the STAF flag) asserts the STRB line, indicating that new data may now be applied to port C.

The STRB line can be configured (with the PLS control bit) to be a pulse output (pulse mode) or a static output (interlocked mode).

The port C data direction register bits should be cleared for each line that is to be used as a latched input line. However, some port C lines can be used as latched inputs with the input handshake protocol while, at the same time, using some port C lines as static inputs, and some port C lines as static outputs. The input handshake protocol has no effect on the use of port C lines as static inputs or as static outputs. Reads of the PORTC data register always return the static logic level at the port C lines (for lines configured as inputs). Writes to either the PORTC data register or the alternate latched port C register (PORTCL) send information to the same port C output register without affecting the input handshake strobes.

4.4.2 Output Handshake Protocol

In the output handshake protocol, port C is an output port, STRB is a “ready” output, and STRA is an edge-sensitive acknowledge input signal, used to indicate to the MCU that the output data has been accepted by the external device. In a variation of this output handshake protocol, STRA is also used as an output-enable input, as well as an edge-sensitive acknowledge input.

The MCU places data on the port C output lines and then indicates stable data is available by asserting the STRB line. The external device then processes the available data and pulses the STRA line to indicate that new data may be placed on the port C output lines. The active edge on the STRA line causes the STRB line to be deasserted and the STAF status flag to be set. In response to the STAF bit being set, the program transfers new data out of port C as required. Writing data to the PORTCL register causes the data to appear on port C lines and asserts the STRB line.

There is a variation to the output handshake protocol that allows three-state operation on port C. It is possible to directly connect this 8-bit parallel port to other three-state devices with no additional parts.

While the STRA input line is inactive, all port C lines obey the data direction specified by the data direction register so that lines which are configured as inputs are high impedance. When the STRA line is activated, all port C lines are forced to outputs regardless of the data in the data direction register. Note that in output handshake

TCIE — Transmit Complete Interrupt Enable
0 = TC interrupts disabled
1 = SCI Interrupt if TC = 1

RIE — Receive Interrupt Enable
0 = RDRF and OR interrupts disabled
1 = SCI interrupt if RDRF or OR = 1

ILIE — Idle Line Interrupt Enable
0 = IDLE interrupts disabled
1 = SCI interrupt if IDLE = 1

TE — Transmit Enable

When the transmit enable bit is set, the transmit shift register output is applied to the TxD line. Depending on the state of control bit M (SCCR1), a preamble of 10 (M = 0) or 11 (M = 1) consecutive ones is transmitted when software sets the TE bit from a cleared state. After loading the last byte in the serial communications data register and receiving the TDRE flag, the user can clear TE. Transmission of the last byte will then be completed before the transmitter gives up control of the TxD pin. While the transmitter is active, the data direction register control for port D bit 1 is overridden and the line is forced to be an output.

RE — Receive Enable

When the receive enable bit is set, the receiver is enabled. When RE is clear, the receiver is disabled and all of the status bits associated with the receiver (RDRF, IDLE, OR, NF, and FE) are inhibited. While the receiver is enabled, the data direction register control for port D bit 0 is overridden and the line is forced to be an input.

RWU — Receiver Wake Up

When the receiver wake-up bit is set by the user’s software, it puts the receiver to sleep and enables the “wake up” function. If the WAKE bit is cleared, RWU is cleared by the SCI logic after receiving 10 (M = 0) or 11 (M = 1) consecutive ones. If the WAKE bit is set, RWU is cleared by the SCI logic after receiving a data word whose MSB is set.

SBK — Send Break

If the send break bit is toggled set and cleared, the transmitter sends 10 (M = 0) or 11 (M = 1) zeros and then reverts to idle or sending data. If SBK remains set, the transmitter will continually send whole blocks of zeros (sets of 10 or 11) until cleared. At the completion of the break code, the transmitter sends at least one high bit to guarantee recognition of a valid start bit. If the transmitter is currently empty and idle, setting and clearing SBK is likely to queue two character times of break because the first break transfers almost immediately to the shift register and the second is then queued into the parallel transmit buffer.

Figure 6-3 illustrates the MOSI, MISO, SCK, and $\overline{\text{SS}}$ master-slave interconnections.

Due to data direction register control of SPI outputs and the port D wire-OR mode (DWOM) option, the SPI system can be configured in a variety of ways. Systems with a single bidirectional data path rather than separate MISO and MOSI paths can be accommodated. Since MC68HC11A8 SPI slaves can selectively disable their MISO output, a broadcast message protocol is also possible.

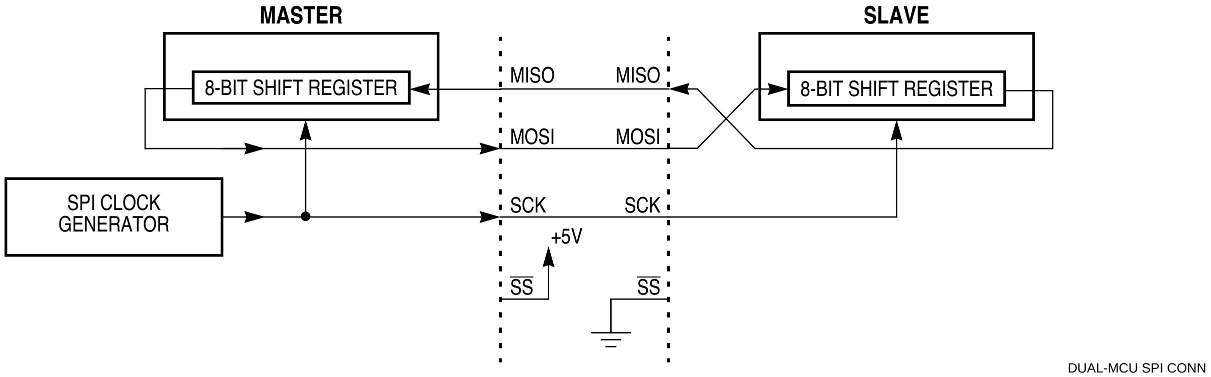


Figure 6-3 Serial Peripheral Interface Master-Slave Interconnection

6.4 SPI Registers

There are three registers in the serial peripheral interface which provide control, status, and data storage functions. These registers are called the serial peripheral control register (SPCR), serial peripheral status register (SPSR), and serial peripheral data I/O register (SPDR) and are described in the following paragraphs.

6.4.1 Serial Peripheral Control Register (SPCR)

	7	6	5	4	3	2	1	0	
$1028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
RESET	0	0	0	0	0	1	U	U	

SPIE — Serial Peripheral Interrupt Enable

0 = SPIF interrupts disabled
1 = SPI interrupt if SPIF = 1

SPE — Serial Peripheral System Enable

0 = SPI system off
1 = SPI system on

DWOM — Port D Wire-OR Mode Option

DWOM affects all six port D pins together.

0 = Port D outputs are normal CMOS outputs
1 = Port D outputs act as open-drain outputs

8.1.9 Timer Control Register 2 (TCTL2)

	7	6	5	4	3	2	1	0	
$1021	0	0	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2
RESET	0	0	0	0	0	0	0	0	

Bits 7-6 — Not Implemented

These bits always read zero.

EDGxB and EDGxA — Input Capture x Edge Control.

These two bits (EDGxB and EDGxA) are cleared to zero by reset and are encoded to configure the input sensing logic for input capture x as follows:

EDGxB	EDBxA	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

8.1.10 Timer Interrupt Mask Register 1 (TMSK1)

	7	6	5	4	3	2	1	0	
$1022	OC1I	OC2I	OC3I	OC4I	OC5I	IC1I	IC2I	IC3I	TMSK1
RESET	0	0	0	0	0	0	0	0	

OCxI — Output Compare x Interrupt

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

ICxI — Input Capture x Interrupt

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

8.1.11 Timer Interrupt Flag Register 1 (TFLG1)

Timer interrupt flag register 1 is used to indicate the occurrence of timer system events, and together with the TMSK1 register allows the timer subsystem to operate in a polled or interrupt driven system. For each bit in TFLG1, there is a corresponding bit in TMSK1 in the same bit position. If the mask bit is set, each time the conditions for the corresponding flag are met, a hardware interrupt sequence is requested as well as the flag bit being set.

These timer system status flags are cleared by writing a one to the bit positions corresponding to the flag(s) which are to be cleared. Bit manipulation instructions would be inappropriate for flag clearing because they are read-modify-write instructions. Even though the instruction mask implies that the programmer is only interested in some of the bits in the manipulated location, the entire location is actually read and rewritten which may clear other bits in the register.

PAIF — Pulse Accumulator Input Edge Interrupt Flag

This bit is set when an active edge is detected on the PAI input pin. This bit is cleared by a write to the TFLG2 register with bit 4 set.

Bits 3-0 — Not Implemented

These bits always read zero.

8.2 Real-Time Interrupt

The real-time interrupt feature on the MCU is configured and controlled by using two bits (RTR1 and RTR0) in the PACTL register to select one of four interrupt rates. The RTII bit in the TMSK2 register enables the interrupt capability. Every timeout causes the RTIF bit in TFLG2 to be set, and if RTII is set, an interrupt request is generated. After reset, one entire real time interrupt period elapses before the RTIF flag is set for the first time.

8.3 Pulse Accumulator

The pulse accumulator is an 8-bit read/write counter which can operate in either of two modes (external event counting or gated time accumulation) depending on the state of the PAMOD control bit in the PACTL register. 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 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 enabled.

The pulse accumulator uses port A bit 7 as its PAI input, but this pin also shares function as a general purpose I/O pin and as a timer output compare pin. Normally port A bit 7 would be configured as an input when being used for the pulse accumulator. Note that even when port A bit 7 is configured for output, this pin still drives the input to the pulse accumulator.

8.3.1 Pulse Accumulator Control Register (PACTL)

Four bits in this register are used to control an 8-bit pulse accumulator system and two other bits are used to select the rate for the real time interrupt system.

	7	6	5	4	3	2	1	0	
$1026	DDRA7	PAEN	PAMOD	PEDGE	0	0	RTR1	RTR0	PACTL
RESET	0	0	0	0	0	0	0	0	

DDRA7 — Data Direction for Port A Bit 7

0 = Input only
1 = Output

PAEN — Pulse Accumulator System Enable

0 = Pulse accumulator off
1 = Pulse accumulator on

bit related interrupt structure has no effect on the X bit, the external $\overline{\text{XIRQ}}$ pin remains effectively non-masked. 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 condition code register byte, but the X bit is not affected. When an X bit related interrupt occurs, both the X bit and the I bit are automatically set by hardware after stacking the condition code register. An RTI (return from interrupt) instruction restores the X and I bits to their pre-interrupt request state.

9.2.4 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 position in the resolution circuit. The first six interrupt sources are not masked by the I bit in the condition code register and have the fixed priority interrupt relationship of: reset, clock monitor fail, COP fail, illegal opcode, and $\overline{\text{XIRQ}}$. (SWI is actually an instruction and has highest priority other than reset in the sense that once the SWI opcode is fetched, no other interrupt can be honored until the SWI vector has been fetched). Each of these sources is an input to the priority resolution circuit. The highest I bit masked priority input to the resolution circuit is assigned under software control (of the HPRIO register) to be connected to any one of the remaining I bit related interrupt sources. In order to avoid timing races, the HPRIO register may only be written while the I bit related interrupts are inhibited (I bit in condition code register is a logic one). An interrupt that is assigned to this high priority position is still subject to masking by any associated control bits or the I bit in the condition code register. The interrupt vector address is not affected by assigning a source to this higher priority position.

Figure 9-4, **Figure 9-5**, and **Figure 9-6** illustrate the interrupt process as it relates to normal processing. **Figure 9-4** shows how the CPU begins from a reset and how interrupt detection relates to normal opcode fetches. **Figure 9-5** is an expansion of a block in **Figure 9-4** and shows how interrupt priority is resolved. **Figure 9-6** is an expansion of the SCI interrupt block in **Figure 9-5**. **Figure 9-6** shows the resolution of interrupt sources within the SCI subsystem.

9.2.5 Highest Priority I Interrupt Register (HPRIO)

This register is used to select one of the I bit related interrupt sources to be elevated to the highest I bit masked position in the priority resolution circuit. In addition, four miscellaneous system control bits are included in this register.

	7	6	5	4	3	2	1	0	
$**1**03C	RBOOT	SMOD	MDA	IRV	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO
RESET	—	—	—	—	0	1	0	1	

RBOOT — Read Bootstrap ROM

The read bootstrap ROM bit only has meaning when the SMOD bit is a one (special bootstrap mode or special test mode). At all other times, this bit is clear and may not be written.

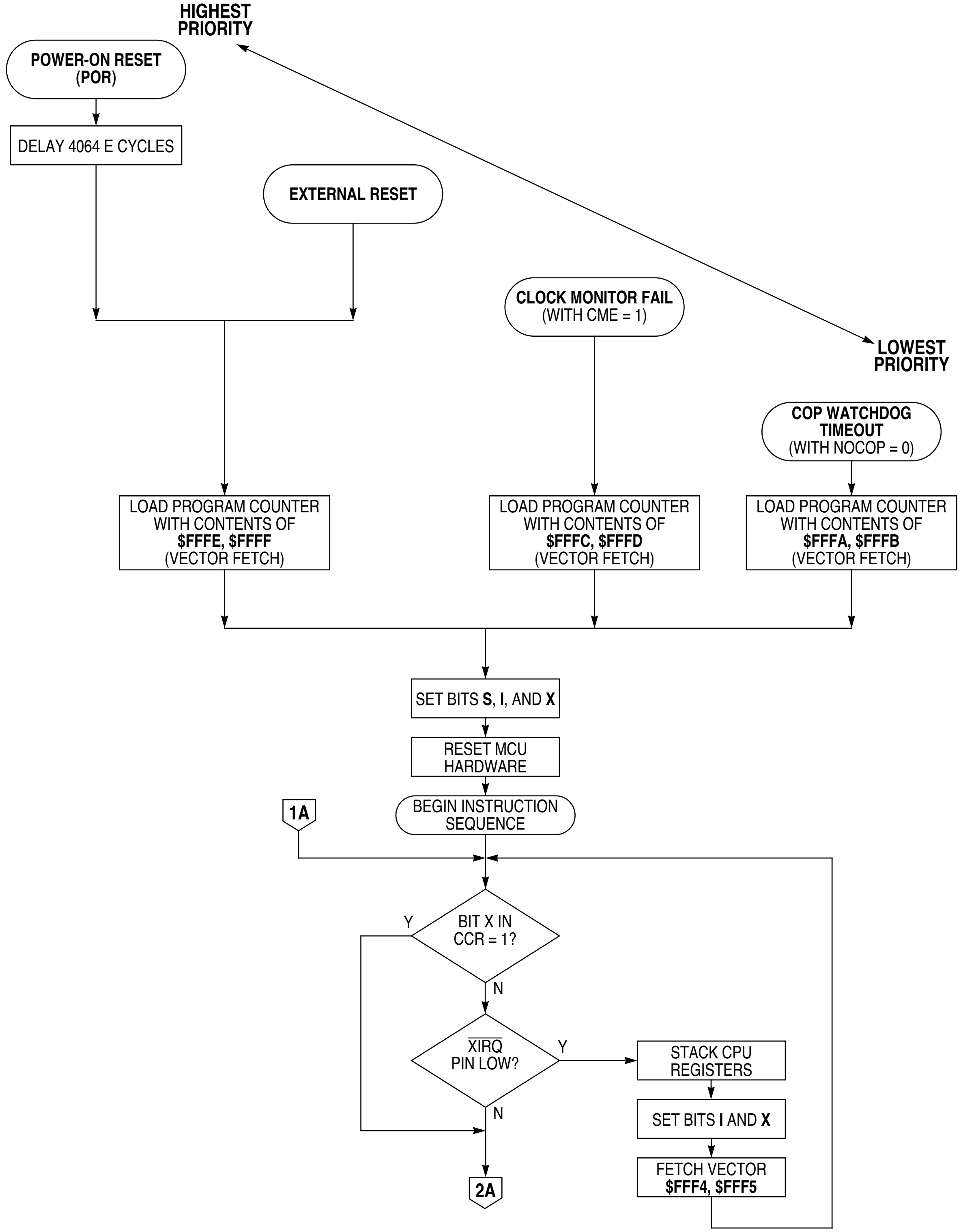

HIGHEST PRIORITY
LOWEST PRIORITY
POWER-ON RESET (POR)
DELAY 4064 E CYCLES
EXTERNAL RESET
CLOCK MONITOR FAIL (WITH CME = 1)
COP WATCHDOG TIMEOUT (WITH NOCOP = 0)
LOAD PROGRAM COUNTER WITH CONTENTS OF $FFFE, $FFFF (VECTOR FETCH)
LOAD PROGRAM COUNTER WITH CONTENTS OF $FFFC, $FFFD (VECTOR FETCH)
LOAD PROGRAM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

Figure 9-4 Processing Flow Out of Resets (Sheet 1 of 2)

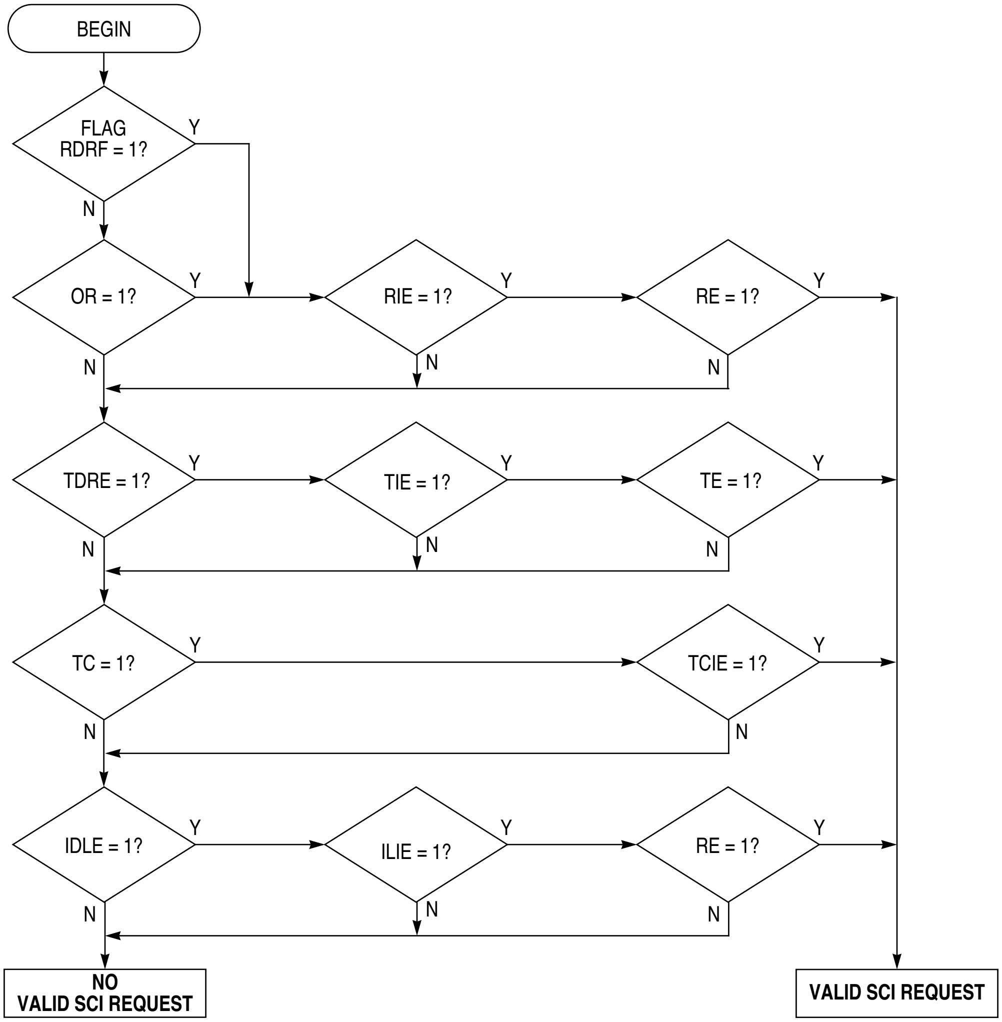


Figure 9-6 Interrupt Source Resolution Within SCI

Table 10-5 Cycle-by-Cycle Operation — Extended Mode (Sheet 1 of 2)

Reference Number*	Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{\text{W}}$ Line	Data Bus
5-1	JMP	3	1 2 3	Opcode Address Opcode Address + 1 Opcode Address + 2	1 1 1	Opcode ($7E) Jump Address (High Byte) Jump Address (Low Byte)
5-2	ADCA, ADCB, ADDA, ADDB, ANDA, ANDB, BITA, BITB, CMPA, CMPB, EORA, EORB, LDAA, LDAB, ORAA ORAB, SBCA, SBCB, SUBA, SUBB	4	1 2 3 4	Opcode Address Opcode Address + 1 Opcode Address + 2 Operand Address	1 1 1 1	Opcode Operand Address (High Byte) Operand Address (Low Byte) Operand Data
5-3	STAA, STAB	4	1 2 3 4	Opcode Address Opcode Address + 1 Opcode Address + 2 Operand Address	1 1 1 0	Opcode Operand Address (High Byte) Operand Address (Low Byte) Accumulator Data
5-4	LDD, LDS, LDX	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Opcode Address + 2 Operand Address Operand Address + 1	1 1 1 1 1	Opcode Operand Address (High Byte) Operand Address (Low Byte) Operand Data (High Byte) Operand Data (Low Byte)
5-5	STD, STS, STX	5	1 2 3 4 5	Opcode Address Opcode Address + 1 Opcode Address + 2 Operand Address Operand Address + 1	1 1 1 0 0	Opcode Operand Address (High Byte) Operand Address (Low Byte) Register Data (High Byte) Register Data (Low Byte)
5-6	LDY	6	1 2 3 4 5 6	Opcode Address Opcode Address + 1 Opcode Address + 2 Opcode Address + 3 Operand Address Operand Address + 1	1 1 1 1 1 1	Opcode (Page Select Byte) ($18) Opcode (Second Byte) ($FE) Operand Address (High Byte) Operand Address (Low Byte) Operand Data (High Byte) Operand Data (Low Byte)
5-7	STY	6	1 2 3 4 5 6	Opcode Address Opcode Address + 1 Opcode Address + 2 Opcode Address + 3 Operand Address Operand Address + 1	1 1 1 1 0 0	Opcode (Page Select Byte) ($18) Opcode (Second Byte) ($FF) Operand Address (High Byte) Operand Address (Low Byte) Register Data (High Byte) Register Data (Low Byte)
5-8	ASL, ASR, CLR, COM, DEC, INC, LSL, LSR, NEG, ROL, ROR	6	1 2 3 4 5 6	Opcode Address Opcode Address + 1 Opcode Address + 2 Operand Address $FFFF Operand Address	1 1 1 1 1 0	Opcode Operand Address (High Byte) Operand Address (Low Byte) Original Operand Data Irrelevant Data Result Operand Data
5-9	TST	6	1 2 3 4 5 6	Opcode Address Opcode Address + 1 Opcode Address + 2 Operand Address $FFFF $FFFF	1 1 1 1 1 1	Opcode ($7D) Operand Address (High Byte) Operand Address (Low Byte) Original Operand Data Irrelevant Data Irrelevant Data
5-10	ADDD, CPX, SUBD	6	1 2 3 4 5 6	Opcode Address Opcode Address + 1 Opcode Address + 2 Operand Address Operand Address + 1 $FFFF	1 1 1 1 1 1	Opcode Operand Address (High Byte) Operand Address (Low Byte) Operand Data (High Byte) Operand Data (Low Byte) Irrelevant Data

*The reference number is given to provide a cross-reference to Table 10-1.

Table 10-7 Cycle-by-Cycle Operation — Indexed Y Mode (Sheet 1 of 2)

Reference Number*	Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
7-1	JMP	4	1	Opcode Address	1	Opcode (Page Select Byte) ($18)
			2	Opcode Address + 1	1	Opcode (Second Byte) ($6E)
			3	Opcode Address + 2	1	Index Offset
			4	$FFFF	1	Irrelevant Data
7-2	ADCA, ADCB, ADDA, ADDB, ANDA ANDB, BITA, BITB, CMPA, CMPB, EORA, EORB, LDAA, LDAB, ORAA, ORAB, SBCA, SBCB, SUBA, SUBB,	5	1	Opcode Address	1	Opcode (Page Select Byte) ($18)
			2	Opcode Address + 1	1	Opcode (Second Byte)
			3	Opcode Address + 2	1	Index Offset
			4	$FFFF	1	Irrelevant Data
			5	(IY) + Offset	1	Operand Data
7-3	ASL, ASR, CLR, COM, DEC, INC, LSL, LSR, NEG, ROL, ROR	7	1	Opcode Address	1	Opcode (Page Select Byte)
			2	Opcode Address + 1	1	Opcode (Second Byte)
			3	Opcode Address + 2	1	Index Offset
			4	$FFFF	1	Irrelevant Data
			5	(IY) + Offset	1	Original Operand Data
			6	$FFFF	1	Irrelevant Data
			7	(IY) + Offset	0	Result Operand Data
7-4	TST	7	1	Opcode Address	1	Opcode (Page Select Byte) ($18)
			2	Opcode Address + 1	1	Opcode (Second Byte) ($6D)
			3	Opcode Address + 2	1	Index Offset
			4	$FFFF	1	Irrelevant Data
			5	(IY) + Offset	1	Original Operand Data
			6	$FFFF	1	Irrelevant Data
			7	$FFFF	1	Irrelevant Data
7-5	STAA, STAB	5	1	Opcode Address	1	Opcode (Page Select Byte) ($18)
			2	Opcode Address + 1	1	Opcode (Second Byte)
			3	Opcode Address + 2	1	Index Offset
			4	$FFFF	1	Irrelevant Data
			5	(IY) + Offset	0	Accumulator Data
7-6	LDD, LDS, LDX, LDY	6	1	Opcode Address	1	Opcode (Page Select Byte)
			2	Opcode Address + 1	1	Opcode (Second Byte)
			3	Opcode Address + 2	1	Index Offset
			4	$FFFF	1	Irrelevant Data
			5	(IY) + Offset	1	Operand Data (High Byte)
			6	(IY) + Offset + 1	1	Operand Data (Low Byte)
7-7	STD, STS, STX, STY	6	1	Opcode Address	1	Opcode (Page Select Byte)
			2	Opcode Address + 1	1	Opcode (Second Byte)
			3	Opcode Address + 2	1	Index Offset
			4	$FFFF	1	Irrelevant Data
			5	(IY) + Offset	0	Register Data (High Byte)
			6	(IY) + Offset + 1	0	Register Data (Low Byte)
7-8	ADDD, CPD, CPX, CPY, SUBD	7	1	Opcode Address	1	Opcode (Page Select Byte)
			2	Opcode Address + 1	1	Opcode (Second Byte)
			3	Opcode Address + 2	1	Index Offset
			4	$FFFF	1	Irrelevant Data
			5	(IY) + Offset	1	Operand Data (High Byte)
			6	(IY) + Offset + 1	1	Operand Data (Low Byte)
			7	$FFFF	1	Irrelevant Data

* The reference number is given to provide a cross-reference to Table 10-1.

Table 10-7 Cycle-by-Cycle Operation — Indexed Y Mode (Sheet 2 of 2)

Reference Number*	Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
7-9	JSR	7	1 2 3 4 5 6 7	Opcode Address Opcode Address + 1 Opcode Address + 2 $FFFF (IY) + Offset Stack Pointer Stack Pointer – 1	1 1 1 1 1 0 0	Opcode (Page Select Byte) ($18) Opcode (Second Byte) ($AD) Index Offset Irrelevant Data First Opcode in Subroutine Return Address (Low Byte) Return Address (High Byte)
7-10	BCLR, BSET	8	1 2 3 4 5 6 7 8	Opcode Address Opcode Address + 1 Opcode Address + 2 $FFFF (IY) + Offset Opcode Address + 3 $FFFF (IY) + Offset	1 1 1 1 1 1 1 0	Opcode (Page Select Byte) ($18) Opcode (Second Byte) Index Offset Irrelevant Data Original Operand Data Mask Byte Irrelevant Data Result Operand Data
7-11	BRCLR, BRSET	8	1 2 3 4 5 6 7 8	Opcode Address Opcode Address + 1 Opcode Address + 2 $FFFF (IY) + Offset Opcode Address + 3 Opcode Address + 4 $FFFF	1 1 1 1 1 1 1 1	Opcode (Page Select Byte) ($18) Opcode (Second Byte) Index Offset Irrelevant Data Original Operand Data Mask Byte Branch Offset Irrelevant Data

* The reference number is given to provide a cross-reference to Table 10-1.

Table 10-8 Cycle-by-Cycle Operation — Relative Mode

Reference Number*	Address Mode and Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
8-1	BCC, BCS, BEQ, BGE, BGT, BHI, BHS, BLE, BLO, BLS, BLT, BMI, BNE, BPL, BRA, BRN, BVC, BVS,	3	1 2 3	Opcode Address Opcode Address + 1 $FFFF	1 1 1	Opcode Branch Offset Irrelevant Data
8-2	BSR	6	1 2 3 4 5 6	Opcode Address Opcode Address + 1 $FFFF Subroutine Address Stack Pointer Stack Pointer – 1	1 1 1 1 0 0	Opcode ($8D) Branch Offset Irrelevant Data Opcode of Next Instruction Return Address (Low Byte) Return Address (High Byte)

*The reference number is given to provide a cross-reference to Table 10-1.

Table A-4a Control Timing (MC68L11A8)

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

Characteristic	Symbol	1.0 MHz		2.0 MHz		Unit
		Min	Max	Min	Max	
Frequency of Operation	f_o	dc	1.0	dc	2.0	MHz
E-Clock Period	t_{cyc}	1000	—	500	—	ns
Crystal Frequency	f_{XTAL}	—	4.0	—	8.0	MHz
External Oscillator Frequency	4 f_o	dc	4.0	dc	8.0	MHz
Processor Control Setup Time t_{PCSU} = 1/4 t_{cyc} + 50 ns	t_{PCSU}	325	—	200	—	ns
Reset Input Pulse Width (Note 1)	PW_{RSTL}					t_{cyc}
(To Guarantee External Reset Vector)		8	—	8	—	
(Minimum Input Time; Can Be Preempted by Internal Reset)		1	—	1	—	
Mode Programming Setup Time	t_{MPS}	2	—	2	—	t_{cyc}
Mode Programming Hold Time	t_{MPH}	10	—	10	—	ns
Interrupt Pulse Width, $\overline{IRQ}$ Edge-Sensitive Mode $PW_{IRQ} = t_{cyc}$ + 20 ns	PW_{IRQ}	1020	—	520	—	ns
Wait Recovery Start-up Time	t_{WRS}	—	4	—	4	t_{cyc}
Timer Pulse Width Input Capture, Pulse Accumulator Input $PW_{TIM} = t_{cyc}$ + 20 ns	PW_{TIM}	1020	—	520	—	ns

NOTES:

1. $\overline{RESET}$ is recognized during the first clock cycle it is held low. Internal circuitry then drives the pin low for four clock cycles, releases the pin, and samples the pin level two cycles later to determine the source of the interrupt. Refer to **9 RESETS, INTERRUPTS, AND LOW POWER MODES** for further detail.
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.

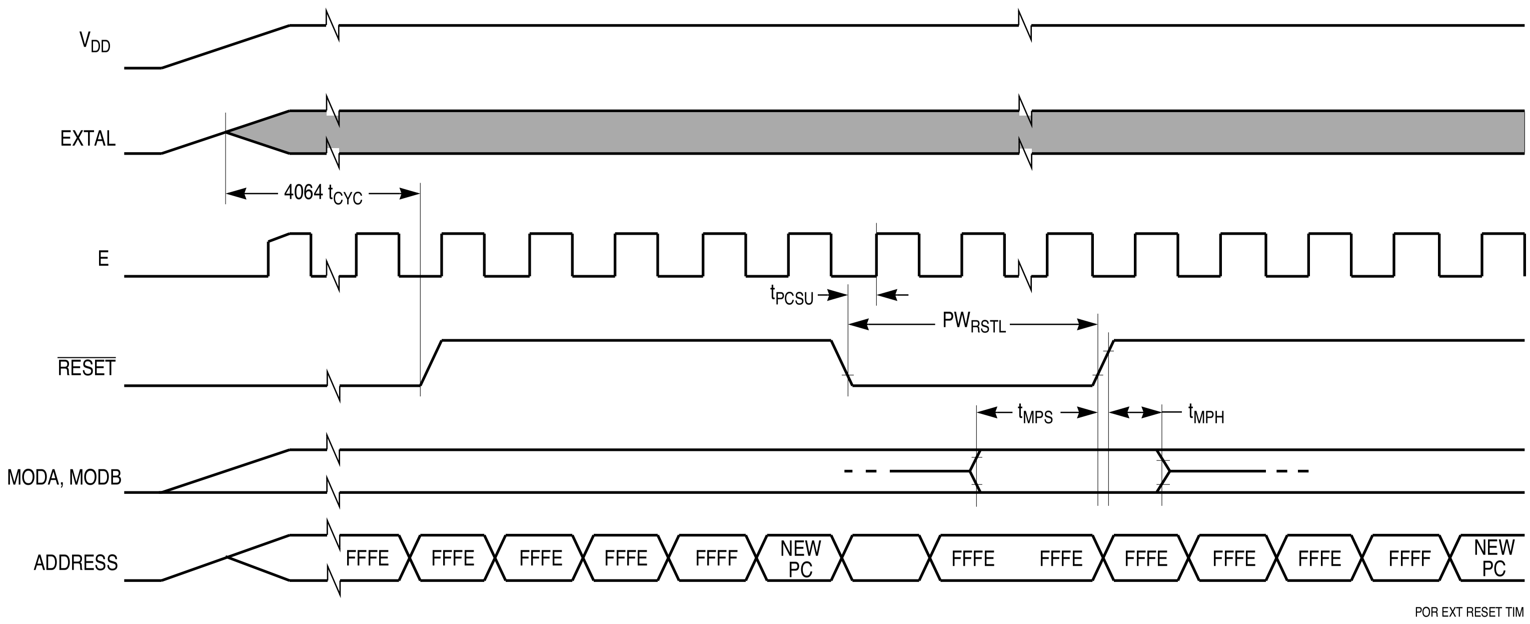
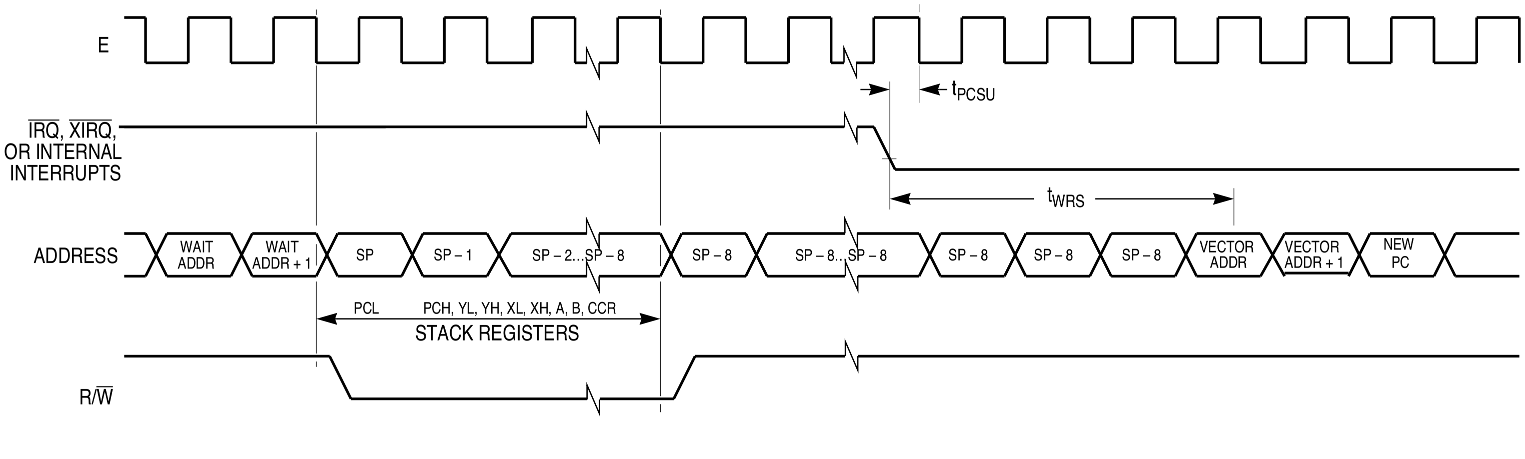
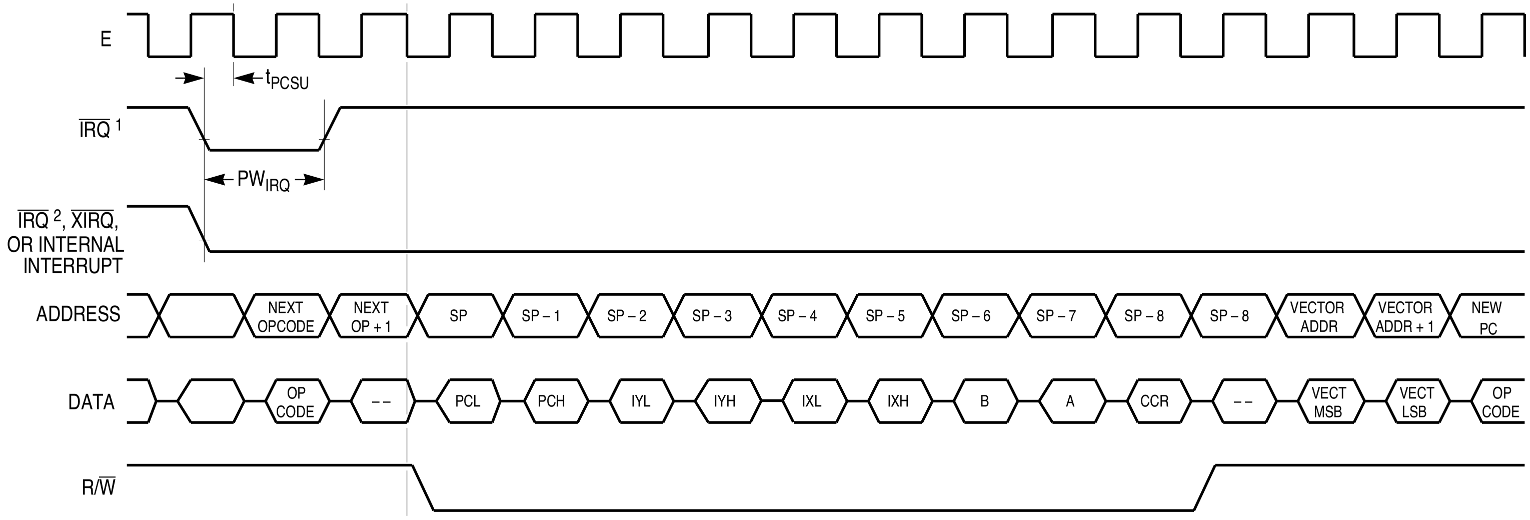


Figure A-3 POR and External Reset Timing Diagram



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

Figure A-5 WAIT Recovery Timing Diagram



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

INTERRUPT TIM

Figure A-6 Interrupt Timing Diagram

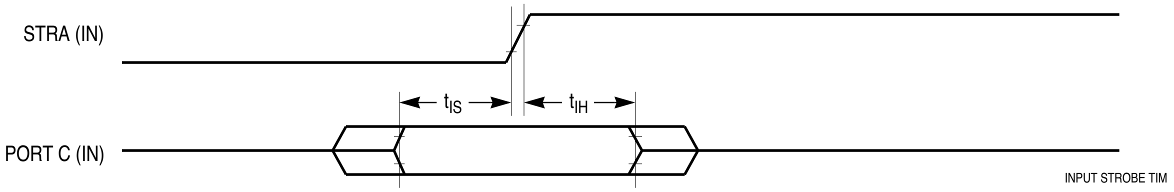
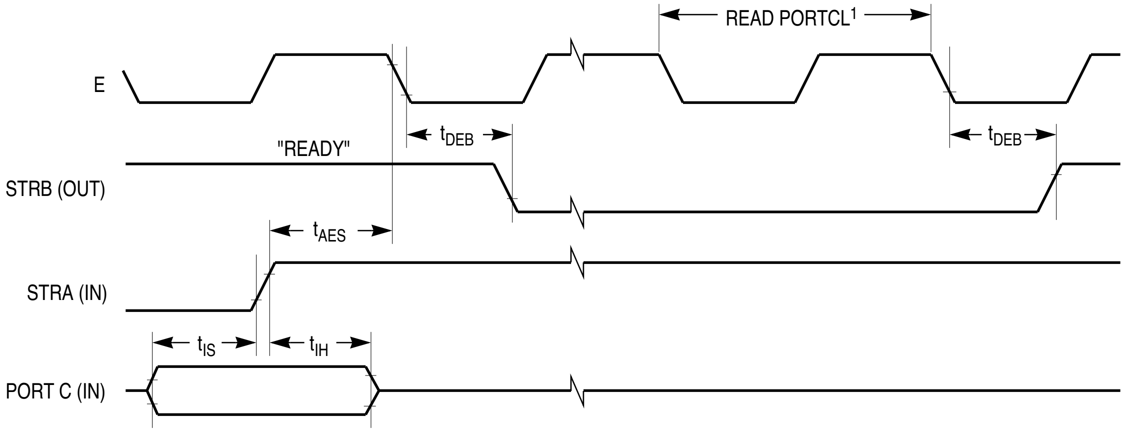


Figure A-10 Simple Input Strobe Timing Diagram

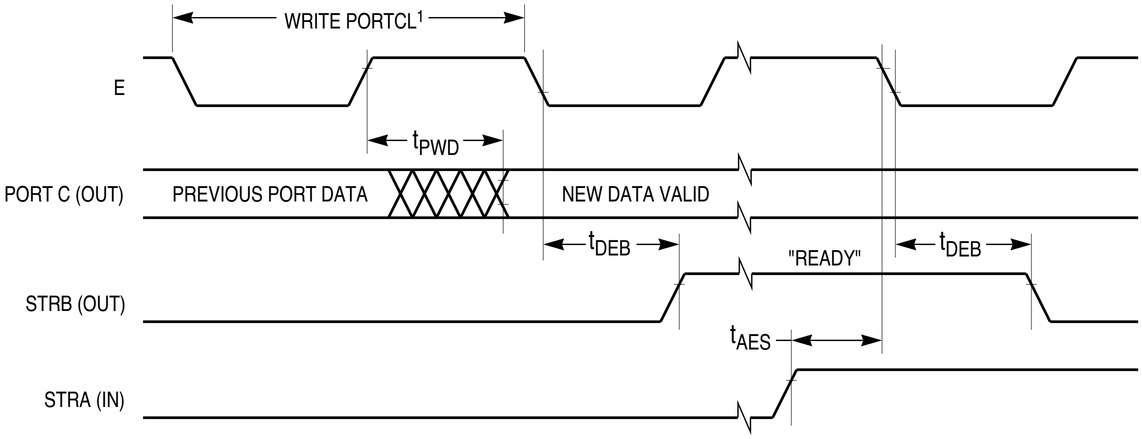


NOTES:

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

PORTC INPUT HNDSHK TIM

Figure A-11 Port C Input Handshake Timing Diagram



NOTES:

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

PORTC OUTPUT HNDSHK TIM

Figure A-12 Port C Output Handshake Timing Diagram