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	4MHz
Connectivity	SCI, SPI
Peripherals	POR, WDT
Number of I/O	30
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	4.75V ~ 5.25V
Data Converters	A/D 8x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	68-LCC (J-Lead)
Supplier Device Package	68-PLCC (24.21x24.21)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc11f1cfn4

TABLE OF CONTENTS
(Continued)

Paragraph **Title** **Page**

TABLE OF CONTENTS
(Continued)

LIST OF ILLUSTRATIONS

2.11.2 Port B

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

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

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

2.11.3 Port C

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

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

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

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

2.11.4 Port D

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

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

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

When an interrupt is recognized, the current instruction finishes normally, the return address (the current value in the program counter) is pushed onto the stack, all of the CPU registers are pushed onto the stack, and execution continues at the address specified by the vector for the interrupt. At the end of the interrupt service routine, an RTI instruction is executed. The RTI instruction causes the saved registers to be pulled off the stack in reverse order. Program execution resumes at the return address.

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

3.1.5 Program Counter (PC)

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

Table 3-1 Reset Vector Comparison

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

3.1.6 Condition Code Register (CCR)

This 8-bit register contains five condition code indicators (C, V, Z, N, and H), two interrupt masking bits, (I and X) and a stop disable bit (S). In the M68HC11 CPU, condition codes are automatically updated by most instructions. For example, load accumulator A (LDAA) and store accumulator A (STAA) instructions automatically set or clear the N, Z, and V condition code flags. Pushes, pulls, add B to X (ABX), add B to Y (ABY), and transfer/exchange instructions do not affect the condition codes. Refer to **Table 3-2**, which shows what condition codes are affected by a particular instruction.

3.1.6.1 Carry/Borrow (C)

The C bit is set if the arithmetic logic unit (ALU) performs a carry or borrow during an arithmetic operation. The C bit also acts as an error flag for multiply and divide operations. Shift and rotate instructions operate with and through the carry bit to facilitate multiple-word shift operations.

3.1.6.2 Overflow (V)

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

5.1.3 Computer Operating Properly (COP) Reset

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

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

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

Table 5-1 COP Timer Rate Selection

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

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

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

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

5.1.4 Clock Monitor Reset

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

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

Table 5-4 Interrupt and Reset Vector Assignments

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

For some interrupt sources, such as the SCI interrupts, the flags are automatically cleared during the normal course of responding to the interrupt requests. For example, the RDRF flag in the SCI system is cleared by the automatic clearing mechanism consisting of a read of the SCI status register while RDRF is set, followed by a read of the SCI data register. The normal response to an RDRF interrupt request would be to read the SCI status register to check for receive errors, then to read the received data from the SCI data register. These two steps satisfy the automatic clearing mechanism without requiring any special instructions.

5.4.1 Interrupt Recognition and Register Stacking

An interrupt can be recognized at any time after it is enabled by its local mask, if any, and by the global mask bit in the CCR. Once an interrupt source is recognized, the CPU responds at the completion of the instruction being executed. Interrupt latency varies according to the number of cycles required to complete the current instruction. When the CPU begins to service an interrupt, the contents of the CPU registers are pushed onto the stack in the order shown in **Table 5-5**. After the CCR value is stacked, the I bit and the X bit (if $\overline{\text{XIRQ}}$ is pending) are set to inhibit further interrupts. The interrupt vector for the highest priority pending source is fetched, and execution continues

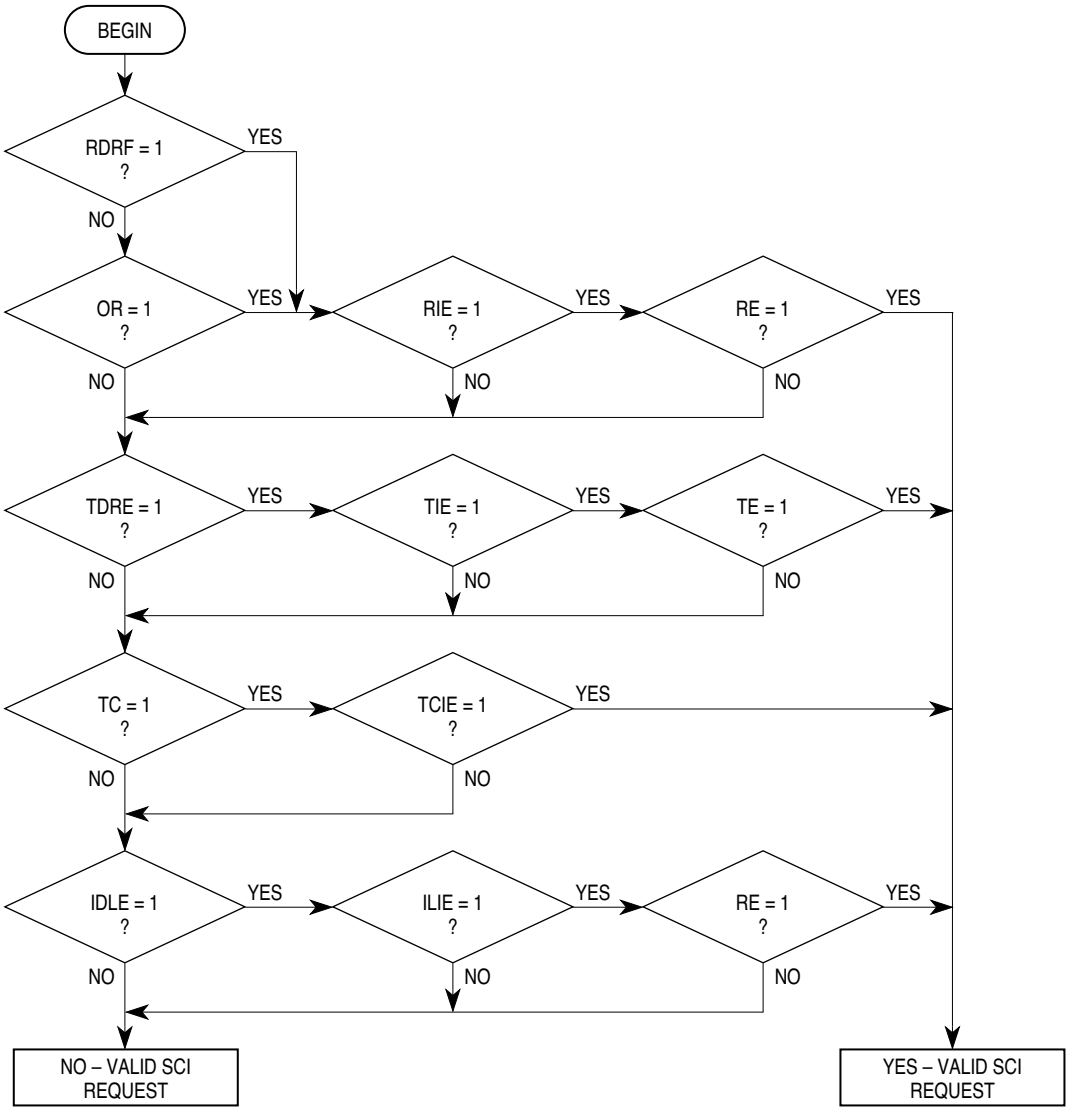


Figure 5-5 Interrupt Source Resolution Within SCI

5.5 Low Power Operation

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

SECTION 6 PARALLEL INPUT/OUTPUT

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

Table 6-1 I/O Port Configuration

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

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

6.1 Port A

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

PORTA — Port A Data **$1000**

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

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

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

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

NOTE

To enable PA3 as fourth input capture, set the I4/O5 bit in the PACTL register. Otherwise, PA3 is configured as a fifth output compare out of reset, with bit I4/O5 being cleared. If the DDA3 bit is set (configuring PA3 as an output), and IC4 is enabled, writes to PA3 cause edges on the pin to result in input captures. Writing to TI4/O5 has no effect when the TI4/O5 register is acting as IC4. PA7 drives the pulse accumulator input but also can be configured for general-purpose I/O, or output compare. Note that even when PA7 is configured as an output, the pin still drives the pulse accumulator input.

6.2 Port B

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

PORTB — Port B Data **$1004**

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

6.3 Port C

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

The CWOM control bit in the OPT2 register disables port C's P-channel output drivers. Because the N-channel driver is not affected by CWOM, setting CWOM causes port C to become an open-drain-type output port suitable for wired-OR operation. In wired-OR mode, (PORTC bits are at logic level zero), pins are actively driven low by the N-channel driver. When a port C bit is at logic level one, the associated pin is in a high-

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

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

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

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

NOTE

When the SPI system is in slave mode, DDD5 has no meaning nor effect. When the SPI system is in master mode, DDD5 determines whether bit 5 of PORTD is an error detect input (DDD5 = 0) or a general-purpose output (DDD5 = 1). If the SPI system is enabled and expects any of bits [4:2] to be an input, that bit will be an input regardless of the state of the associated DDR bit. If any of bits [4:2] are expected to be outputs that bit will be an output **only** if the associated DDR bit is set.

6.5 Port E

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

PORTE — Port E Data **$100A**

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

6.6 Port F

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

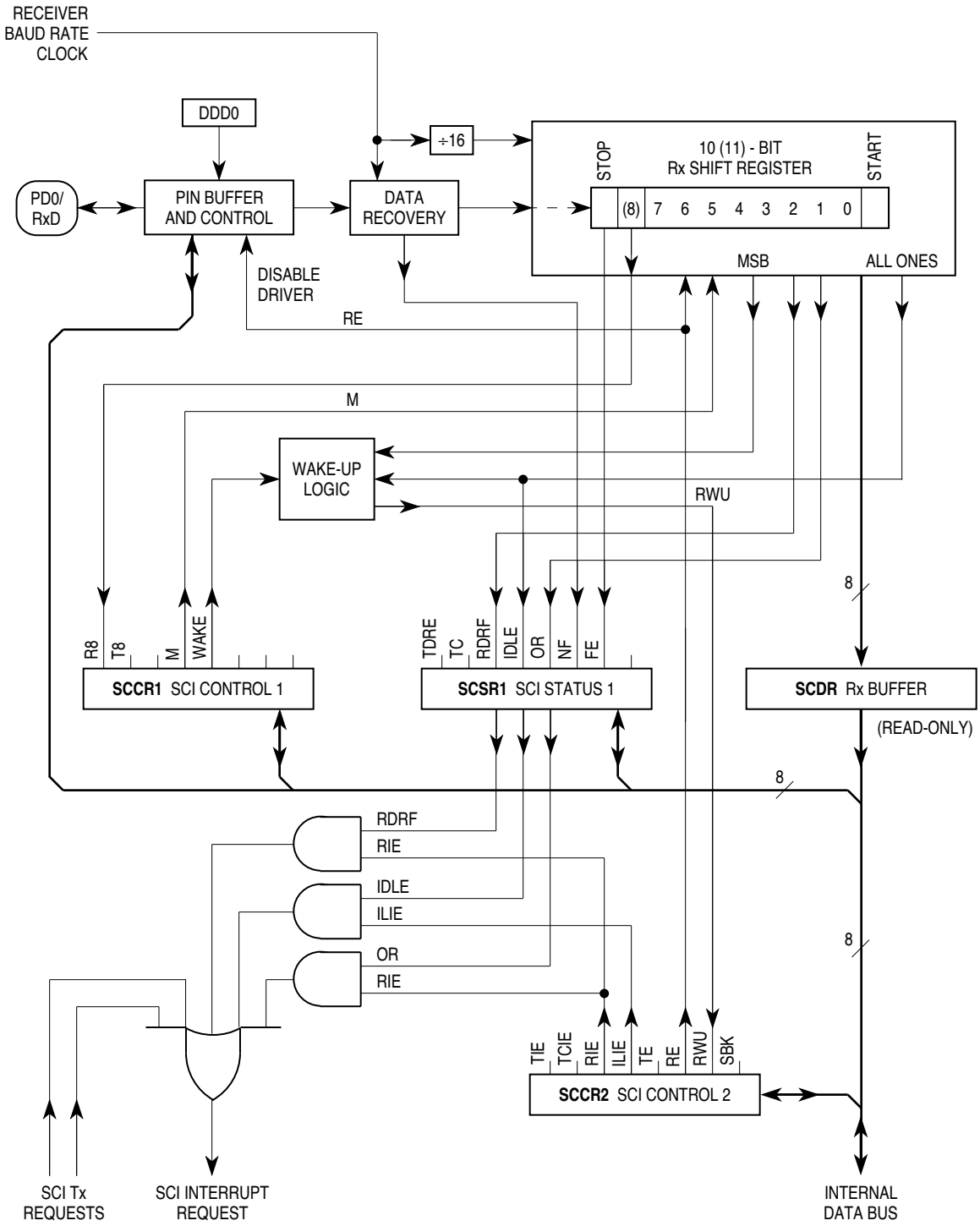


Figure 7-2 SCI Receiver Block Diagram

SECTION 8 SERIAL PERIPHERAL INTERFACE

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

8.1 Functional Description

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

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

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

9.6.3 Pulse Accumulator Status and Interrupt Bits

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

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

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

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

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

PAOVI and PAOVF — Pulse Accumulator Interrupt Enable and Overflow Flag

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

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

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

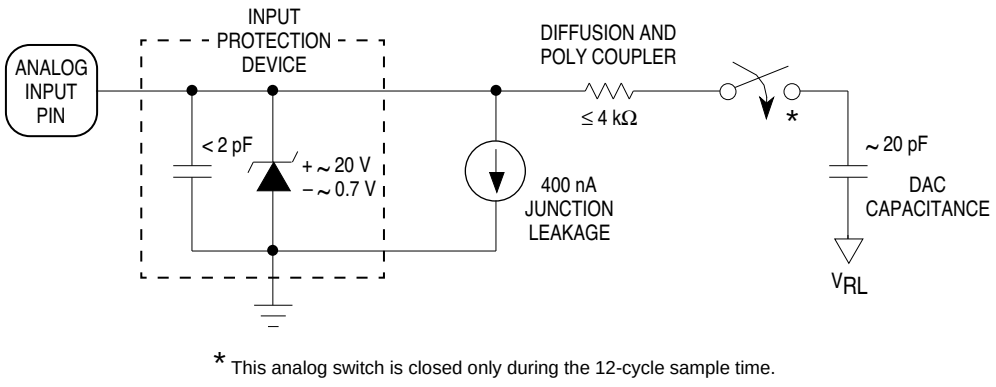


Figure 10-2 Electrical Model of an A/D Input Pin (Sample Mode)

10.1.2 Analog Converter

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

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

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

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

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

10.1.3 Digital Control

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

10.1.4 Result Registers

Four 8-bit registers (ADR1–ADR4) store conversion results. Each of these registers can be accessed by the processor in the CPU. The conversion complete flag (CCF)

NOTE

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

CD–CA — Channel Selects D–A

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

Table 10-2 A/D Converter Channel Selection

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

*Used for factory testing

10.9 A/D Converter Result Registers

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

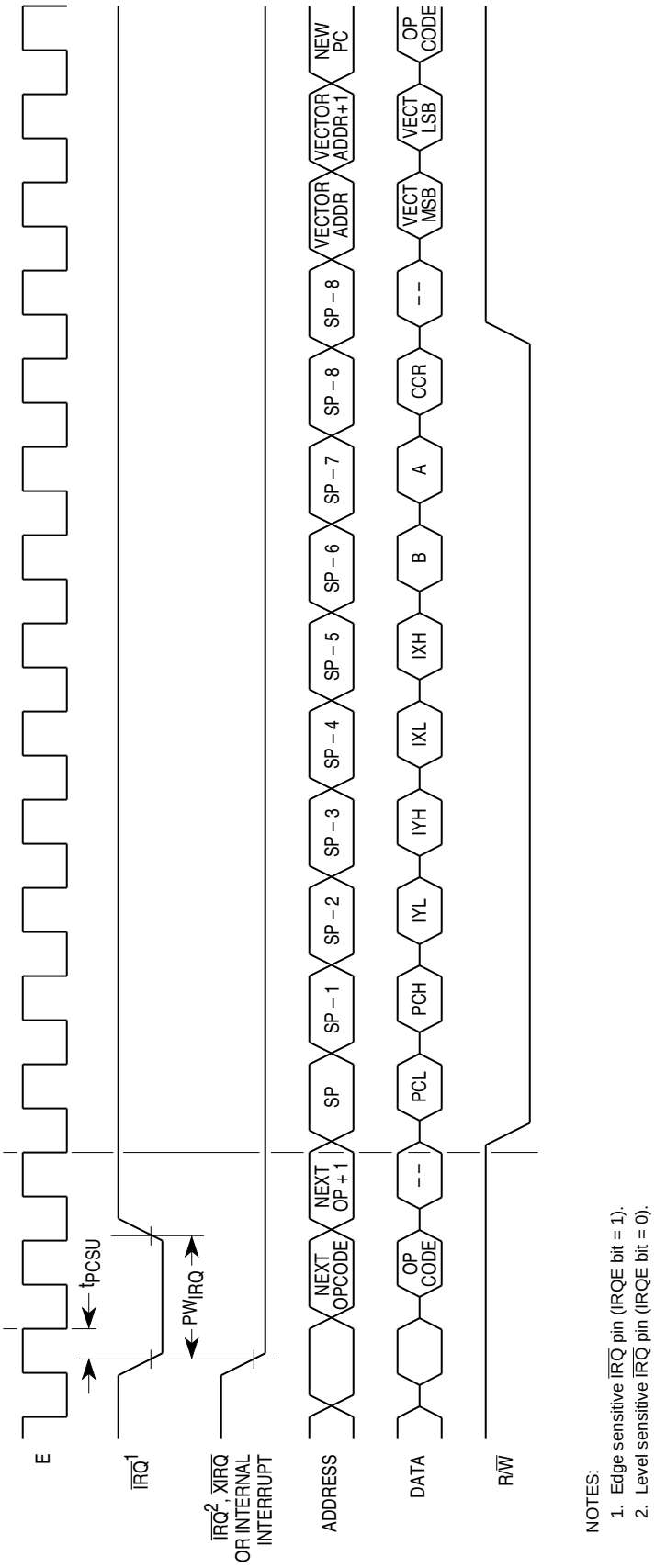


Figure A-6 Interrupt Timing Diagram

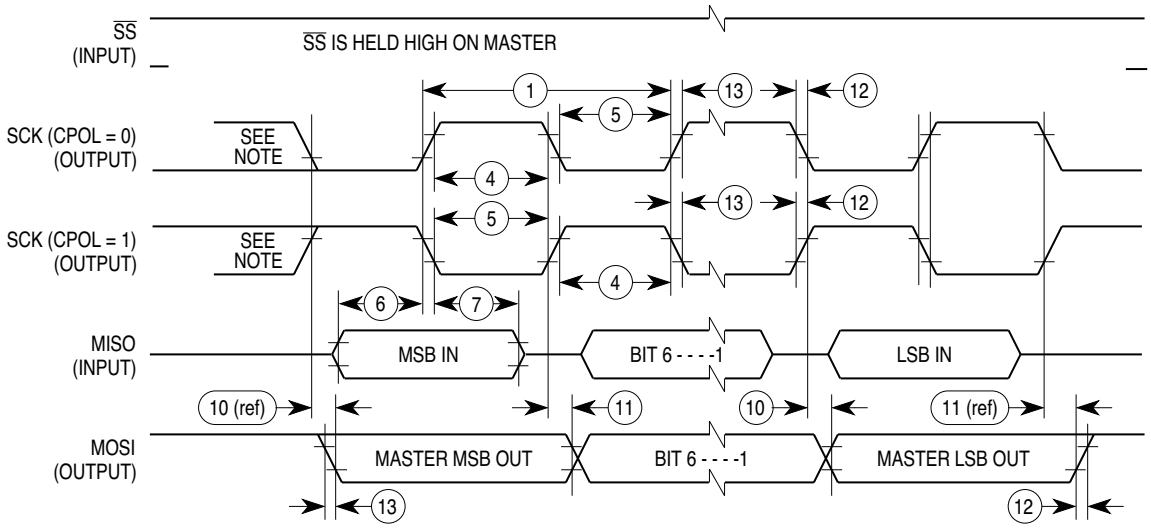
Table A-7 Expansion Bus Timing

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

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

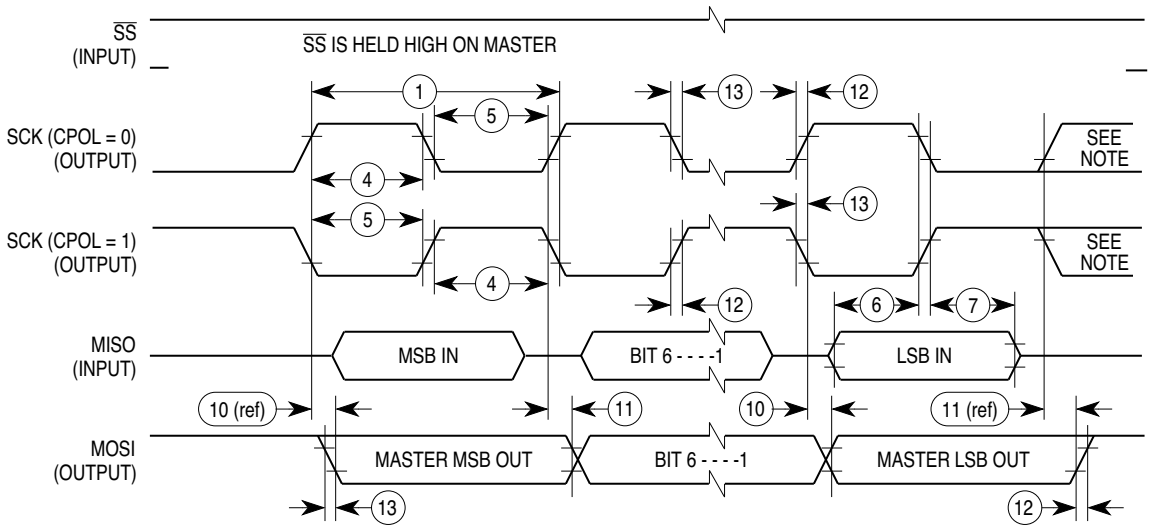
NOTES:

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



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

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



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

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

APPENDIX CDEVELOPMENT SUPPORT

C.1 MC68HC11F1 Development Tools

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

Table C-1 MC68HC11F1 Development Tools

Device	Evaluation Systems	Modular Development Systems
MC68HC11F1	M68HC11F1EVS	MMDS11*

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

C.2 MC68HC11EVS — Evaluation System

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

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

C.3 M68MMDS11 — Modular Development System for M68HC11 Devices

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

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