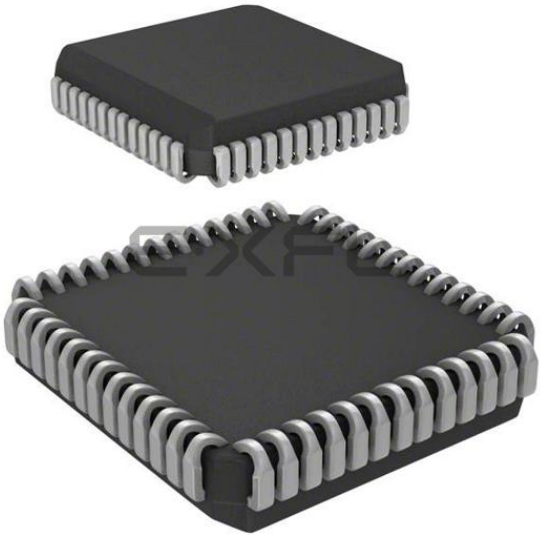


Welcome to E-XFL.COM

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.

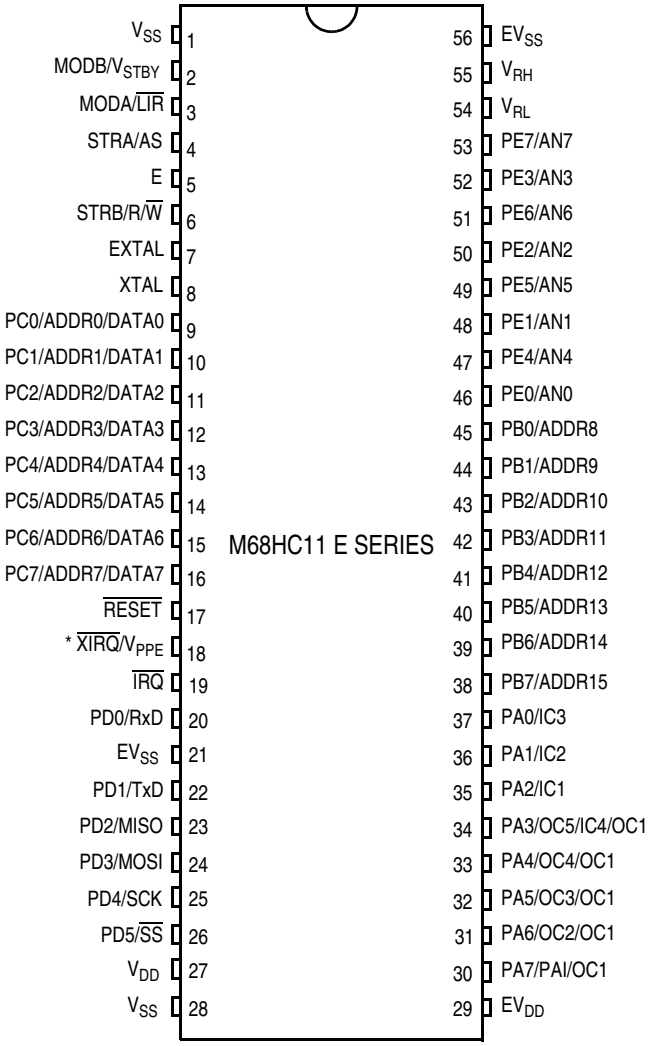
Applications of "Embedded - Microcontrollers"

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	512 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x8b
Oscillator Type	Internal
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/mc68hc11e1cfn2r2

Table of Contents

Section 1. General Description

Section 2. Operating Modes and On-Chip Memory



* V_{PPE} applies only to devices with EPROM/OTPROM.

Figure 1-5. Pin Assignments for 56-Pin SDIP

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$101E	Timer Input Capture 4/Output Compare 5 Register High (TI4/O5) See page 148.	Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Reset:	1	1	1	1	1	1	1	1
$101F	Timer Input Capture 4/Output Compare 5 Register Low (TI4/O5) See page 148.	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$1020	Timer Control Register 1 (TCTL1) See page 153.	Read: Write:	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5
		Reset:	0	0	0	0	0	0	0	0
$1021	Timer Control Register 2 (TCTL2) See page 146.	Read: Write:	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
		Reset:	0	0	0	0	0	0	0	0
$1022	Timer Interrupt Mask 1 Register (TMSK1) See page 154.	Read: Write:	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I
		Reset:	0	0	0	0	0	0	0	0
$1023	Timer Interrupt Flag 1 (TFLG1) See page 154.	Read: Write:	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F
		Reset:	0	0	0	0	0	0	0	0
$1024	Timer Interrupt Mask 2 Register (TMSK2) See page 155.	Read: Write:	TOI	RTII	PAOVI	PAII			PR1	PR0
		Reset:	0	0	0	0	0	0	0	0
$1025	Timer Interrupt Flag 2 (TFLG2) See page 158.	Read: Write:	TOF	RTIF	PAOVF	PAIF				
		Reset:	0	0	0	0	0	0	0	0
$1026	Pulse Accumulator Control Register (PACTL) See page 159.	Read: Write:	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0
		Reset:	0	0	0	0	0	0	0	0
$1027	Pulse Accumulator Count Register (PACNT) See page 162.	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	Indeterminate after reset							
$1028	Serial Peripheral Control Register (SPCR) See page 138.	Read: Write:	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0
		Reset:	0	0	0	0	0	1	U	U
$1029	Serial Peripheral Status Register (SPSR) See page 139.	Read: Write:	SPIF	WCOL		MODF				
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented R = Reserved U = Unaffected
I = Indeterminate after reset

Figure 2-7. Register and Control Bit Assignments (Sheet 4 of 6)

Address: $103F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	EE3	EE2	EE1	EE0	NOSEC	NOCOP		EEON
Resets:								
Single chip:	1	1	1	1	U	U	1	1
Bootstrap:	1	1	1	1	U	U(L)	1	1
Expanded:	U	U	U	U	1	U	1	U
Test:	U	U	U	U	1	U(L)	1	0

= Unimplemented

U indicates a previously programmed bit. U(L) indicates that the bit resets to the logic level held in the latch prior to reset, but the function of COP is controlled by the DISR bit in TEST1 register.

Figure 2-11. MC68HC811E2 System Configuration Register (CONFIG)

EE[3:0] — EEPROM Mapping Bits

EE[3:0] apply only to MC68HC811E2 and allow the 2048 bytes of EEPROM to be remapped to any 4-Kbyte boundary. See **Table 2-3**.

Table 2-3. EEPROM Mapping

EE[3:0]	EEPROM Location
0 0 0 0	$0800–$0FFF
0 0 0 1	$1800–$1FFF
0 0 1 0	$2800–$2FFF
0 0 1 1	$3800–$3FFF
0 1 0 0	$4800–$4FFF
0 1 0 1	$5800–$5FFF
0 1 1 0	$6800–$6FFF
0 1 1 1	$7800–$7FFF
1 0 0 0	$8800–$8FFF
1 0 0 1	$9800–$9FFF
1 0 1 0	$A800–$AFFF
1 0 1 1	$B800–$BFFF
1 1 0 0	$C800–$CFFF
1 1 0 1	$D800–$DFFF
1 1 1 0	$E800–$EFFF
1 1 1 1	$F800–$FFFF

NOSEC — Security Disable Bit

NOSEC is invalid unless the security mask option is specified before the MCU is manufactured. If the security mask option is omitted NOSEC always reads 1. The enhanced security feature is available in the MC68S711E9 MCU. The enhancement to the standard security feature protects the EPROM as well as RAM and EEPROM.

0 = Security enabled
1 = Security disabled

NOCOP — COP System Disable Bit

Refer to **Section 5. Resets and Interrupts**.

1 = COP disabled
0 = COP enabled

ROMON — ROM/EPROM/OTPROM Enable Bit

When this bit is 0, the ROM or EPROM is disabled and that memory space becomes externally addressed. In single-chip mode, ROMON is forced to 1 to enable ROM/EPROM regardless of the state of the ROMON bit.

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

EEON — EEPROM Enable Bit

When this bit is 0, the EEPROM is disabled and that memory space becomes externally addressed.

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

2.3.3.2 RAM and I/O Mapping Register

The internal registers used to control the operation of the MCU can be relocated on 4-Kbyte boundaries within the memory space with the use of the RAM and I/O mapping register (INIT). This 8-bit special-purpose register can change the default locations of the RAM and control registers within the MCU memory map. It can be written only once within the first 64 E-clock cycles after a reset in normal modes, and then it becomes a read-only register.

Address: $103D

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

Figure 2-12. RAM and I/O Mapping Register (INIT)

RAM[3:0] — RAM Map Position Bits

These four bits, which specify the upper hexadecimal digit of the RAM address, control position of RAM in the memory map. RAM can be positioned at the beginning of any 4-Kbyte page in the memory map. It is initialized to address $0000 out of reset. Refer to **Table 2-4**.

4.2.6 Condition Code Register (CCR)

This 8-bit register contains:

- Five condition code indicators (C, V, Z, N, and H),
- Two interrupt masking bits ($\overline{\text{IRQ}}$ and $\overline{\text{XIRQ}}$)
- A stop disable bit (S)

In the M68HC11 CPU, condition codes are updated automatically by most instructions. For example, load accumulator A (LDAA) and store accumulator A (STAA) instructions automatically set or clear the N, Z, and V condition code flags. Pushes, pulls, add B to X (ABX), add B to Y (ABY), and transfer/exchange instructions do not affect the condition codes. Refer to **Table 4-2**, which shows what condition codes are affected by a particular instruction.

4.2.6.1 Carry/Borrow (C)

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

4.2.6.2 Overflow (V)

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

4.2.6.3 Zero (Z)

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

4.2.6.4 Negative (N)

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

4.2.6.5 Interrupt Mask (I)

The interrupt request (IRQ) mask (I bit) is a global mask that disables all maskable interrupt sources. While the I bit is set, interrupts can become pending, but the

Table 4-2. Instruction Set (Sheet 7 of 7)

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
SWI	Software Interrupt	See Figure 3–2	INH	3F	—	14	—	—	—	1	—	—	—	—
TAB	Transfer A to B	A ⇒ B	INH	16	—	2	—	—	—	—	Δ	Δ	0	—
TAP	Transfer A to CC Register	A ⇒ CCR	INH	06	—	2	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ
TBA	Transfer B to A	B ⇒ A	INH	17	—	2	—	—	—	—	Δ	Δ	0	—
TEST	TEST (Only in Test Modes)	Address Bus Counts	INH	00	—	*	—	—	—	—	—	—	—	—
TPA	Transfer CC Register to A	CCR ⇒ A	INH	07	—	2	—	—	—	—	—	—	—	—
TST (opr)	Test for Zero or Minus	M – 0	EXT IND,X IND,Y	7D 6D 18 6D	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	0	0
TSTA	Test A for Zero or Minus	A – 0	A INH	4D	—	2	—	—	—	—	Δ	Δ	0	0
TSTB	Test B for Zero or Minus	B – 0	B INH	5D	—	2	—	—	—	—	Δ	Δ	0	0
TSX	Transfer Stack Pointer to X	SP + 1 ⇒ IX	INH	30	—	3	—	—	—	—	—	—	—	—
TSY	Transfer Stack Pointer to Y	SP + 1 ⇒ IY	INH	18 30	—	4	—	—	—	—	—	—	—	—
TXS	Transfer X to Stack Pointer	IX – 1 ⇒ SP	INH	35	—	3	—	—	—	—	—	—	—	—
TYS	Transfer Y to Stack Pointer	IY – 1 ⇒ SP	INH	18 35	—	4	—	—	—	—	—	—	—	—
WAI	Wait for Interrupt	Stack Regs & WAIT	INH	3E	—	**	—	—	—	—	—	—	—	—
XGDX	Exchange D with X	IX ⇒ D, D ⇒ IX	INH	8F	—	3	—	—	—	—	—	—	—	—
XGDY	Exchange D with Y	IY ⇒ D, D ⇒ IY	INH	18 8F	—	4	—	—	—	—	—	—	—	—

Cycle

* Infinity or until reset occurs

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

Operands

`dd` = 8-bit direct address ($0000–$00FF) (high byte assumed to be $00)
`ff` = 8-bit positive offset $00 (0) to $FF (255) (is added to index)
`hh` = High-order byte of 16-bit extended address
`ii` = One byte of immediate data
`jj` = High-order byte of 16-bit immediate data
`kk` = Low-order byte of 16-bit immediate data
`ll` = Low-order byte of 16-bit extended address
`mm` = 8-bit mask (set bits to be affected)
`rr` = Signed relative offset $80 (–128) to $7F (+127)
(offset relative to address following machine code offset byte))

Operators

( ) Contents of register shown inside parentheses
⇐ Is transferred to
⇑ Is pulled from stack
⇓ Is pushed onto stack
• Boolean AND
\+ Arithmetic addition symbol except where used as inclusive-OR symbol in Boolean formula
⊕ Exclusive-OR
* Multiply
: Concatenation
– Arithmetic subtraction symbol or negation symbol (two's complement)

Condition Codes

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

7.4 Receive Operation

During receive operations, the transmit sequence is reversed. The serial shift register receives data and transfers it to a parallel receive data register (SCDR) as a complete word. This double buffered operation allows a character to be shifted in serially while another character is already in the SCDR. An advanced data recovery scheme distinguishes valid data from noise in the serial data stream. The data input is selectively sampled to detect receive data, and a majority voting circuit determines the value and integrity of each bit. See **Figure 7-2**.

7.5 Wakeup Feature

The wakeup feature reduces SCI service overhead in multiple receiver systems. Software for each receiver evaluates the first character of each message. The receiver is placed in wakeup mode by writing a 1 to the RWU bit in the SCCR2 register. While RWU is 1, all of the receiver-related status flags (RDRF, IDLE, OR, NF, and FE) are inhibited (cannot become set). Although RWU can be cleared by a software write to SCCR2, to do so would be unusual. Normally, RWU is set by software and is cleared automatically with hardware. Whenever a new message begins, logic alerts the sleeping receivers to wake up and evaluate the initial character of the new message.

Two methods of wakeup are available:

- Idle-line wakeup
- Address-mark wakeup

During idle-line wakeup, a sleeping receiver awakens as soon as the RxD line becomes idle. In the address-mark wakeup, logic 1 in the most significant bit (MSB) of a character wakes up all sleeping receivers.

7.5.1 Idle-Line Wakeup

To use the receiver wakeup method, establish a software addressing scheme to allow the transmitting devices to direct a message to individual receivers or to groups of receivers. This addressing scheme can take any form as long as all transmitting and receiving devices are programmed to understand the same scheme. Because the addressing information is usually the first frame(s) in a message, receivers that are not part of the current task do not become burdened with the entire set of addressing frames. All receivers are awake (RWU = 0) when each message begins. As soon as a receiver determines that the message is not intended for it, software sets the RWU bit (RWU = 1), which inhibits further flag setting until the RxD line goes idle at the end of the message. As soon as an idle line is detected by receiver logic, hardware automatically clears the RWU bit so that the first frame of the next message can be received. This type of receiver wakeup requires a minimum of one idle-line frame time between messages and no idle time between frames in a message.

7.7.3 Serial Communications Control Register 2

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

Address: $102D

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

Figure 7-5. Serial Communications Control Register 2 (SCCR2)

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

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

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

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

TE — Transmitter Enable Bit

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

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

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

SBK — Send Break

At least one character time of break is queued and sent each time SBK is written to 1. As long as the SBK bit is set, break characters are queued and sent. More than one break may be sent if the transmitter is idle at the time the SBK bit is toggled on and off, as the baud rate clock edge could occur between writing the 1 and writing the 0 to SBK.
- 0 = Break generator off
- 1 = Break codes generated

Section 8. Serial Peripheral Interface (SPI)

8.1 Introduction

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

- Frequency synthesizers
- Liquid crystal display (LCD) drivers
- Analog-to-digital (A/D) converter subsystems
- Other microprocessors

The SPI is also capable of inter-processor communication in a multiple master system. The SPI system can be configured as either a master or a slave device. When configured as a master, data transfer rates can be as high as one-half the E-clock rate (1.5 Mbits per second for a 3-MHz bus frequency). When configured as a slave, data transfers can be as fast as the E-clock rate (3 Mbits per second for a 3-MHz bus frequency).

8.2 Functional Description

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

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

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

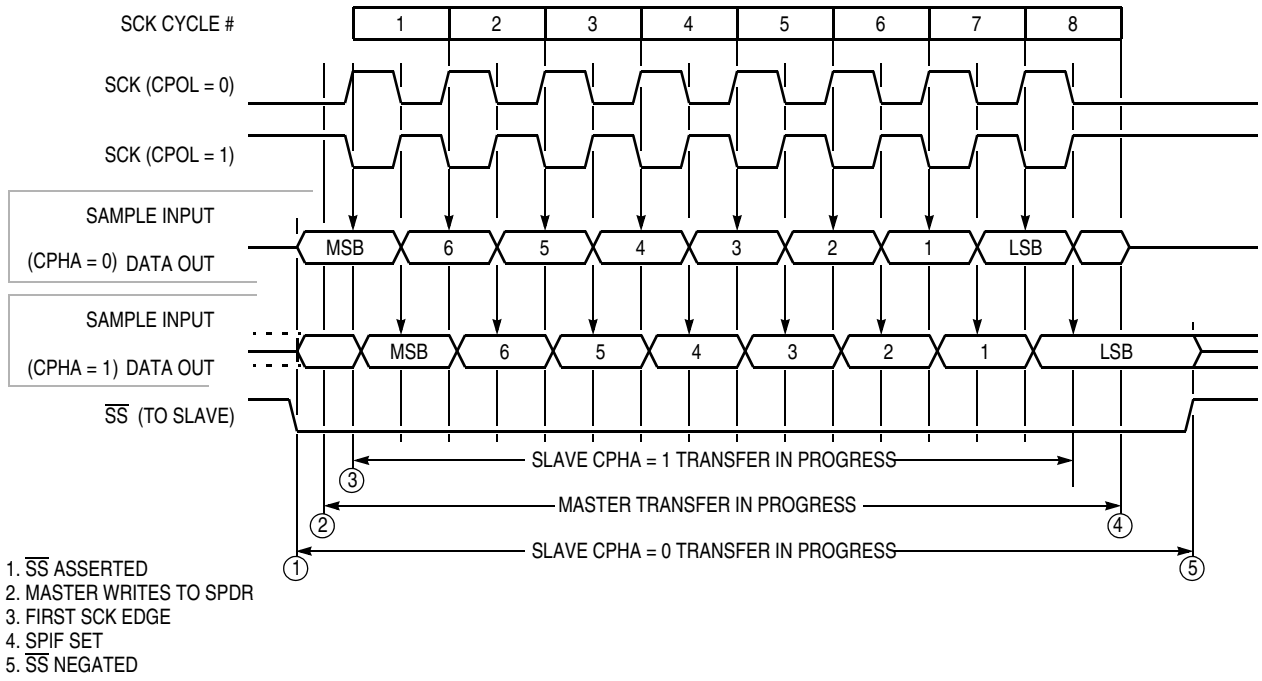


Figure 8-2. SPI Transfer Format

8.4 Clock Phase and Polarity Controls

Software can select one of four combinations of serial clock phase and polarity using two bits in the SPI control register (SPCR). The clock polarity is specified by the CPOL control bit, which selects an active high or active low clock, and has no significant effect on the transfer format. The clock phase (CPHA) control bit selects one of two different transfer formats. The clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transfers to allow a master device to communicate with peripheral slaves having different requirements.

When CPHA equals 0, the $\overline{SS}$ line must be negated and reasserted between each successive serial byte. Also, if the slave writes data to the SPI data register (SPDR) while $\overline{SS}$ is low, a write collision error results.

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

8.5 SPI Signals

This subsection contains descriptions of the four SPI signals:

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

register is compared to the free-running counter value during each E-clock cycle. If a match is found, the particular output compare flag is set in timer interrupt flag register 1 (TFLG1). If that particular interrupt is enabled in the timer interrupt mask register 1 (TMSK1), an interrupt is generated. In addition to an interrupt, a specified action can be initiated at one or more timer output pins. For OC[5:2], the pin action is controlled by pairs of bits (OMx and OLx) in the TCTL1 register. The output action is taken on each successful compare, regardless of whether or not the OCxF flag in the TFLG1 register was previously cleared.

OC1 is different from the other output compares in that a successful OC1 compare can affect any or all five of the OC pins. The OC1 output action taken when a match is found is controlled by two 8-bit registers with three bits unimplemented: the output compare 1 mask register, OC1M, and the output compare 1 data register, OC1D. OC1M specifies which port A outputs are to be used, and OC1D specifies what data is placed on these port pins.

9.4.1 Timer Output Compare Registers

All output compare registers are 16-bit read-write. Each is initialized to $FFFF at reset. If an output compare register is not used for an output compare function, it can be used as a storage location. A write to the high-order byte of an output compare register pair inhibits the output compare function for one bus cycle. This inhibition prevents inappropriate subsequent comparisons. Coherency requires a complete 16-bit read or write. However, if coherency is not needed, byte accesses can be used.

For output compare functions, write a comparison value to output compare registers TOC1–TOC4 and TI4/O5. When TCNT value matches the comparison value, specified pin actions occur.

Register name: Timer Output Compare 1 Register (High) Address: $1016

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	1	1	1	1	1	1	1	1

Register name: Timer Output Compare 1 Register (Low) Address: $1017

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

Figure 9-8. Timer Output Compare 1 Register Pair (TOC1)

Register name: Timer Output Compare 2 Register (High) Address: $1018

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	1	1	1	1	1	1	1	1

Register name: Timer Output Compare 2 Register (Low) Address: $1019

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

Figure 9-9. Timer Output Compare 2 Register Pair (TOC2)

Register name: Timer Output Compare 3 Register (High) Address: $101A

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	1	1	1	1	1	1	1	1

Register name: Timer Output Compare 3 Register (Low) Address: $101B

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

Figure 9-10. Timer Output Compare 3 Register Pair (TOC3)

Register name: Timer Output Compare 4 Register (High) Address: $101C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	1	1	1	1	1	1	1	1

Register name: Timer Output Compare 4 Register (Low) Address: $101D

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

Figure 9-11. Timer Output Compare 4 Register Pair (TOC4)

The clock source for the RTI function is a free-running clock that cannot be stopped or interrupted except by reset. This clock causes the time between successive RTI timeouts to be a constant that is independent of the software latencies associated with flag clearing and service. For this reason, an RTI period starts from the previous timeout, not from when RTIF is cleared.

Every timeout causes the RTIF bit in TFLG2 to be set, and if RTII is set, an interrupt request is generated. After reset, one entire RTI period elapses before the RTIF is set for the first time. Refer to the **9.4.9 Timer Interrupt Mask 2 Register**, **9.5.2 Timer Interrupt Flag Register 2**, and **9.5.3 Pulse Accumulator Control Register**.

9.5.1 Timer Interrupt Mask Register 2

This register contains the real-time interrupt enable bits.

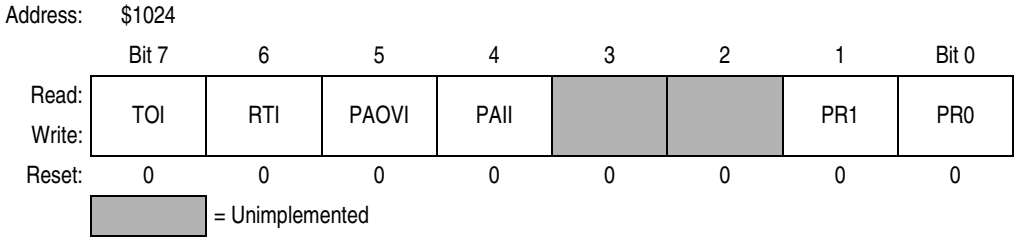


Figure 9-21. Timer Interrupt Mask 2 Register (TMSK2)

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

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

PAOVI — Pulse Accumulator Overflow Interrupt Enable Bit
Refer to **9.7 Pulse Accumulator**.

PAII — Pulse Accumulator Input Edge Bit
Refer to **9.7 Pulse Accumulator**.

Bits [3:2] — Unimplemented
Always read 0

PR[1:0] — Timer Prescaler Select Bits
Refer to **Table 9-4**.

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

10.11 Peripheral Port Timing

Characteristic[1] [2]	Symbol	1.0 MHz		2.0 MHz		3.0 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of operation E-clock frequency	f_o	dc	1.0	dc	2.0	dc	3.0	MHz
E-clock period	t_{CYC}	1000	—	500	—	333	—	ns
Peripheral data setup time MCU read of ports A, C, D, and E	t_{PDSU}	100	—	100	—	100	—	ns
Peripheral data hold time MCU read of ports A, C, D, and E	t_{PDH}	50	—	50	—	50	—	ns
Delay time, peripheral data write t_{PWD} = 1/4 t_{CYC}+ 100 ns MCU writes to port A MCU writes to ports B, C, and D	t_{PWD}	 — —	 200 350	 — —	 200 225	 — —	 200 183	ns
Port C input data setup time	t_{IS}	60	—	60	—	60	—	ns
Port C input data hold time	t_{IH}	100	—	100	—	100	—	ns
Delay time, E fall to STRB t_{DEB} = 1/4 t_{CYC}+ 100 ns	t_{DEB}	—	350	—	225	—	183	ns
Setup time, STRA asserted to E fall[3]	t_{AES}	0	—	0	—	0	—	ns
Delay time, STRA asserted to port C data output valid	t_{PCD}	—	100	—	100	—	100	ns
Hold time, STRA negated to port C data	t_{PCH}	10	—	10	—	10	—	ns
3-state hold time	t_{PCZ}	—	150	—	150	—	150	ns

1. V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, all timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted
2. Ports C and D timing is valid for active drive. (CWOM and DWOM bits are not set in PIOC and SPCR registers, respectively.)
3. If this setup time is met, STRB acknowledges in the next cycle. If it is not met, the response may be delayed one more cycle.

Electrical Characteristics

Description	CONFIG	Temperature	Frequency	MC Order Number
52-pin plastic leaded chip carrier (PLCC) (Continued)				
OTPROM	$0F	–40°C to +85°C	2 MHz	MC68HC711E9CFN2
			3 MHz	MC68HC711E9CFN3
		–40°C to +105°C	2 MHz	MC68HC711E9VFN2
		–40°C to +125°C	2 MHz	MC68HC711E9MFN2
OTPROM, enhanced security feature	$0F	–40°C to +85°C	2 MHz	MC68S711E9CFN2
20 Kbytes OTPROM	$0F	0°C to +70°C	3 MHz	MC68HC711E20FN3
		–40°C to +85°C	2 MHz	MC68HC711E20CFN2
			3 MHz	MC68HC711E20CFN3
		–40°C to +105°C	2 MHz	MC68HC711E20VFN2
		–40°C to +125°C	2 MHz	MC68HC711E20MFN2
No ROM, 2 Kbytes EEPROM	$FF	0°C to +70°C	2 MHz	MC68HC811E2FN2
		–40°C to +85°C	2 MHz	MC68HC811E2CFN2
		–40°C to +105°C	2 MHz	MC68HC811E2VFN2
		–40°C to +125°C	2 MHz	MC68HC811E2MFN2
64-pin quad flat pack (QFP)				
BUFFALO ROM	$0F	–40°C to +85°C	2 MHz	MC68HC11E9BCFU2
			3 MHz	MC68HC11E9BCFU3
No ROM	$0D	–40°C to +85°C	2 MHz	MC68HC11E1CFU2
			3 MHz	MC68HC11E1CFU3
		–40°C to +105°C	2 MHz	MC68HC11E1VFU2
No ROM, no EEPROM	$0C	–40°C to +85°C	2 MHz	MC68HC11E0CFU2
		–40°C to +105°C	2 MHz	MC68HC11E0VFU2
20 Kbytes OTPROM	$0F	0°C to +70°°C	3 MHz	MC68HC711E20FU3
		–40°C to +85°C	2 MHz	MC68HC711E20CFU2
			3 MHz	MC68HC711E20CFU3
		–40°C to +105°C	2 MHz	MC68HC711E20VFU2
		–40°C to +125°C	2 MHz	MC68HC711E20MFU2
52-pin thin quad flat pack (TQFP)				
BUFFALO ROM	$0F	–40°C to +85°C	2 MHz	MC68HC11E9BCPB2
			3 MHz	MC68HC11E9BCPB3

series resistor will prevent direct conflict between the internal TxD driver and the external driver connected to PD1 through the series resistor.

Other

The bootloader firmware sets the DWOM control bit, which configures all port D pins for wire-OR operation. During the bootloading process, all port D pins except the PD1/TxD pin are configured as high-impedance inputs. Any port D pin that normally is used as an output should have a pullup resistor so it does not float during the bootloading process.

Driving Boot Mode from Another M68HC11

A second M68HC11 system can easily act as the host to drive bootstrap loading of an M68HC11 MCU. This method is used to examine and program non-volatile memories in target M68HC11s in Motorola EVMs. The following hardware and software example will demonstrate this and other bootstrap mode features.

The schematic in **Figure 6** shows the circuitry for a simple EPROM duplicator for the MC68HC711E9. The circuitry is built in the wire-wrap area of an M68HC11EVBU evaluation board to simplify construction. The schematic shows only the important portions of the EVBU circuitry to avoid confusion. To see the complete EVBU schematic, refer to the *M68HC11EVBU Universal Evaluation Board User's Manual*, Motorola document order number M68HC11EVBU/D.

The default configuration of the EVBU must be changed to make the appropriate connections to the circuitry in the wire-wrap area and to configure the master MCU for bootstrap mode. A fabricated jumper must be installed at J6 to connect the XTAL output of the master MCU to the wire-wrap connector P5, which has been wired to the EXTAL input of the target MCU. Cut traces that short across J8 and J9 must be cut on the solder side of the printed circuit board to disconnect the normal SCI connections to the RS232 level translator (U4) of the EVBU. The J8 and J9 connections can be restored easily at a later time by installing fabricated jumpers on the component side of the board. A fabricated

```
1598 GOSUB 8020
1599 WEND
1600 XMT = 0: RCV = 0           'POINTERS TO XMIT AND RECEIVE BYTES
1610 A$ = CHR$(CODE%(XMT))
1620 GOSUB 6500                 'SEND FIRST BYTE
1625 FOR I = 1 TO CODESIZE% - 1         'ZERO BASED ARRAY 0 -> CODESIZE-1
1630 A$ = CHR$(CODE%(I))        'SEND SECOND BYTE TO GET ONE IN QUEUE
1635 GOSUB 6500                 'SEND IT
1640 GOSUB 8000                 'GET BYTE FOR VERIFICATION
1650 RCV = I - 1
1660 LOCATE 10,1:PRINT "Verifying byte #"; I; "    "
1664 IF CHR$(CODE%(RCV)) = B$ THEN 1670
1665 K=CODE%(RCV):GOSUB 8500
1666 LOCATE 1,1:PRINT "Byte #"; I; "    ", " - Sent "; HX$;
1668 K=ASC(B$):GOSUB 8500
1669 PRINT "  Received "; HX$;
1670 NEXT I
1680 GOSUB 8000                 'GET BYTE FOR VERIFICATION
1690 RCV = CODESIZE% - 1
1700 LOCATE 10,1:PRINT "Verifying byte #"; CODESIZE%; "    "
1710 IF CHR$(CODE%(RCV)) = B$ THEN 1720
1713 K=CODE(RCV):GOSUB 8500
1714 LOCATE 1,1:PRINT "Byte #"; CODESIZE%; "    ", " - Sent "; HX$;
1715 K=ASC(B$):GOSUB 8500
1716 PRINT "  Received "; HX$;
1720 LOCATE 8, 1: PRINT : PRINT "Done!!!!"
4900 CLOSE
4910 INPUT "Press [RETURN] to quit...", Q$
5000 END
5900 '*************************************************************************
5910 '*         SUBROUTINE TO READ IN ONE BYTE FROM A DISK FILE
5930 '*                 RETURNS BYTE IN A$
5940 '*************************************************************************
6000 FLAG = 0
6010 IF EOF(1) THEN FLAG = 1: RETURN
6020 A$ = INPUT$(1, #1)
6030 RETURN
6490 '*************************************************************************
6492 '*         SUBROUTINE TO SEND THE STRING IN A$ OUT TO THE DEVICE
6494 '*                 OPENED AS FILE #2.
6496 '*************************************************************************
6500 PRINT #2, A$;
6510 RETURN
6590 '*************************************************************************
6594 '*         SUBROUTINE THAT CONVERTS THE HEX DIGIT IN A$ TO AN INTEGER
6596 '*************************************************************************
7000 X = INSTR(H$, A$)
7010 IF X = 0 THEN FLAG = 1
7020 X = X - 1
7030 RETURN
```