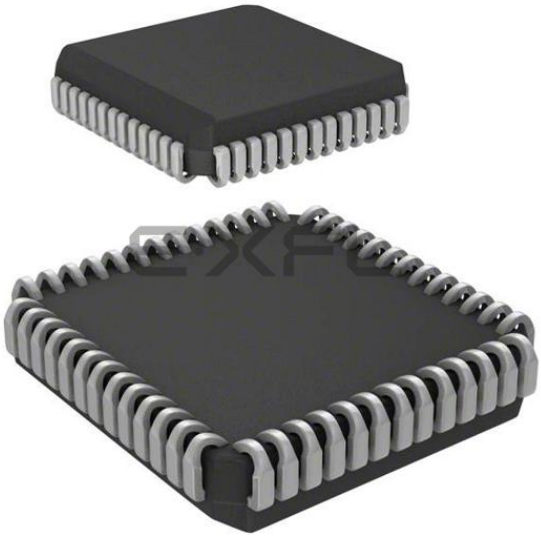


Welcome to E-XFL.COM

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	2MHz
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/mc68hcp11e1cfne2

PA6–PA4 serve as either general-purpose outputs, timer input captures, or timer output compare 2–4. In addition, PA6–PA4 can be controlled by OC1.

PA3 can be a general-purpose I/O pin or a timer IC/OC pin. Timer functions associated with this pin include OC1 and IC4/OC5. IC4/OC5 is software selectable as either a fourth input capture or a fifth output compare. PA3 can also be configured to allow OC1 edges to trigger IC4 captures.

PA2–PA0 serve as general-purpose inputs or as IC1–IC3.

PORTA can be read at any time. Reads of pins configured as inputs return the logic level present on the pin. Pins configured as outputs return the logic level present at the pin driver input. If written, PORTA stores the data in an internal latch, bits 7 and 3. It drives the pins only if they are configured as outputs. Writes to PORTA do not change the pin state when pins are configured for timer input captures or output compares. Refer to Chapter 6 Parallel Input/Output (I/O) Ports.

1.4.13 Port B

During single-chip operating modes, all port B pins are general-purpose output pins. During MCU reads of this port, the level sensed at the input side of the port B output drivers is read. Port B can also be used in simple strobed output mode. In this mode, an output pulse appears at the STRB signal each time data is written to port B.

In expanded multiplexed operating modes, all of the port B pins act as high order address output signals. During each MCU cycle, bits 15–8 of the address bus are output on the PB7–PB0 pins. The PORTB register is treated as an external address in expanded modes.

1.4.14 Port C

While in single-chip operating modes, all port C pins are general-purpose I/O pins. Port C inputs can be latched into an alternate PORTCL register by providing an input transition to the STRA signal. Port C can also be used in full handshake modes of parallel I/O where the STRA input and STRB output act as handshake control lines.

When in expanded multiplexed modes, all port C pins are configured as multiplexed address/data signals. During the address portion of each MCU cycle, bits 7–0 of the address are output on the PC7–PC0 pins. During the data portion of each MCU cycle (E high), PC7–PC0 are bidirectional data signals, DATA7–DATA0. The direction of data at the port C pins is indicated by the R/$\overline{W}$ signal.

The CWOM control bit in the PIOC register disables the port C P-channel output driver. CWOM simultaneously affects all eight bits of port C. 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:

- When a port C bit is at logic level 0, it is driven low by the N-channel driver.
- When a port C bit is at logic level 1, the associated pin has high-impedance, as neither the N-channel nor the P-channel devices are active.

It is customary to have an external pullup 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 mode. Refer to Chapter 6 Parallel Input/Output (I/O) Ports for additional information about port C functions.

IRQE — Configure $\overline{\text{IRQ}}$ for Edge-Sensitive Only Operation Bit

Refer to Chapter 5 Resets and Interrupts.

DLY — Enable Oscillator Startup Delay Bit

0 = The oscillator startup delay coming out of stop mode is bypassed and the MCU resumes processing within about four bus cycles.

1 = A delay of approximately 4000 E-clock cycles is imposed as the MCU is started up from the stop power-saving mode. This delay allows the crystal oscillator to stabilize.

CME — Clock Monitor Enable Bit

Refer to Chapter 5 Resets and Interrupts.

Bit 2 — Not implemented

Always reads 0

CR[1:0] — COP Timer Rate Select Bits

The internal E clock is divided by 2^{15} before it enters the COP watchdog system. These control bits determine a scaling factor for the watchdog timer. Refer to Chapter 5 Resets and Interrupts.

2.4 EPROM/OTPROM

Certain devices in the M68HC11 E series include on-chip EPROM/OTPROM. For instance:

- The MC68HC711E9 devices contain 12 Kbytes of on-chip EPROM (OTPROM in non-windowed package).
- The MC68HC711E20 has 20 Kbytes of EPROM (OTPROM in non-windowed package).
- The MC68HC711E32 has 32 Kbytes of EPROM (OTPROM in non-windowed package).

Standard MC68HC71E9 and MC68HC711E20 devices are shipped with the EPROM/OTPROM contents erased (all 1s). The programming operation programs zeros. Windowed devices must be erased using a suitable ultraviolet light source before reprogramming. Depending on the light source, erasing can take from 15 to 45 minutes.

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

PPROG is the combined EPROM/OTPROM and EEPROM programming register on all devices with EPROM/OTPROM except the MC68HC711E20. For the MC68HC711E20, there is a separate register for EPROM/OTPROM programming called the EPROG register.

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

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

3.10 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 3-3, which shows the A/D conversion sequence diagram.

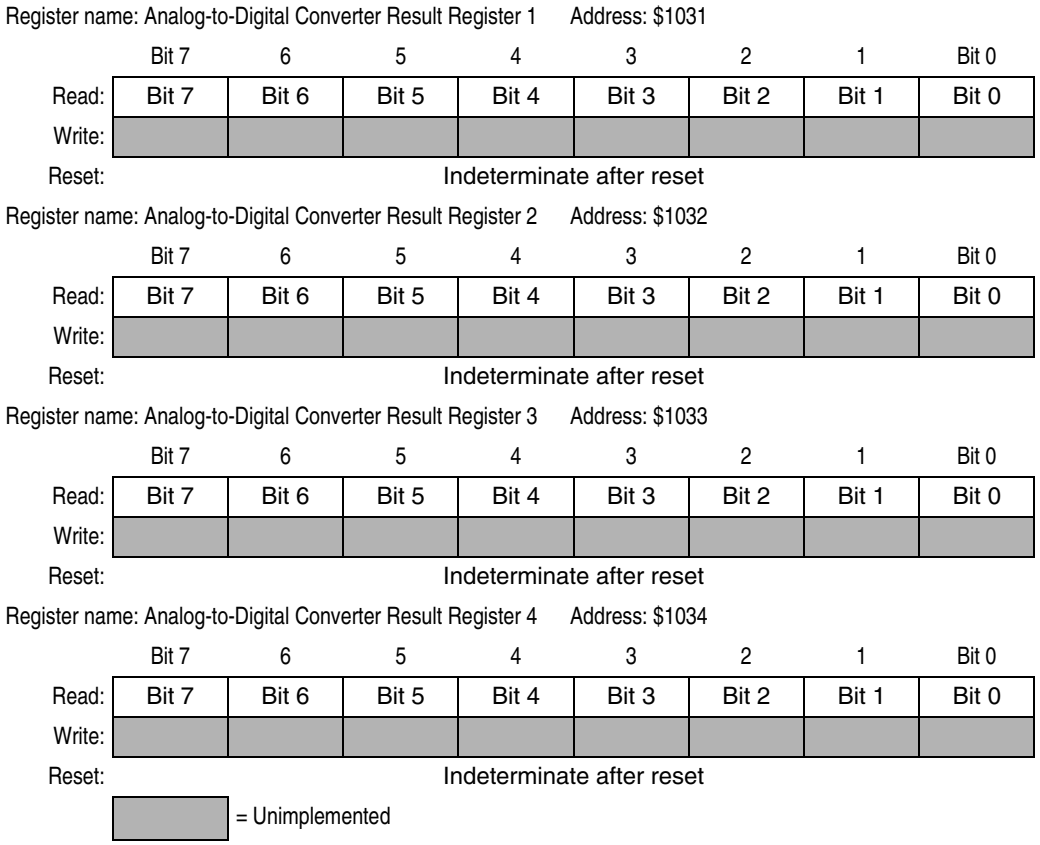


Figure 3-6. Analog-to-Digital Converter Result Registers (ADR1–ADR4)

6.3 Port B

In single-chip or bootstrap modes, port B pins are general-purpose outputs. In expanded or special test modes, port B pins are high-order address outputs.

Address: $1004

	Bit 7	6	5	4	3	2	1	Bit 0
Single-chip or bootstrap modes:								
Read: Write:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Reset:	0	0	0	0	0	0	0	0
Expanded or special test modes:								
Read: Write:	ADDR15	ADDR14	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8
Reset:	0	0	0	0	0	0	0	0

Figure 6-3. Port B Data Register (PORTB)

6.4 Port C

In single-chip and bootstrap modes, port C pins reset to high-impedance inputs. (DDRC bits are set to 0.) In expanded and special test modes, port C pins are multiplexed address/data bus and the port C register address is treated as an external memory location.

Address: $1003

	Bit 7	6	5	4	3	2	1	Bit 0
Single-chip or bootstrap modes:								
Read: Write:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Reset:	Indeterminate after reset							
Expanded or special test modes:								
Read: Write:	ADDR7 DATA7	ADDR6 DATA6	ADDR5 DATA5	ADDR4 DATA4	ADDR3 DATA3	ADDR2 DATA2	ADDR1 DATA1	ADDR0 DATA0
Reset:	Indeterminate after reset							

Figure 6-4. Port C Data Register (PORTC)

Address: $1005

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0
Reset:	Indeterminate after reset							

Figure 6-5. Port C Latched Register (PORTCL)

FE — Framing Error Flag

FE is set when a 0 is detected where a stop bit was expected. Clear the FE flag by reading SCSR with FE set and then reading SCDR.

0 = Stop bit detected
1 = Zero detected

Bit 0 — Unimplemented

Always reads 0

7.7.5 Baud Rate Register

Use this register to select different baud rates for the SCI system. The SCP[1:0] (SCP[2:0] in MC68HC(7)11E20) bits function as a prescaler for the SCR[2:0] bits. Together, these five bits provide multiple baud rate combinations for a given crystal frequency. Normally, this register is written once during initialization. The prescaler is set to its fastest rate by default out of reset and can be changed at any time. Refer to Table 7-1 for normal baud rate selections.

Address: $102B

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	TCLR	SCP2	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0
Reset:	0	0	0	0	0	U	U	U

U = Unaffected

Figure 7-7. Baud Rate Register (BAUD)

TCLR — Clear Baud Rate Counter Bit (Test)

SCP[2:0] — SCI Baud Rate Prescaler Select Bits

NOTE

SCP2 applies to MC68HC(7)11E20 only. When SCP2 = 1, SCP[1:0] must equal 0s. Any other values for SCP[1:0] are not decoded in the prescaler and the results are unpredictable. Refer to Figure 7-8 and Figure 7-9.

RCKB — SCI Baud Rate Clock Check Bit (Test)

See Table 7-1.

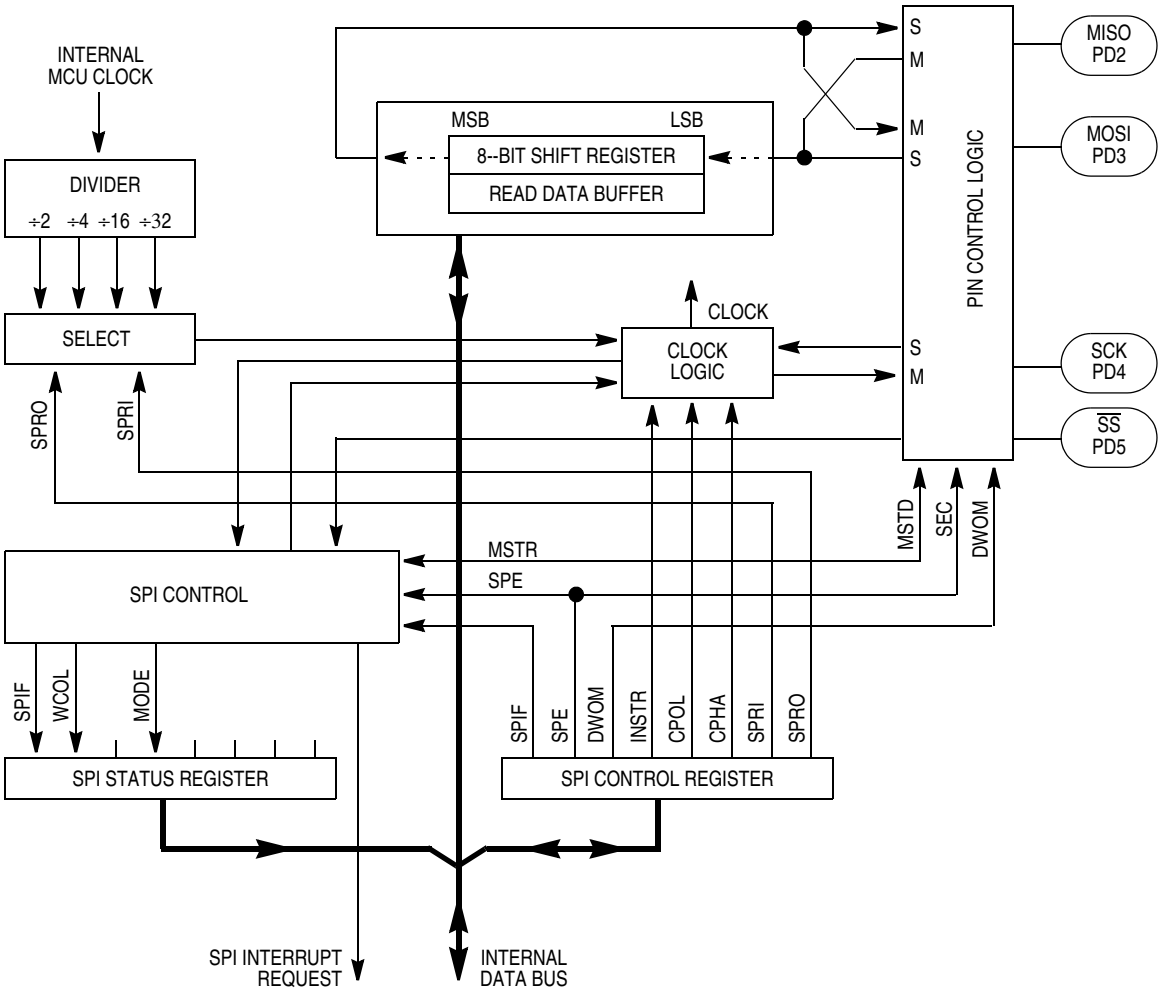


Figure 8-1. SPI Block Diagram

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.

A write collision is normally a slave error because a slave has no control over when a master initiates a transfer. A master knows when a transfer is in progress, so there is no reason for a master to generate a write-collision error, although the SPI logic can detect write collisions in both master and slave devices.

The SPI configuration determines the characteristics of a transfer in progress. For a master, a transfer begins when data is written to SPDR and ends when SPIF is set. For a slave with CPHA equal to 0, a transfer starts when $\overline{SS}$ goes low and ends when $\overline{SS}$ returns high. In this case, SPIF is set at the middle of the eighth SCK cycle when data is transferred from the shifter to the parallel data register, but the transfer is still in progress until $\overline{SS}$ goes high. For a slave with CPHA equal to 1, transfer begins when the SCK line goes to its active level, which is the edge at the beginning of the first SCK cycle. The transfer ends in a slave in which CPHA equals 1 when SPIF is set.

8.7 SPI Registers

The three SPI registers are:

- Serial peripheral control register (SPCR)
- Serial peripheral status register (SPSR)
- Serial peripheral data register (SPDR)

These registers provide control, status, and data storage functions.

8.7.1 Serial Peripheral Control Register

Address: $1028

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

U = Unaffected

Figure 8-3. Serial Peripheral Control Register (SPCR)

SPIE — Serial Peripheral Interrupt Enable Bit

Set the SPE bit to 1 to request a hardware interrupt sequence each time the SPIF or MODF status flag is set. SPI interrupts are inhibited if this bit is clear or if the I bit in the condition code register is 1.

0 = SPI system interrupts disabled
1 = SPI system interrupts enabled

SPE — Serial Peripheral System Enable Bit

When the SPE bit is set, the port D bit 2, 3, 4, and 5 pins are dedicated to the SPI function. If the SPI is in the master mode and DDRD bit 5 is set, then the port D bit 5 pin becomes a general-purpose output instead of the $\overline{SS}$ input.

0 = SPI system disabled
1 = SPI system enabled

DWOM — Port D Wired-OR Mode Bit

DWOM affects all port D pins.

0 = Normal CMOS outputs
1 = Open-drain outputs

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)

9.4.10 Timer Interrupt Flag Register 2

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

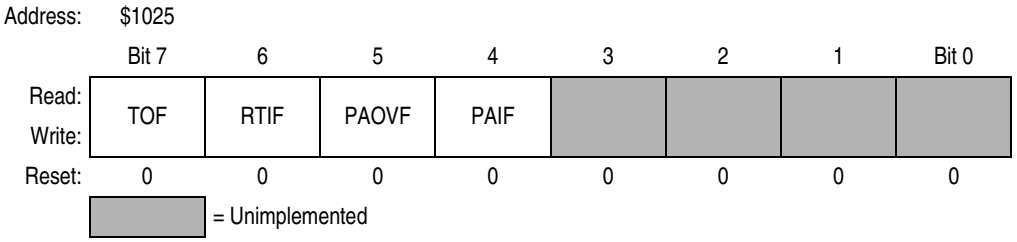


Figure 9-20. Timer Interrupt Flag 2 Register (TFLG2)

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

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

RTIF — Real-Time (Periodic) Interrupt Flag
Refer to 9.5 Real-Time Interrupt (RTI).

PAOVF — Pulse Accumulator Overflow Interrupt Flag
Refer to 9.7 Pulse Accumulator.

PAIF — Pulse Accumulator Input Edge Interrupt Flag
Refer to 9.7 Pulse Accumulator.

Bits [3:0] — Unimplemented
Always read 0

9.5 Real-Time Interrupt (RTI)

The real-time interrupt (RTI) feature, used to generate hardware interrupts at a fixed periodic rate, is controlled and configured by two bits (RTR1 and RTR0) in the pulse accumulator control (PACTL) register. The RTII bit in the TMSK2 register enables the interrupt capability. The four different rates available are a product of the MCU oscillator frequency and the value of bits RTR[1:0]. Refer to Table 9-5, which shows the periodic real-time interrupt rates.

Table 9-5. RTI Rates

RTR[1:0]	E = 3 MHz	E = 2 MHz	E = 1 MHz	E = X MHz
0 0	2.731 ms	4.096 ms	8.192 ms	$(E/2^{13})$
0 1	5.461 ms	8.192 ms	16.384 ms	$(E/2^{14})$
1 0	10.923 ms	16.384 ms	32.768 ms	$(E/2^{15})$
1 1	21.845 ms	32.768 ms	65.536 ms	$(E/2^{16})$

The clock source for the RTI function is a free-running clock that cannot be stopped or interrupted except by reset. This clock causes the time between successive RTI timeouts to be a constant that is

Chapter 10
Electrical Characteristics

10.1 Introduction

This section contains electrical specifications for the M68HC11 E-series devices.

10.2 Maximum Ratings for Standard and Extended Voltage Devices

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

NOTE

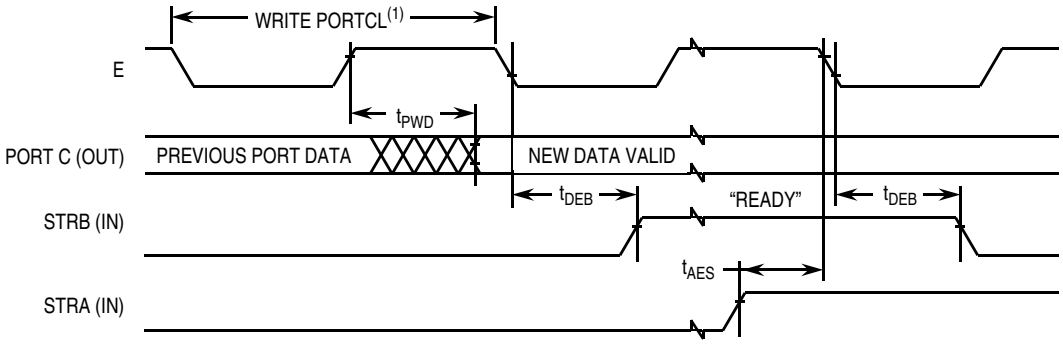
This device is not guaranteed to operate properly at the maximum ratings. Refer to 10.5 DC Electrical Characteristics, 10.6 Supply Currents and Power Dissipation, 10.7 MC68L11E9/E20 DC Electrical Characteristics, and 10.8 MC68L11E9/E20 Supply Currents and Power Dissipation for guaranteed operating conditions.

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

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

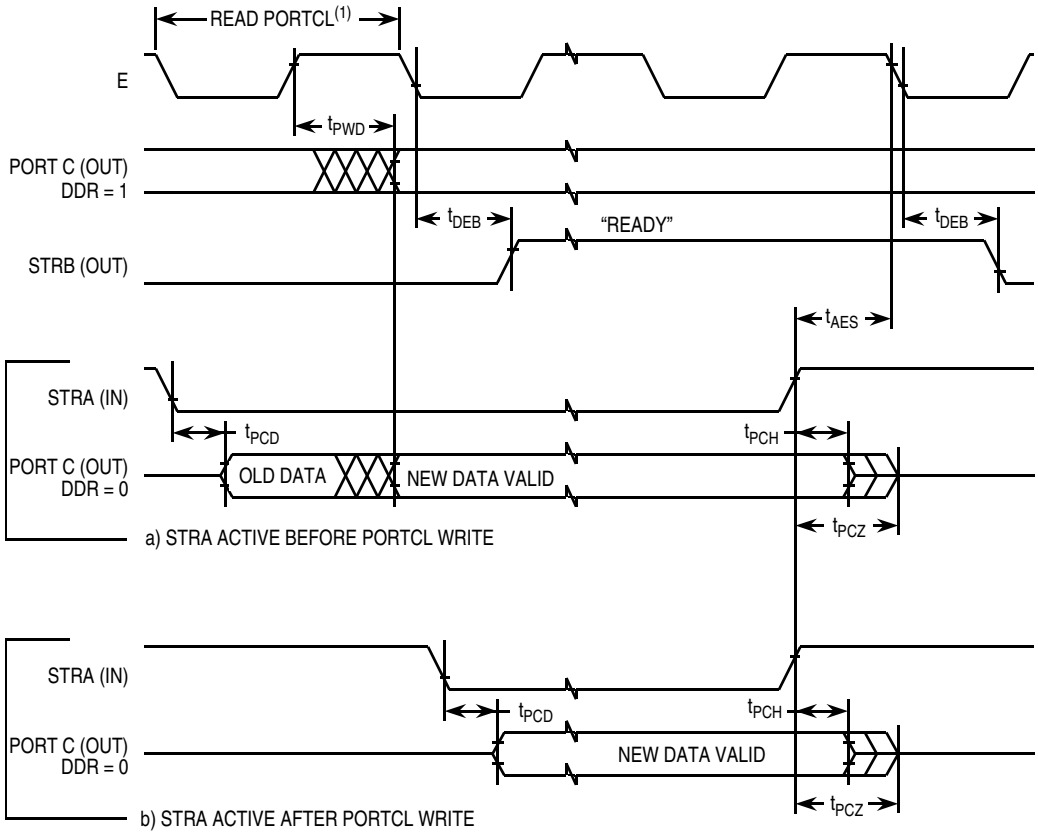
NOTE

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



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

Figure 10-12. Port C Output Handshake Timing Diagram

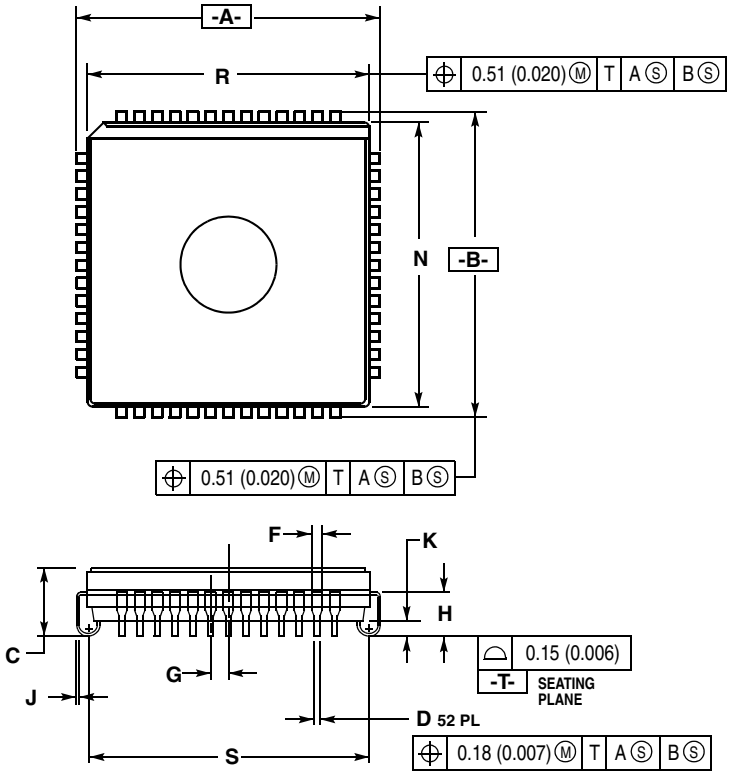


Notes:
1. After reading PIOC with STAF set
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

Figure 10-13. 3-State Variation of Output Handshake Timing Diagram (STRA Enables Output Buffer)

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

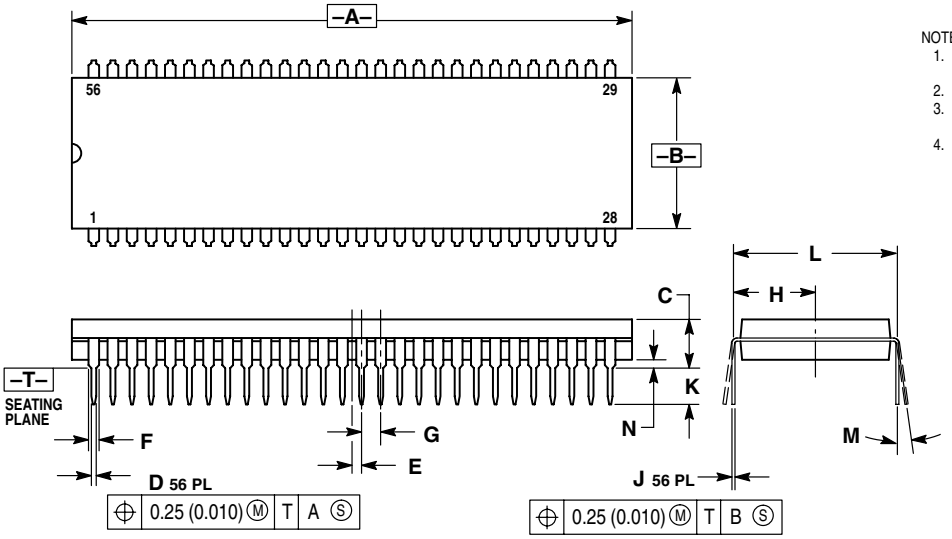
11.6 52-Pin Windowed Ceramic-Leaded Chip Carrier (Case 778B)



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION R AND N DO NOT INCLUDE GLASS PROTRUSION. GLASS PROTRUSION TO BE 0.25 (0.010) MAXIMUM.
4. ALL DIMENSIONS AND TOLERANCES INCLUDE LEAD TRIM OFFSET AND LEAD

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	0.785	0.795	19.94	20.19
B	0.785	0.795	19.94	20.19
C	0.165	0.200	4.20	5.08
D	0.017	0.021	0.44	0.53
F	0.026	0.032	0.67	0.81
G	0.050 BSC		1.27 BSC	
H	0.090	0.130	2.29	3.30
J	0.006	0.010	0.16	0.25
K	0.035	0.045	0.89	1.14
N	0.735	0.756	18.67	19.20
R	0.735	0.756	18.67	19.20
S	0.690	0.730	17.53	18.54

11.9 56-Pin Dual in-Line Package (Case 859)



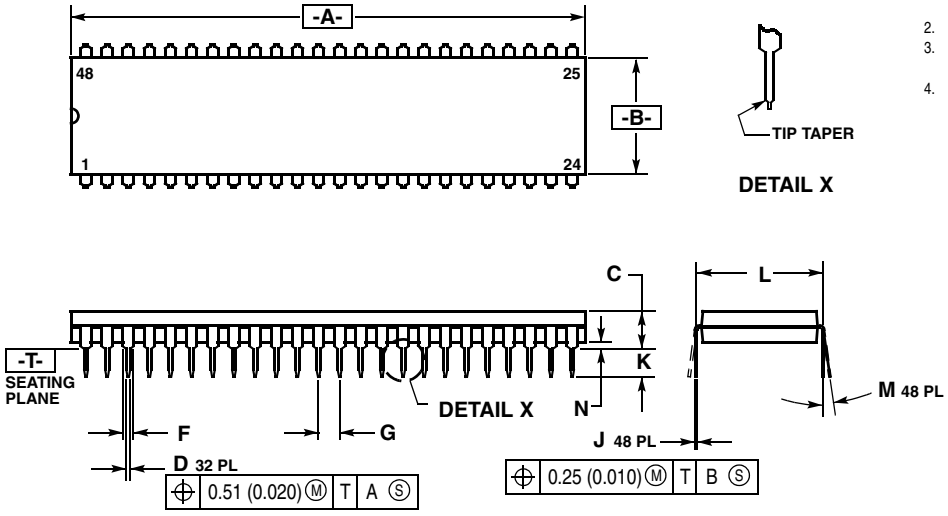
NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.
4. DIMENSIONS A AND B DO NOT INCLUDE MOLD FLASH. MAXIMUM MOLD FLASH 0.25 (0.010)

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	2.035	2.065	51.69	52.45
B	0.540	0.560	13.72	14.22
C	0.155	0.200	3.94	5.08
D	0.014	0.022	0.36	0.56
E	0.035 BSC		0.89 BSC	
F	0.032	0.046	0.81	1.17
G	0.070 BSC		1.778 BSC	
H	0.300 BSC		7.62 BSC	
J	0.008	0.015	0.20	0.38
K	0.115	0.135	2.92	3.43
L	0.600 BSC		15.24 BSC	
M	0 °	15 °	0 °	15 °
N	0.020	0.040	0.51	1.02

11.10 48-Pin Plastic DIP (Case 767)

NOTE

The MC68HC811E2 is the only member of the E series that is offered in a 48-pin plastic dual in-line package.



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION L TO CENTER OF LEAD WHEN FORMED PARALLEL.
4. DIMENSIONS A AND B DO NOT INCLUDE MOLD FLASH. MAXIMUM MOLD FLASH 0.25 (0.010).

	INCHES		MILLIMETERS	
DIM	MIN	MAX	MIN	MAX
A	2.415	2.445	61.34	62.10
B	0.540	0.560	13.72	14.22
C	0.155	0.200	3.94	5.08
D	0.014	0.022	0.36	0.55
F	0.040	0.060	1.02	1.52
G	0.100 BSC		2.54 BSC	
H	0.070 BSC		1.79 BSC	
J	0.008	0.015	0.20	0.38
K	0.115	0.150	2.92	3.81
L	0.600 BSC		15.24 BSC	
M	0×	15×	0×	15×
N	0.020	0.040	0.51	1.01

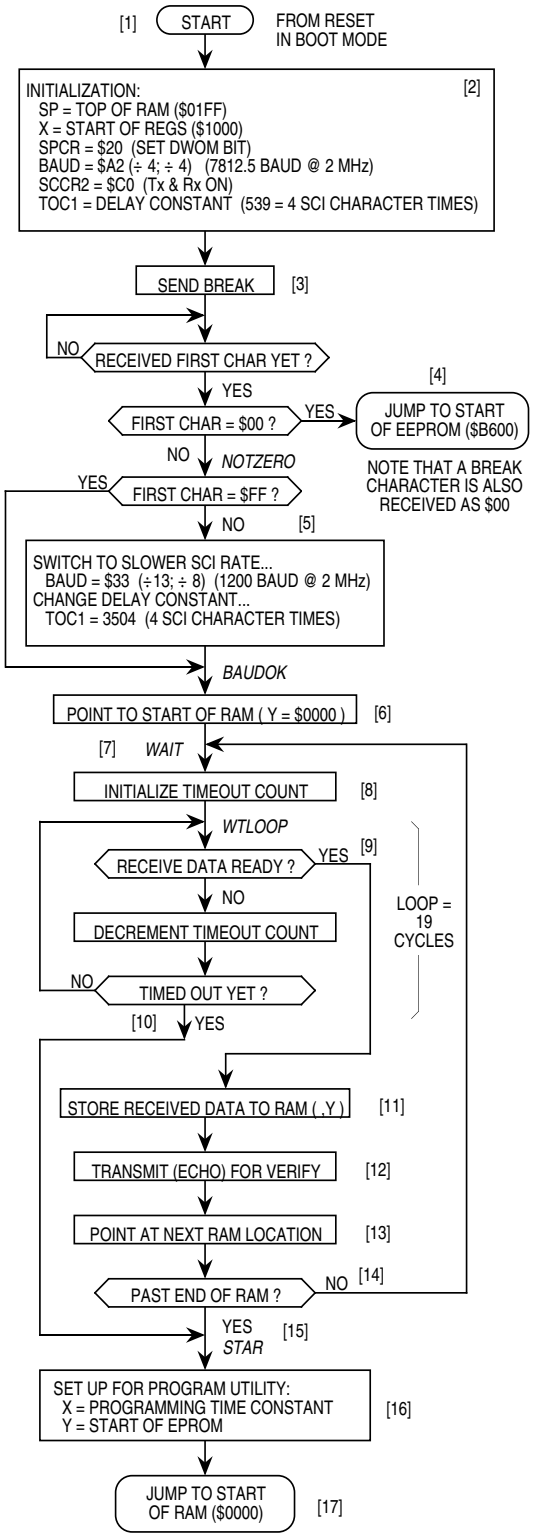


Figure 3. MC68HC711E9 Bootloader Flowchart

UPLOAD Utility

The UPLOAD utility subroutine transfers data from the MCU to a host computer system over the SCI serial data link.

NOTE

Only EPROM versions of the M68HC11 include this utility.

Verification of EPROM contents is one example of how the UPLOAD utility could be used. Before calling this program, the Y index register is loaded (by user firmware) with the address of the first data byte to be uploaded. If a baud rate other than the current SCI baud rate is to be used for the upload process, the user's firmware must also write to the baud register. The UPLOAD program sends successive bytes of data out the SCI transmitter until a reset is issued (the upload loop is infinite).

For a complete commented listing example of the UPLOAD utility, refer to Listing 3. MC68HC711E9 Bootloader ROM.

EPROM Programming Utility

The EPROM programming utility is one way of programming data into the internal EPROM of the MC68HC711E9 MCU. An external 12-V programming power supply is required to program on-chip EPROM. The simplest way to use this utility program is to bootload a 3-byte program consisting of a single jump instruction to the start of the PROGRAM utility program ($BF00). The bootloader program sets the X and Y index registers to default values before jumping to the downloaded program (see [16] at the bottom of Figure 3). When the host computer sees the $FF character, data to be programmed into the EPROM is sent, starting with the character for location $D000. After the last byte to be programmed is sent to the MC68HC711E9 and the corresponding verification data is returned to the host, the programming operation is terminated by resetting the MCU.

The number of bytes to be programmed, the first address to be programmed, and the programming time can be controlled by the user if values other than the default values are desired.

To understand the detailed operation of the EPROM programming utility, refer to Figure 4 during the following discussion. Figure 4 is composed of three interrelated parts. The upper-left portion shows the flowchart of the PROGRAM utility running in the boot ROM of the MCU. The upper-right portion shows the flowchart for the user-supplied driver program running in the host computer. The lower portion of Figure 4 is a timing sequence showing the relationship of operations between the MCU and the host computer. Reference numbers in the flowcharts in the upper half of Figure 4 have matching numbers in the lower half to help the reader relate the three parts of the figure.

The shaded area [1] refers to the software and hardware latency in the MCU leading to the transmission of a character (in this case, the $FF). The shaded area [2] refers to a similar latency in the host computer (in this case, leading to the transmission of the first data character to the MCU).

The overall operation begins when the MCU sends the first character ($FF) to the host computer, indicating that it is ready for the first data character. The host computer sends the first data byte [3] and enters its main loop. The second data character is sent [4], and the host then waits [5] for the first verify byte to come back from the MCU.

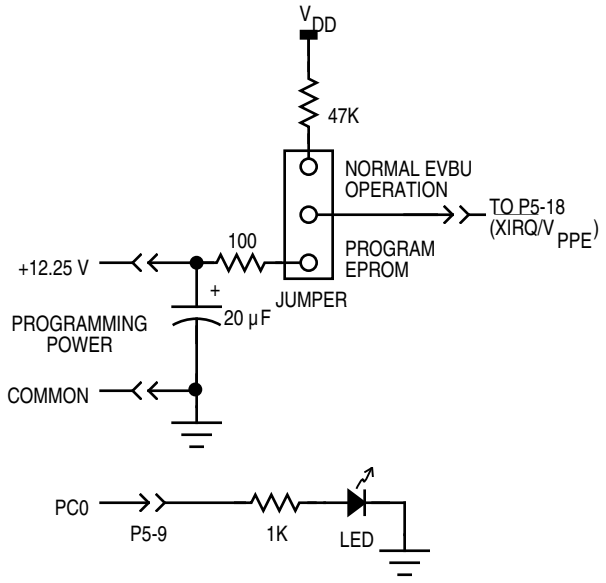


Figure 8. PC-to-MCU Programming Circuit

Lines 50–95 read in the small bootloader from DATA statements at the end of the listing. The source code for this bootloader is presented in the DATA statements. The bootloaded code makes port C bit 0 low, initializes the X and Y registers for use by the EPROM programming utility routine contained in the boot ROM, and then jumps to that routine. The hexadecimal values read in from the DATA statements are converted to binary values by a subroutine. The binary values are then saved as one string (BOOTCODE$).

The next long section of code (lines 97–1250) reads in the S records from an external disk file (in this case, BUF34.S19), converts them to integer, and saves them in an array. The techniques used in this section show how to convert ASCII S records to binary form that can be sent (bootloaded) to an M68HC11.

This S-record translator only looks for the S1 records that contain the actual object code. All other S-record types are ignored.

When an S1 record is found (lines 1000–1024), the next two characters form the hex byte giving the number of hex bytes to follow. This byte is converted to integer by the same subroutine that converted the bootloaded code from the DATA statements. This BYTECOUNT is adjusted by subtracting 3, which accounts for the address and checksum bytes and leaves just the number of object-code bytes in the record.

Starting at line 1100, the 2-byte (4-character) starting address is converted to decimal. This address is the starting address for the object code bytes to follow. An index into the CODE% array is formed by subtracting the base address initialized at the start of the program from the starting address for this S record.

A FOR-NEXT loop starting at line 1130 converts the object code bytes to decimal and saves them in the CODE% array. When all the object code bytes have been converted from the current S record, the program loops back to find the next S1 record.

Step 5

The CONFIG register defaults to hexadecimal 103F on the MC68HC711E9. PCBUG11 needs adressing parameters to allow programming of a specific block of memory so the following parameter must be given.

At the PCbug11 command prompt, type: EEPROM 0.

Then type: EEPROM 103F 103F.

Step 6

Erase the CONFIG to allow byte programming.

At the PCbug11 command prompt, type: EEPROM ERASE BULK 103F.

Step 7

You are now ready to download the program into the EEPROM and EPROM.

At the PCbug11command prompt, type: LOADSC:\MYPROG\MYPROG.S19.

For more details on programming the EPROM, read the engineering bulletin *Programming MC68HC711E9 Devices with PCbug11 and the M68HC11EVB*, Freescale document number EB187.

Step 8

You are now ready to enable the security feature on the MCHC711E9.

At the PCbug11 command prompt type: MS 103F 05.

Step 9

After the programming operation is complete, verifyng the CONFIG on the MCHC711E9 is not possible because in bootstrap mode the default value is always forced.

Step 10

The part is now in secure mode and whatever code you loaded into EEPROM will be erased if you tried to bring the microcontroller up in either expanded mode or bootstrap mode.

NOTE

It is important to note that the microcontroller will work properly in secure mode only in single chip mode.

NOTE

If the part is placed in bootstrap or expanded, the code in EEPROM and RAM will be erased and the microcontroller cannot be reused. The security software will constantly read the NOSEC bit and lock the part.