

Welcome to [E-XFL.COM](https://www.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	38
Program Memory Size	20KB (20K x 8)
Program Memory Type	OTP
EEPROM Size	512 x 8
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/mc68711e20mfne3

Table of Contents

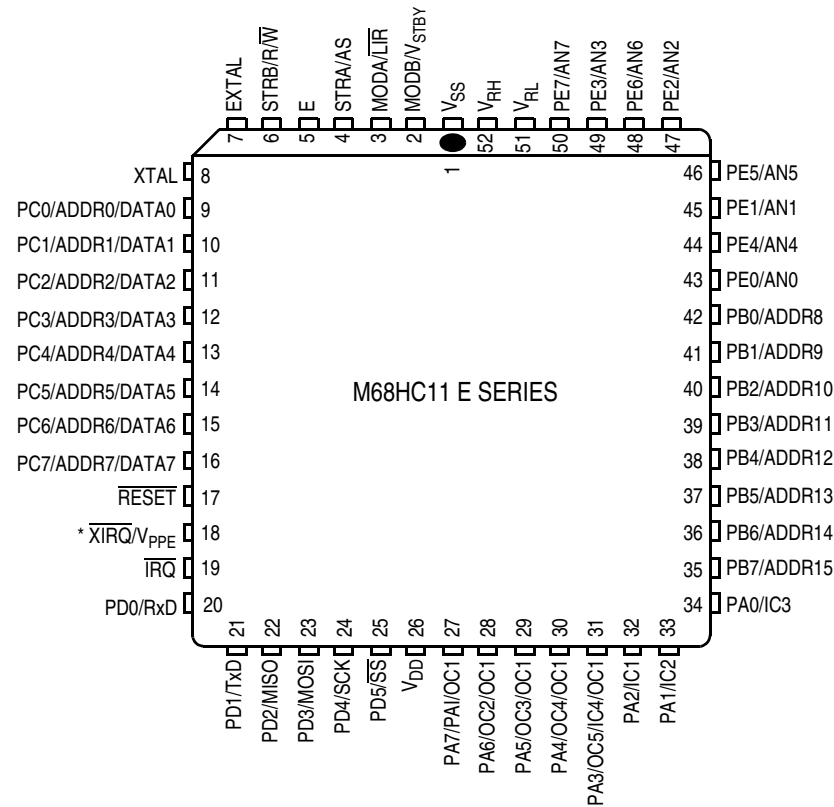
2.4.3	EPROM and EEPROM Programming Control Register	49
2.5	EEPROM	51
2.5.1	EEPROM and CONFIG Programming and Erasure	51
2.5.1.1	Block Protect Register	51
2.5.1.2	EPROM and EEPROM Programming Control Register	53
2.5.1.3	EEPROM Bulk Erase	54
2.5.1.4	EEPROM Row Erase	54
2.5.1.5	EEPROM Byte Erase	55
2.5.1.6	CONFIG Register Programming	55
2.5.2	EEPROM Security	55

Chapter 3 Analog-to-Digital (A/D) Converter

3.1	Introduction	57
3.2	Overview	57
3.2.1	Multiplexer	57
3.2.2	Analog Converter	57
3.2.3	Digital Control	59
3.2.4	Result Registers	59
3.2.5	A/D Converter Clocks	59
3.2.6	Conversion Sequence	59
3.3	A/D Converter Power-Up and Clock Select	60
3.4	Conversion Process	61
3.5	Channel Assignments	61
3.6	Single-Channel Operation	61
3.7	Multiple-Channel Operation	62
3.8	Operation in Stop and Wait Modes	62
3.9	A/D Control/Status Register	62
3.10	A/D Converter Result Registers	64

Chapter 4 Central Processor Unit (CPU)

4.1	Introduction	65
4.2	CPU Registers	65
4.2.1	Accumulators A, B, and D	66
4.2.2	Index Register X (IX)	66
4.2.3	Index Register Y (IY)	66
4.2.4	Stack Pointer (SP)	66
4.2.5	Program Counter (PC)	68
4.2.6	Condition Code Register (CCR)	68
4.2.6.1	Carry/Borrow (C)	68
4.2.6.2	Overflow (V)	68
4.2.6.3	Zero (Z)	68
4.2.6.4	Negative (N)	68
4.2.6.5	Interrupt Mask (I)	69
4.2.6.6	Half Carry (H)	69
4.2.6.7	X Interrupt Mask (X)	69
4.2.6.8	STOP Disable (S)	69



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

Figure 1-2. Pin Assignments for 52-Pin PLCC and CLCC

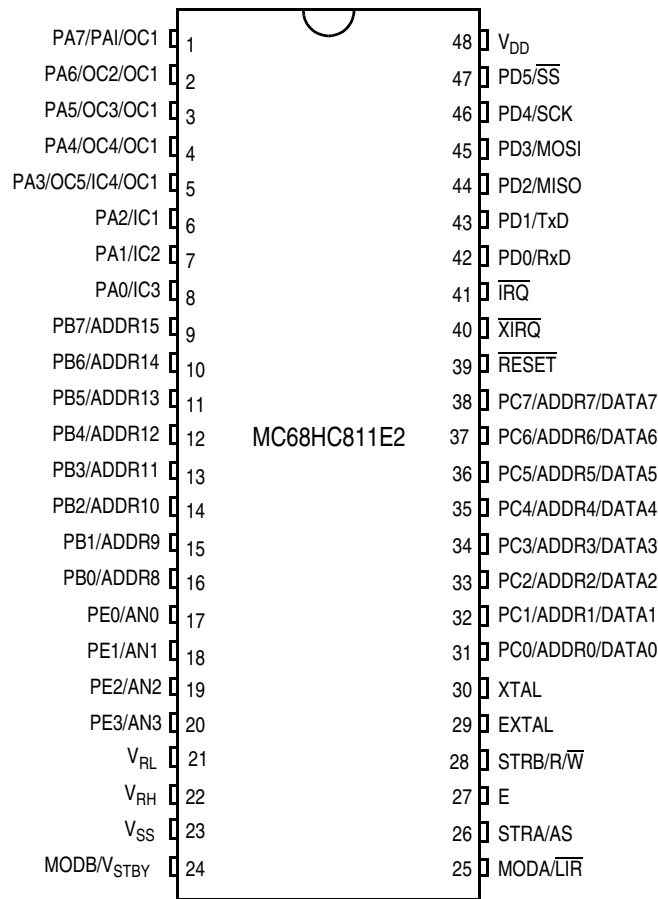


Figure 1-6. Pin Assignments for 48-Pin DIP (MC68HC811E2)

2.4.3 EPROM and EEPROM Programming Control Register

The EPROM and EEPROM programming control register (PPROG) enables the EPROM programming voltage and controls the latching of data to be programmed.

- For MC68HC711E9, PPROG is also the EEPROM programming control register.
- For the MC68HC711E20, EPROM programming is controlled by the EPROG register and EEPROM programming is controlled by the PPROG register.

Address: \$103B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ODD	EVEN	ELAT ⁽¹⁾	BYTE	ROW	ERASE	EELAT	EPGM
Write:								
Reset:	0	0	0	0	0	0	0	0

1. MC68HC711E9 only

Figure 2-14. EPROM and EEPROM Programming Control Register (PPROG)

ODD — Program Odd Rows in Half of EEPROM (Test) Bit

Refer to [2.5 EEPROM](#).

EVEN — Program Even Rows in Half of EEPROM (Test) Bit

Refer to [2.5 EEPROM](#).

ELAT — EPROM/OTPROM Latch Control Bit

When ELAT = 1, writes to EPROM cause address and data to be latched and the EPROM/OTPROM cannot be read. ELAT can be read any time. ELAT can be written any time except when EPGM = 1; then the write to ELAT is disabled.

0 = EPROM address and data bus configured for normal reads

1 = EPROM address and data bus configured for programming

For the MC68HC711E9:

- EPGM enables the high voltage necessary for both EEPROM and EPROM/OTPROM programming.
- ELAT and EELAT are mutually exclusive and cannot both equal 1.

BYTE — Byte/Other EEPROM Erase Mode Bit

Refer to [2.5 EEPROM](#).

ROW — Row/All EEPROM Erase Mode Bit

Refer to [2.5 EEPROM](#).

ERASE — Erase Mode Select Bit

Refer to [2.5 EEPROM](#).

EELAT — EEPROM Latch Control Bit

Refer to [2.5 EEPROM](#).

EPGM — EPROM/OTPROM/EEPROM Programming Voltage Enable Bit

EPGM can be read any time and can be written only when ELAT = 1 (for EPROM/OTPROM programming) or when EELAT = 1 (for EEPROM programming).

0 = Programming voltage to EPROM/OTPROM/EEPROM array disconnected

1 = Programming voltage to EPROM/OTPROM/EEPROM array connected

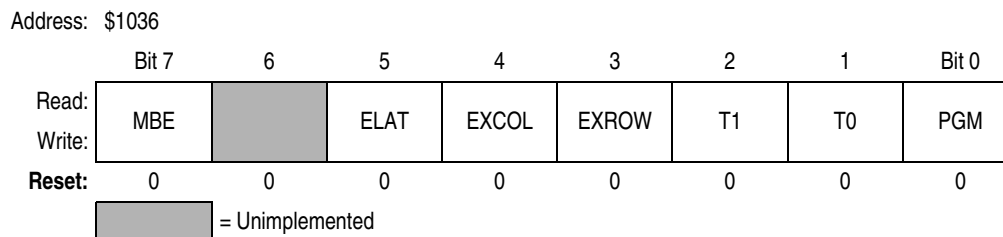


Figure 2-15. MC68HC711E20 EPROM Programming Control Register (EPROG)

MBE — Multiple-Byte Programming Enable Bit

When multiple-byte programming is enabled, address bit 5 is considered a don't care so that bytes with address bit 5 = 0 and address bit 5 = 1 both get programmed. MBE can be read in any mode and always reads 0 in normal modes. MBE can be written only in special modes.

0 = EPROM array configured for normal programming

1 = Program two bytes with the same data

Bit 6 — Unimplemented

Always reads 0

ELAT — EPROM/OTPROM Latch Control Bit

When ELAT = 1, writes to EPROM cause address and data to be latched and the EPROM/OTPROM cannot be read. ELAT can be read any time. ELAT can be written any time except when PGM = 1; then the write to ELAT is disabled.

0 = EPROM/OTPROM address and data bus configured for normal reads

1 = EPROM/OTPROM address and data bus configured for programming

EXCOL — Select Extra Columns Bit

0 = User array selected

1 = User array is disabled and extra columns are accessed at bits [7:0]. Addresses use bits [13:5] and bits [4:0] are don't care. EXCOL can be read and written only in special modes and always returns 0 in normal modes.

EXROW — Select Extra Rows Bit

0 = User array selected

1 = User array is disabled and two extra rows are available. Addresses use bits [7:0] and bits [13:8] are don't care. EXROW can be read and written only in special modes and always returns 0 in normal modes.

T[1:0] — EPROM Test Mode Select Bits

These bits allow selection of either gate stress or drain stress test modes. They can be read and written only in special modes and always read 0 in normal modes.

T1	T0	Function Selected
0	0	Normal mode
0	1	Reserved
1	0	Gate stress
1	1	Drain stress



3.3 A/D Converter Power-Up and Clock Select

Bit 7 of the OPTION register controls A/D converter power-up. Clearing ADPU removes power from and disables the A/D converter system. Setting ADPU enables the A/D converter system. Stabilization of the analog bias voltages requires a delay of as much as 100 μ s after turning on the A/D converter. When the A/D converter system is operating with the MCU E clock, all switching and comparator operations are inherently synchronized to the main MCU clocks. This allows the comparator output to be sampled at relatively quiet times during MCU clock cycles. Since the internal RC oscillator is asynchronous to the MCU clock, there is more error attributable to internal system clock noise. A/D converter accuracy is reduced slightly while the internal RC oscillator is being used (CSEL = 1).

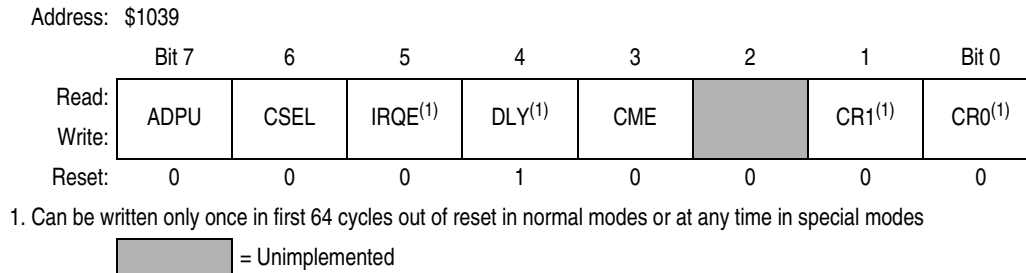


Figure 3-4. System Configuration Options Register (OPTION)

ADPU — A/D Power-Up Bit

- 0 = A/D powered down
- 1 = A/D powered up

CSEL — Clock Select Bit

- 0 = A/D and EEPROM use system E clock.
- 1 = A/D and EEPROM use internal RC clock.

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

Refer to [Chapter 5 Resets and Interrupts](#).

DLY — Enable Oscillator Startup Delay Bit

- 0 = The oscillator startup delay coming out of stop 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

Refer to [Chapter 5 Resets and Interrupts](#) and [Chapter 9 Timing Systems](#).

3.4 Conversion Process

The A/D conversion sequence begins one E-clock cycle after a write to the A/D control/status register, ADCTL. The bits in ADCTL select the channel and the mode of conversion.

An input voltage equal to V_{RL} converts to \$00 and an input voltage equal to V_{RH} converts to \$FF (full scale), with no overflow indication. For ratiometric conversions of this type, the source of each analog input should use V_{RH} as the supply voltage and be referenced to V_{RL} .

3.5 Channel Assignments

The multiplexer allows the A/D converter to select one of 16 analog signals. Eight of these channels correspond to port E input lines to the MCU, four of the channels are internal reference points or test functions, and four channels are reserved. Refer to [Table 3-1](#).

Table 3-1. Converter Channel Assignments

Channel Number	Channel Signal	Result in ADRx if MULT = 1
1	AN0	ADR1
2	AN1	ADR2
3	AN2	ADR3
4	AN3	ADR4
5	AN4	ADR1
6	AN5	ADR2
7	AN6	ADR3
8	AN7	ADR4
9 – 12	Reserved	—
13	$V_{RH}^{(1)}$	ADR1
14	$V_{RL}^{(1)}$	ADR2
15	$(V_{RH})/2^{(1)}$	ADR3
16	Reserved ⁽¹⁾	ADR4

1. Used for factory testing

3.6 Single-Channel Operation

The two types of single-channel operation are:

1. When SCAN = 0, the single selected channel is converted four consecutive times. The first result is stored in A/D result register 1 (ADR1), and the fourth result is stored in ADR4. After the fourth conversion is complete, all conversion activity is halted until a new conversion command is written to the ADCTL register.
2. When SCAN = 1, conversions continue to be performed on the selected channel with the fifth conversion being stored in register ADR1 (overwriting the first conversion result), the sixth conversion overwriting ADR2, and so on.

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
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

:

– 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

Chapter 5

Resets and Interrupts

5.1 Introduction

Resets and interrupt operations load the program counter with a vector that points to a new location from which instructions are to be fetched. A reset immediately stops execution of the current instruction and forces the program counter to a known starting address. Internal registers and control bits are initialized so the MCU can resume executing instructions. An interrupt temporarily suspends normal program execution while an interrupt service routine is being executed. After an interrupt has been serviced, the main program resumes as if there had been no interruption.

5.2 Resets

The four possible sources of reset are:

- Power-on reset (POR)
- External reset ($\overline{\text{RESET}}$)
- Computer operating properly (COP) reset
- Clock monitor reset

POR and $\overline{\text{RESET}}$ share the normal reset vector. COP reset and the clock monitor reset each has its own vector.

5.2.1 Power-On Reset (POR)

A positive transition on V_{DD} generates a power-on reset (POR), which is used only for power-up conditions. POR cannot be used to detect drops in power supply voltages. A $4064 t_{CYC}$ (internal clock cycle) delay after the oscillator becomes active allows the clock generator to stabilize. If $\overline{\text{RESET}}$ is at logical 0 at the end of $4064 t_{CYC}$, the CPU remains in the reset condition until $\overline{\text{RESET}}$ goes to logical 1.

The POR circuit only initializes internal circuitry during cold starts. Refer to [Figure 1-7. External Reset Circuit](#).

NOTE

It is important to protect the MCU during power transitions. Most M68HC11 systems need an external circuit that holds the $\overline{\text{RESET}}$ pin low whenever V_{DD} is below the minimum operating level. This external voltage level detector, or other external reset circuits, are the usual source of reset in a system.

5.3.9 Analog-to-Digital (A/D) Converter

The analog-to-digital (A/D) converter configuration is indeterminate after reset. The ADPU bit is cleared by reset, which disables the A/D system. The conversion complete flag is indeterminate.

5.3.10 System

The EEPROM programming controls are disabled, so the memory system is configured for normal read operation. PSEL[3:0] are initialized with the value %0110, causing the external $\overline{\text{IRQ}}$ pin to have the highest I-bit interrupt priority. The $\overline{\text{IRQ}}$ pin is configured for level-sensitive operation (for wired-OR systems). The RBOOT, SMOD, and MDA bits in the HPRI0 register reflect the status of the MODB and MODA inputs at the rising edge of reset. MODA and MODB inputs select one of the four operating modes. After reset, writing SMOD and MDA in special modes causes the MCU to change operating modes. Refer to the description of HPRI0 register in [Chapter 2 Operating Modes and On-Chip Memory](#) for a detailed description of SMOD and MDA. The DLY control bit is set to specify that an oscillator startup delay is imposed upon recovery from stop mode. The clock monitor system is disabled because CME is cleared.

5.4 Reset and Interrupt Priority

Resets and interrupts have a hardware priority that determines which reset or interrupt is serviced first when simultaneous requests occur. Any maskable interrupt can be given priority over other maskable interrupts.

The first six interrupt sources are not maskable. The priority arrangement for these sources is:

1. POR or $\overline{\text{RESET}}$ pin
2. Clock monitor reset
3. COP watchdog reset
4. $\overline{\text{XIRQ}}$ interrupt
5. Illegal opcode interrupt
6. Software interrupt (SWI)

The maskable interrupt sources have this priority arrangement:

1. $\overline{\text{IRQ}}$
2. Real-time interrupt
3. Timer input capture 1
4. Timer input capture 2
5. Timer input capture 3
6. Timer output compare 1
7. Timer output compare 2
8. Timer output compare 3
9. Timer output compare 4
10. Timer input capture 4/output compare 5
11. Timer overflow
12. Pulse accumulator overflow
13. Pulse accumulator input edge
14. SPI transfer complete
15. SCI system (refer to [Figure 5-7](#))

7.7.4 Serial Communication Status Register

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

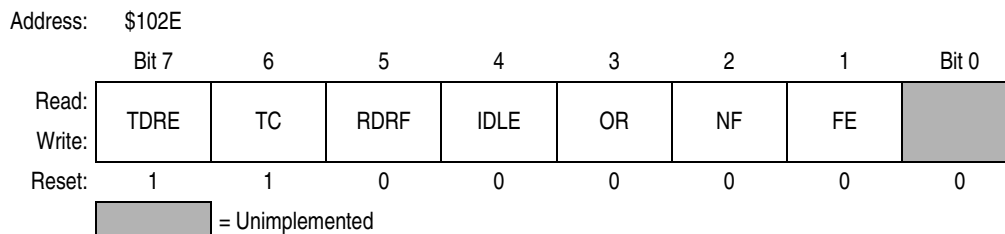


Figure 7-6. Serial Communications Status Register (SCSR)

TDRE — Transmit Data Register Empty Flag

This flag is set when SCDR is empty. Clear the TDRE flag by reading SCSR with TDRE set and then writing to SCDR.

- 0 = SCDR busy
- 1 = SCDR empty

TC — Transmit Complete Flag

This flag is set when the transmitter is idle (no data, preamble, or break transmission in progress). Clear the TC flag by reading SCSR with TC set and then writing to SCDR.

- 0 = Transmitter busy
- 1 = Transmitter idle

RDRF — Receive Data Register Full Flag

This flag is set if a received character is ready to be read from SCDR. Clear the RDRF flag by reading SCSR with RDRF set and then reading SCDR.

- 0 = SCDR empty
- 1 = SCDR full

IDLE — Idle Line Detected Flag

This flag is set if the RxD line is idle. Once cleared, IDLE is not set again until the RxD line has been active and becomes idle again. The IDLE flag is inhibited when RWU = 1. Clear IDLE by reading SCSR with IDLE set and then reading SCDR.

- 0 = RxD line active
- 1 = RxD line idle

OR — Overrun Error Flag

OR is set if a new character is received before a previously received character is read from SCDR. Clear the OR flag by reading SCSR with OR set and then reading SCDR.

- 0 = No overrun
- 1 = Overrun detected

NF — Noise Error Flag

NF is set if majority sample logic detects anything other than a unanimous decision. Clear NF by reading SCSR with NF set and then reading SCDR.

- 0 = Unanimous decision
- 1 = Noise detected

TDRE and TC flags are normally set when the transmitter is first enabled (TE set to 1). The TDRE flag indicates there is room in the transmit queue to store another data character in the TDR. The TIE bit is the local interrupt mask for TDRE. When TIE is 0, TDRE must be polled. When TIE and TDRE are 1, an interrupt is requested.

The TC flag indicates the transmitter has completed the queue. The TCIE bit is the local interrupt mask for TC. When TCIE is 0, TC must be polled. When TCIE is 1 and TC is 1, an interrupt is requested.

Writing a 0 to TE requests that the transmitter stop when it can. The transmitter completes any transmission in progress before actually shutting down. Only an MCU reset can cause the transmitter to stop and shut down immediately. If TE is written to 0 when the transmitter is already idle, the pin reverts to its general-purpose I/O function (synchronized to the bit-rate clock). If anything is being transmitted when TE is written to 0, that character is completed before the pin reverts to general-purpose I/O, but any other characters waiting in the transmit queue are lost. The TC and TDRE flags are set at the completion of this last character, even though TE has been disabled.

7.9 Receiver Flags

The SCI receiver has five status flags, three of which can generate interrupt requests. The status flags are set by the SCI logic in response to specific conditions in the receiver. These flags can be read (polled) at any time by software. Refer to [Figure 7-10](#), which shows SCI interrupt arbitration.

When an overrun takes place, the new character is lost, and the character that was in its way in the parallel RDR is undisturbed. RDRF is set when a character has been received and transferred into the parallel RDR. The OR flag is set instead of RDRF if overrun occurs. A new character is ready to be transferred into RDR before a previous character is read from RDR.

The NF and FE flags provide additional information about the character in the RDR, but do not generate interrupt requests.

The last receiver status flag and interrupt source come from the IDLE flag. The RxD line is idle if it has constantly been at logic 1 for a full character time. The IDLE flag is set only after the RxD line has been busy and becomes idle, which prevents repeated interrupts for the whole time RxD remains idle.

10.12 MC68L11E9/E20 Peripheral Port Timing

Characteristic ^{(1) (2)}	Symbol	1.0 MHz		2.0 MHz		Unit
		Min	Max	Min	Max	
Frequency of operation E-clock frequency	f_o	dc	1.0	dc	2.0	MHz
E-clock period	t_{CYC}	1000	—	500	—	ns
Peripheral data setup time MCU read of ports A, C, D, and E	t_{PDSU}	100	—	100	—	ns
Peripheral data hold time MCU read of ports A, C, D, and E	t_{PDH}	50	—	50	—	ns
Delay time, peripheral data write $t_{PWD} = 1/4 t_{CYC} + 150$ ns MCU writes to port A MCU writes to ports B, C, and D	t_{PWD}	— —	250 400	— —	250 275	ns
Port C input data setup time	t_{IS}	60	—	60	—	ns
Port C input data hold time	t_{IH}	100	—	100	—	ns
Delay time, E fall to STRB $t_{DEB} = 1/4 t_{CYC} + 150$ ns	t_{DEB}	—	400	—	275	ns
Setup time, STRA asserted to E fall ⁽³⁾	t_{AES}	0	—	0	—	ns
Delay time, STRA asserted to port C data output valid	t_{PCD}	—	100	—	100	ns
Hold time, STRA negated to port C data	t_{PCH}	10	—	10	—	ns
3-state hold time	t_{PCZ}	—	150	—	150	ns

- $V_{DD} = 3.0$ Vdc to 5.5 Vdc, $V_{SS} = 0$ 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
- Ports C and D timing is valid for active drive. (CWOM and DWOM bits are not set in PIOC and SPCR registers, respectively.)
- 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.

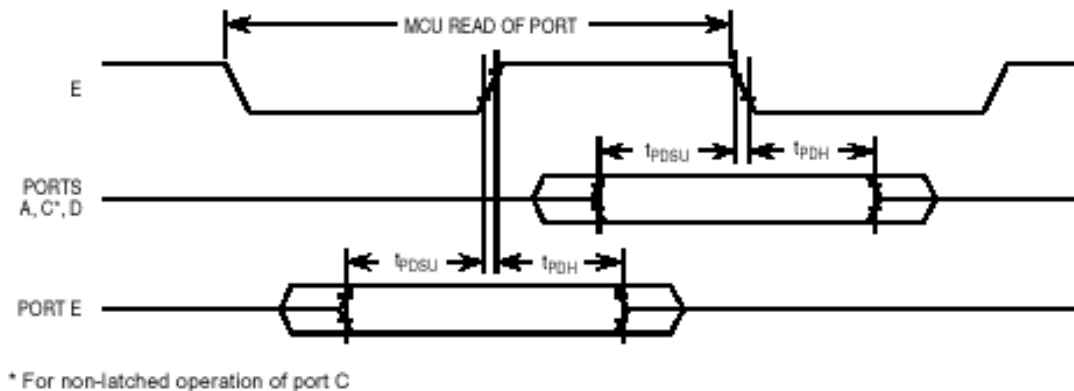


Figure 10-7. Port Read Timing Diagram

10.13 Analog-to-Digital Converter Characteristics

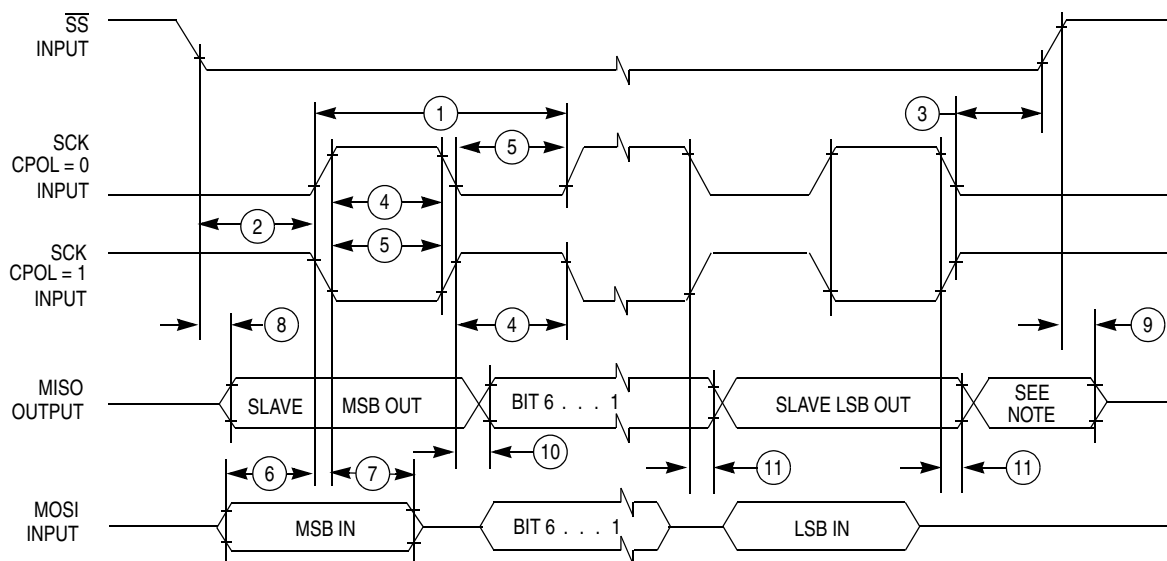
Characteristic ⁽¹⁾	Parameter ⁽²⁾	Min	Absolute	2.0 MHz	3.0 MHz	Unit
				Max	Max	
Resolution	Number of bits resolved by A/D converter	—	8	—	—	Bits
Non-linearity	Maximum deviation from the ideal A/D transfer characteristics	—	—	±1/2	±1	LSB
Zero error	Difference between the output of an ideal and an actual for 0 input voltage	—	—	±1/2	±1	LSB
Full scale error	Difference between the output of an ideal and an actual A/D for full-scale input voltage	—	—	±1/2	±1	LSB
Total unadjusted error	Maximum sum of non-linearity, zero error, and full-scale error	—	—	±1/2	±1/2	LSB
Quantization error	Uncertainty because of converter resolution	—	—	±1/2	±1/2	LSB
Absolute accuracy	Difference between the actual input voltage and the full-scale weighted equivalent of the binary output code, all error sources included	—	—	±1	±2	LSB
Conversion range	Analog input voltage range	V _{RL}	—	V _{RH}	V _{RH}	V
V _{RH}	Maximum analog reference voltage ⁽³⁾	V _{RL}	—	V _{DD} + 0.1	V _{DD} + 0.1	V
V _{RL}	Minimum analog reference voltage ⁽²⁾	V _{SS} - 0.1	—	V _{RH}	V _{RH}	V
ΔV _R	Minimum difference between V _{RH} and V _{RL} ⁽²⁾	3	—	—	—	V
Conversion time	Total time to perform a single A/D conversion: E clock Internal RC oscillator	— —	32 —	— t _{CYC} +32	— t _{CYC} +32	t _{CYC} μs
Monotonicity	Conversion result never decreases with an increase in input voltage; has no missing codes	—	Guaranteed	—	—	—
Zero input reading	Conversion result when V _{In} = V _{RL}	00	—	—	—	Hex
Full scale reading	Conversion result when V _{In} = V _{RH}	—	—	FF	FF	Hex
Sample acquisition time	Analog input acquisition sampling time: E clock Internal RC oscillator	— —	12 —	— 12	— 12	t _{CYC} μs
Sample/hold capacitance	Input capacitance during sample PE[7:0]	—	20 typical	—	—	pF
Input leakage	Input leakage on A/D pins PE[7:0] V _{RL} , V _{RH}	— —	— —	400 1.0	400 1.0	nA μA

1. V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = T_L to T_H, 750 kHz ≤ E ≤ 3.0 MHz, unless otherwise noted

2. Source impedances greater than 10 kΩ affect accuracy adversely because of input leakage.

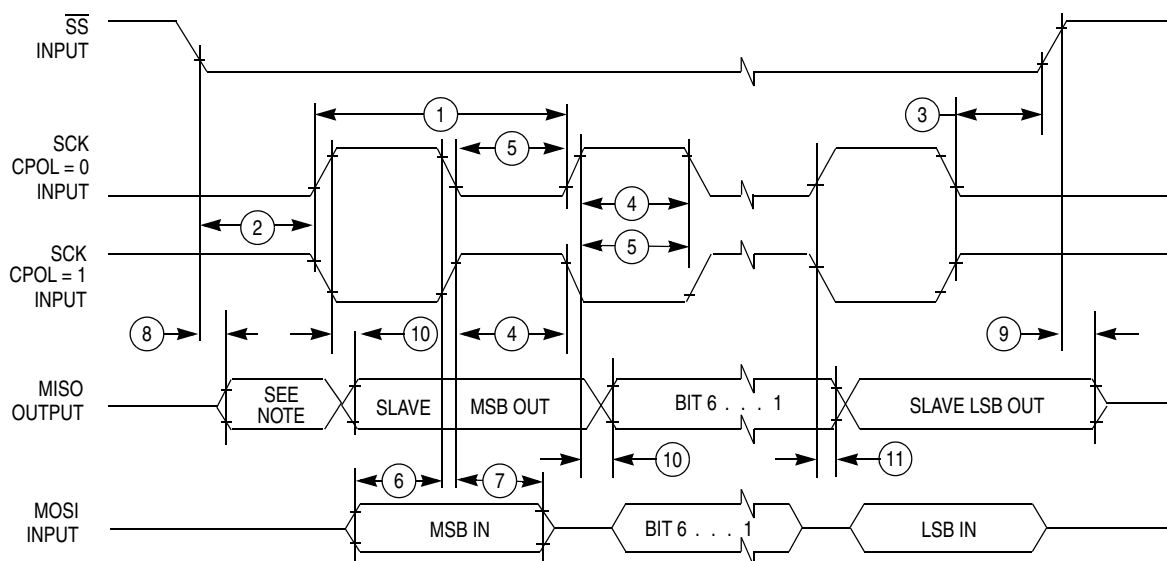
3. Performance verified down to 2.5 V ΔV_R, but accuracy is tested and guaranteed at ΔV_R = 5 V ±10%.

Electrical Characteristics



Note: Not defined but normally MSB of character just received

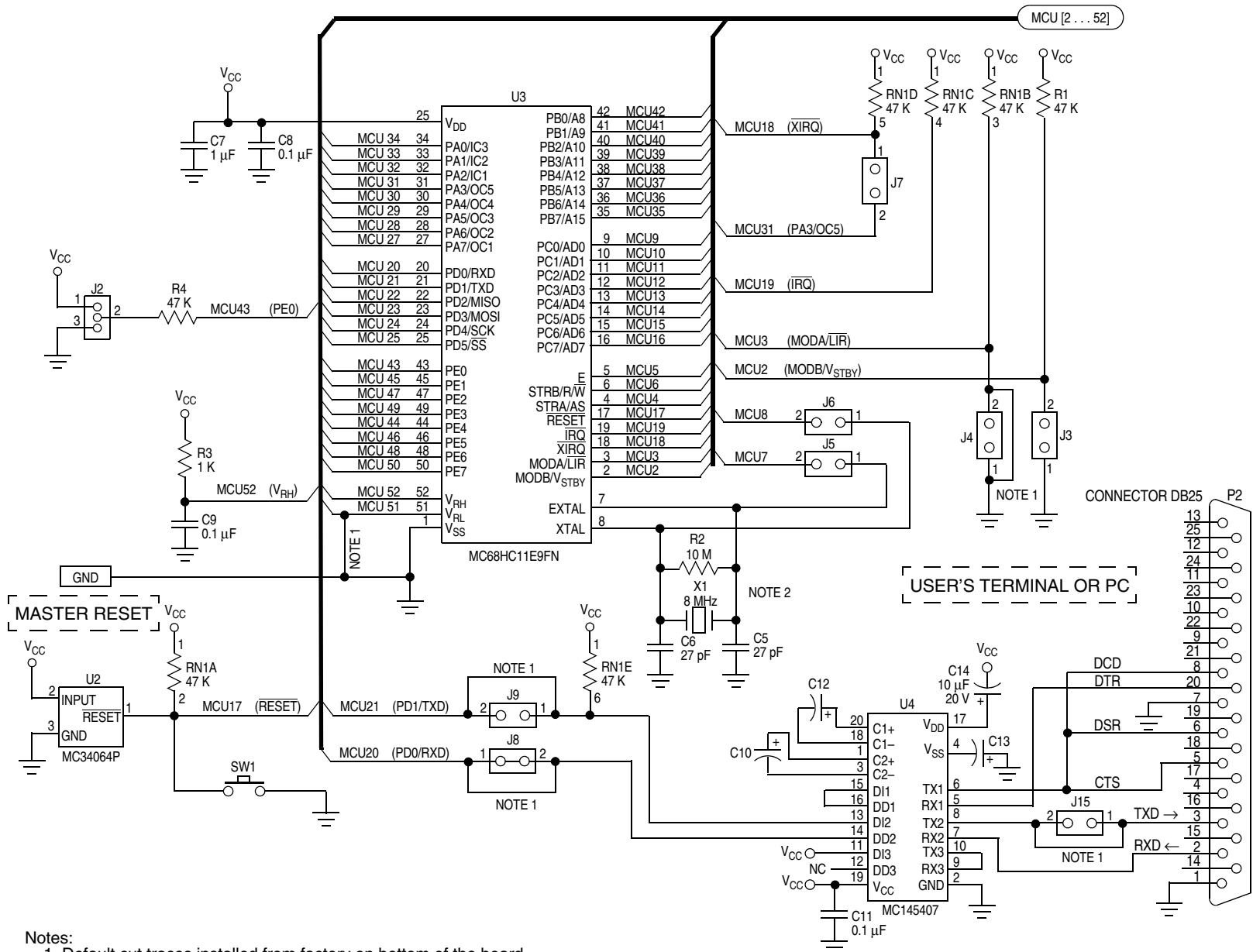
A) SPI Slave Timing (CPHA = 0)



Note: Not defined but normally LSB of character previously transmitted

B) SPI Slave Timing (CPHA = 1)

Figure 11-15. SPI Timing Diagram (Sheet 2 of 2)



- Notes:
1. Default cut traces installed from factory on bottom of the board.
 2. X1 is shipped as a ceramic resonator with built-in capacitors. Holes are provided for a crystal and two capacitors.

Figure B-1. EVBU Schematic Diagram

illustrates the extreme measures used in the bootloader firmware to minimize memory usage. However, such measures are not usually considered good programming technique because they are misleading to someone trying to understand the program or use it as an example.

After initialization, a break character is transmitted [3] by the SCI. By connecting the TxD pin to the Rx pin (with a pullup because of port D wired-OR mode), this break will be received as a \$00 character and cause an immediate jump [4] to the start of the on-chip EEPROM (\$B600 in the MC68HC711E9). This feature is useful to pass control to a program in EEPROM essentially from reset. Refer to [Common Bootstrap Mode Problems](#) before using this feature.

If the first character is received as \$FF, the baud rate is assumed to be the default rate (7812 baud at a 2-MHz E-clock rate). If \$FF was sent at 1200 baud by the host, the SCI will receive the character as \$E0 or \$C0 because of the baud rate mismatch, and the bootloader will switch to 1200 baud [5] for the rest of the download operation. When the baud rate is switched to 1200 baud, the delay constant used to monitor the intercharacter delay also must be changed to reflect the new character time.

At [6], the Y index register is initialized to \$0000 to point to the start of on-chip RAM. The index register Y is used to keep track of where the next received data byte will be stored in RAM. The main loop for loading begins at [7].

The number of data bytes in the downloaded program can be any number between 0 and 512 bytes (the size of on-chip RAM). This procedure is called "variable-length download" and is accomplished by ending the download sequence when an idle time of at least four character times occurs after the last character to be downloaded. In M68HC11 Family members which have 256 bytes of RAM, the download length is fixed at exactly 256 bytes plus the leading \$FF character.

The intercharacter delay counter is started [8] by loading the delay constant from TOC1 into the X index register. The 19-E-cycle wait loop is executed repeatedly until either a character is received [9] or the allowed intercharacter delay time expires [10]. For 7812 baud, the delay constant is 10,241 E cycles (539 x 19 E cycles per loop). Four character times at 7812 baud is 10,240 E cycles (baud prescale of 4 x baud divider of 4 x 16 internal SCI clocks/bit time x 10 bit times/character x 4 character times). The delay from reset to the initial \$FF character is not critical since the delay counter is not started until after the first character (\$FF) is received.

To terminate the bootloading sequence and jump to the start of RAM without downloading any data to the on-chip RAM, simply send \$FF and nothing else. This feature is similar to the jump to EEPROM at [4] except the \$FF causes a jump to the start of RAM. This procedure requires that the RAM has been loaded with a valid program since it would make no sense to jump to a location in uninitialized memory.

After receiving a character, the downloaded byte is stored in RAM [11]. The data is transmitted back to the host [12] as an indication that the download is progressing normally. At [13], the RAM pointer is incremented to the next RAM address. If the RAM pointer has not passed the end of RAM, the main download loop (from [7] to [14]) is repeated.

When all data has been downloaded, the bootloader goes to [16] because of an intercharacter delay timeout [10] or because the entire 512-byte RAM has been filled [15]. At [16], the X and Y index registers are set up for calling the PROGRAM utility routine, which saves the user from having to do this in a downloaded program. The PROGRAM utility is fully explained in [EPROM Programming Utility](#). The final step of the bootloader program is to jump to the start of RAM [17], which starts the user's downloaded program.

Connecting RxD to V_{SS} Does Not Cause the SCI to Receive a Break

To force an immediate jump to the start of EEPROM, the bootstrap firmware looks for the first received character to be \$00 (or break). The data reception logic in the SCI looks for a 1-to-0 transition on the RxD pin to synchronize to the beginning of a receive character. If the RxD pin is tied to ground, no 1-to-0 transition occurs. The SCI transmitter sends a break character when the bootloader firmware starts, and this break character can be fed back to the RxD pin to cause the jump to EEPROM. Since TxD is configured as an open-drain output, a pullup resistor is required.

\$FF Character Is Required before Loading into RAM

The initial character (usually \$FF) that sets the download baud rate is often forgotten.

Original M68HC11 Versions Required Exactly 256 Bytes to be Downloaded to RAM

Even users that know about the 256 bytes of download data sometimes forget the initial \$FF that makes the total number of bytes required for the entire download operation equal to 256 + 1 or 257 bytes.

Variable-Length Download

When on-chip RAM surpassed 256 bytes, the time required to serially load this many characters became more significant. The variable-length download feature allows shorter programs to be loaded without sacrificing compatibility with earlier fixed-length download versions of the bootloader. The end of a download is indicated by an idle RxD line for at least four character times. If a personal computer is being used to send the download data to the MCU, there can be problems keeping characters close enough together to avoid tripping the end-of-download detect mechanism. Using 1200 as the baud rate rather than the faster default rate may help this problem.

Assemblers often produce S-record encoded programs which must be converted to binary before bootloading them to the MCU. The process of reading S-record data from a file and translating it to binary can be slow, depending on the personal computer and the programming language used for the translation. One strategy that can be used to overcome this problem is to translate the file into binary and store it into a RAM array before starting the download process. Data can then be read and downloaded without the translation or file-read delays.

The end-of-download mechanism goes into effect when the initial \$FF is received to set the baud rate. Any amount of time may pass between reset and when the \$FF is sent to start the download process.

EPROM/OTP Versions of M68HC11 Have an EPROM Emulation Mode

The conditions that configure the MCU for EPROM emulation mode are essentially the same as those for resetting the MCU in bootstrap mode. While $\overline{\text{RESET}}$ is low and mode select pins are configured for bootstrap mode (low), the MCU is configured for EPROM emulation mode.

The port pins that are used for EPROM data I/O lines may be inputs or outputs, depending on the pin that is emulating the EPROM output enable pin ($\overline{\text{OE}}$). To make these data pins appear as high-impedance inputs as they would on a non-EPROM part in reset, connect the $\overline{\text{PB7}}/(\overline{\text{OE}})$ pin to a pullup resistor.

Step 6

Erase the CONFIG to allow programming of NOSEC bit (bit 3). It is also recommended to program the EEPROM at this point before programming the CONFIG register. Refer to the engineering bulletin *Programming MC68HC811E2 Devices with PCbug11 and the M68HC711E9PGMR*, Freescale document number EB184.

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

Step 7

You are now ready to enable the security feature on the MCHC811E2.

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

The value \$05 assumes the EEPROM is to be mapped from \$0800 to \$0FFF.

Step 8

After the programming operation is complete, verifying the CONFIG on the MCHC811E2 is not possible because in bootstrap mode the default value is always forced.

Step 9

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. The microcontroller will work properly in the secure mode only in single chip mode.

NOTE

If the part is placed in bootstrap mode or expanded mode, the code in EEPROM and RAM will be erased the microcontroller can be reused.