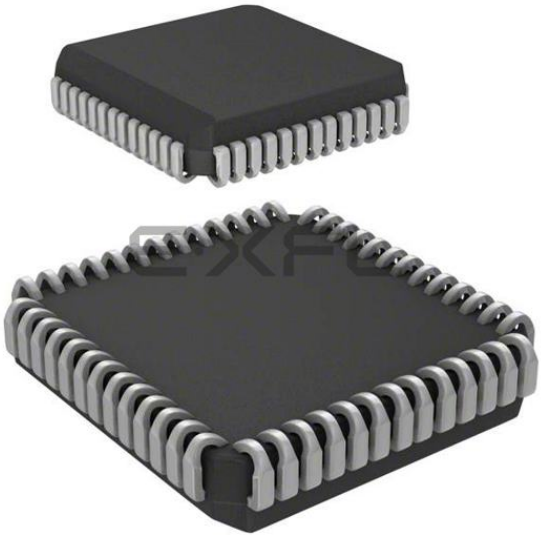


Welcome to E-XFL.COM

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.

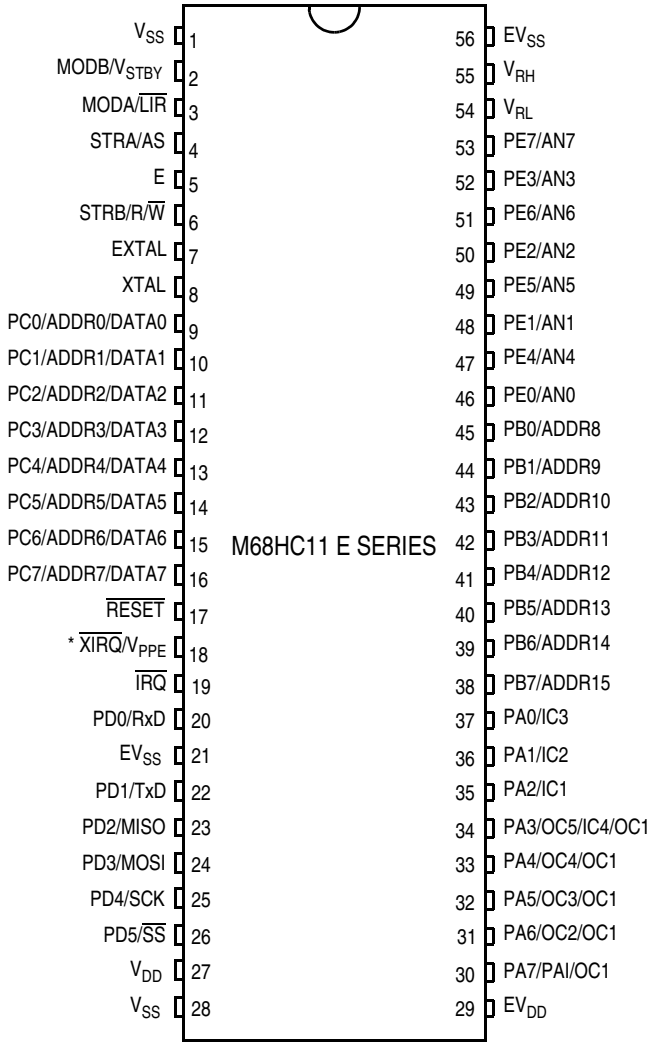
Applications of "Embedded - Microcontrollers"

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	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 ~ 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/mc68hc711e20cfn3

4.3 Data Types . . . 69
4.4 Opcodes and Operands . . . 70
4.5 Addressing Modes . . . 70
4.5.1 Immediate . . . 70
4.5.2 Direct . . . 70
4.5.3 Extended . . . 71
4.5.4 Indexed . . . 71
4.5.5 Inherent . . . 71
4.5.6 Relative . . . 71
4.6 Instruction Set . . . 71

Chapter 5
Resets and Interrupts

5.1 Introduction . . . 79
5.2 Resets . . . 79
5.2.1 Power-On Reset (POR) . . . 79
5.2.2 External Reset (RESET) . . . 80
5.2.3 Computer Operating Properly (COP) Reset . . . 80
5.2.4 Clock Monitor Reset . . . 81
5.2.5 System Configuration Options Register . . . 82
5.2.6 Configuration Control Register . . . 83
5.3 Effects of Reset . . . 83
5.3.1 Central Processor Unit (CPU) . . . 83
5.3.2 Memory Map . . . 84
5.3.3 Timer . . . 84
5.3.4 Real-Time Interrupt (RTI) . . . 84
5.3.5 Pulse Accumulator . . . 84
5.3.6 Computer Operating Properly (COP) . . . 84
5.3.7 Serial Communications Interface (SCI) . . . 84
5.3.8 Serial Peripheral Interface (SPI) . . . 84
5.3.9 Analog-to-Digital (A/D) Converter . . . 85
5.3.10 System . . . 85
5.4 Reset and Interrupt Priority . . . 85
5.4.1 Highest Priority Interrupt and Miscellaneous Register . . . 86
5.5 Interrupts . . . 87
5.5.1 Interrupt Recognition and Register Stacking . . . 88
5.5.2 Non-Maskable Interrupt Request (XIRQ) . . . 89
5.5.3 Illegal Opcode Trap . . . 89
5.5.4 Software Interrupt (SWI) . . . 90
5.5.5 Maskable Interrupts . . . 90
5.5.6 Reset and Interrupt Processing . . . 90
5.6 Low-Power Operation . . . 90
5.6.1 Wait Mode . . . 90
5.6.2 Stop Mode . . . 95



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

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

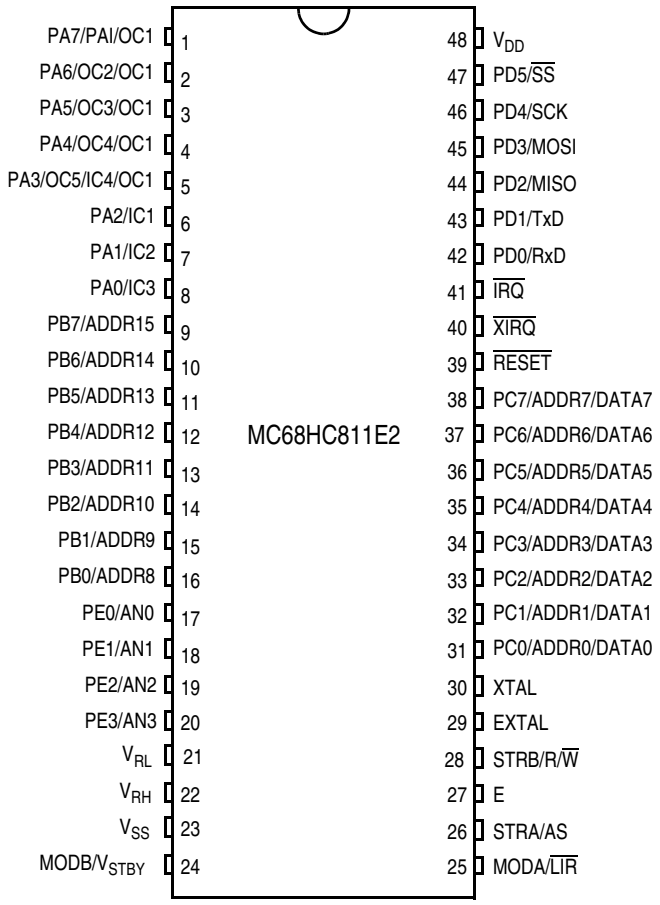


Figure 1-6. Pin Assignments for 48-Pin DIP (MC68HC811E2)

Table 1-1. Port Signal Functions

Port/Bit	Single-Chip and Bootstrap Modes	Expanded and Test Modes
PA0	PA0/IC3	
PA1	PA1/IC2	
PA2	PA2/IC1	
PA3	PA3/OC5/IC4/OC1	
PA4	PA4/OC4/OC1	
PA5	PA5/OC3/OC1	
PA6	PA6/OC2/OC1	
PA7	PA7/PAI/OC1	
PB0	PB0	ADDR8
PB1	PB1	ADDR9
PB2	PB2	ADDR10
PB3	PB3	ADDR11
PB4	PB4	ADDR12
PB5	PB5	ADDR13
PB6	PB6	ADDR14
PB7	PB7	ADDR15
PC0	PC0	ADDR0/DATA0
PC1	PC1	ADDR1/DATA1
PC2	PC2	ADDR2/DATA2
PC3	PC3	ADDR3/DATA3
PC4	PC4	ADDR4/DATA4
PC5	PC5	ADDR5/DATA5
PC6	PC6	ADDR6/DATA6
PC7	PC7	ADDR7/DATA7
PD0	PD0/RxD	
PD1	PD1/TxD	
PD2	PD2/MISO	
PD3	PD3/MOSI	
PD4	PD4/SCK	
PD5	PD5/$\overline{SS}$	
—	STRA	AS
—	STRB	R/$\overline{W}$
PE0	PE0/AN0	
PE1	PE1/AN1	
PE2	PE3/AN2	
PE3	PE3/AN3	
PE4	PE4/AN4	
PE5	PE5/AN5	
PE6	PE6/AN6	
PE7	PE7/AN7	

located in this ROM at $BFC0–$BFFF. The bootstrap ROM contains a small program which initializes the serial communications interface (SCI) and allows the user to download a program into on-chip RAM. The size of the downloaded program can be as large as the size of the on-chip RAM. After a 4-character delay, or after receiving the character for the highest address in RAM, control passes to the loaded program at $0000. Refer to Figure 2-2, Figure 2-3, Figure 2-4, Figure 2-5, and Figure 2-6.

Use of an external pullup resistor is required when using the SCI transmitter pin because port D pins are configured for wired-OR operation by the bootloader. In bootstrap mode, the interrupt vectors are directed to RAM. This allows the use of interrupts through a jump table. Refer to the application note AN1060 entitled *M68HC11 Bootstrap Mode, that is included in this data book*.

2.3 Memory Map

The operating mode determines memory mapping and whether external addresses can be accessed. Refer to Figure 2-2, Figure 2-3, Figure 2-4, Figure 2-5, and Figure 2-6, which illustrate the memory maps for each of the three families comprising the M68HC11 E series of MCUs.

Memory locations for on-chip resources are the same for both expanded and single-chip modes. Control bits in the configuration (CONFIG) register allow EPROM and EEPROM (if present) to be disabled from the memory map. The RAM is mapped to $0000 after reset. It can be placed at any 4-Kbyte boundary ($x000) by writing an appropriate value to the RAM and I/O map register (INIT). The 64-byte register block is mapped to $1000 after reset and also can be placed at any 4-Kbyte boundary ($x000) by writing an appropriate value to the INIT register. If RAM and registers are mapped to the same boundary, the first 64 bytes of RAM will be inaccessible.

Refer to Figure 2-7, which details the MCU register and control bit assignments. Reset states shown are for single-chip mode only.

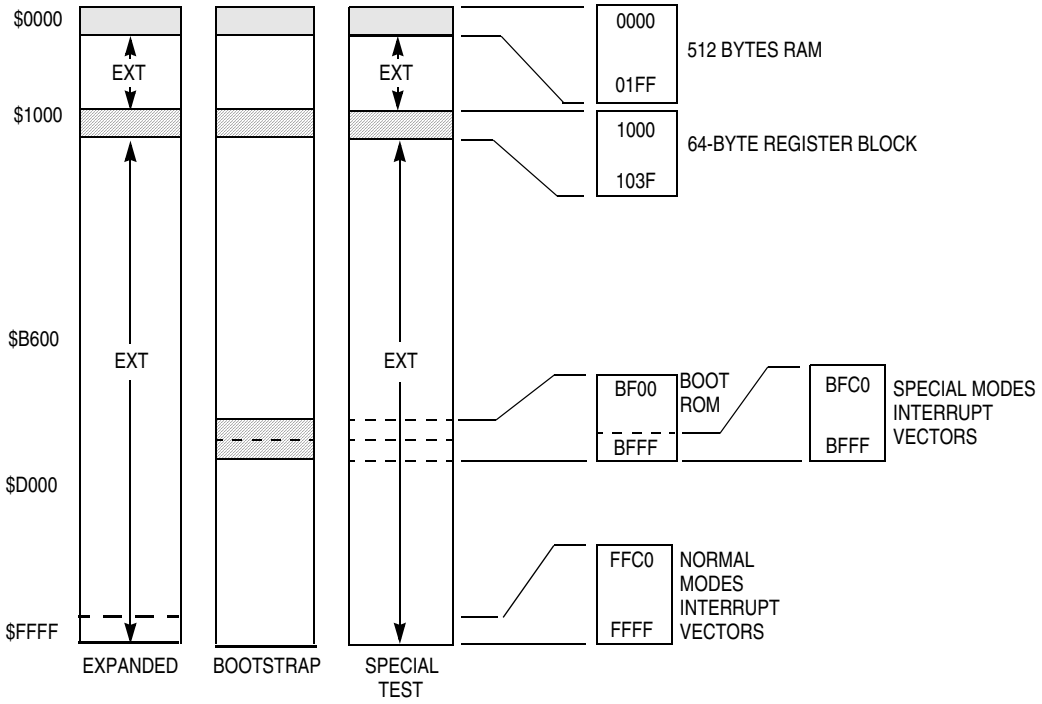


Figure 2-2. Memory Map for MC68HC11E0

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$1031	Analog-to-Digital Results Register 1 (ADR1) See page 64.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$1032	Analog-to-Digital Results Register 2 (ADR2) See page 64.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$1033	Analog-to-Digital Results Register 3 (ADR3) See page 64.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$1034	Analog-to-Digital Results Register 4 (ADR4) See page 64.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$1035	Block Protect Register (BPROT) See page 52.	Read:				PTCON	BPRT3	BPRT2	BPRT1	BPRT0
		Write:								
		Reset:	0	0	0	1	1	1	1	1
$1036	EPROM Programming Control Register (EPROG)(1) See page 53.	Read:	MBE		ELAT	EXCOL	EXROW	T1	T0	PGM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$1037	Reserved		R	R	R	R	R	R	R	R

1. MC68HC711E20 only

$1038	Reserved		R	R	R	R	R	R	R	R
$1039	System Configuration Options Register (OPTION) See page 46.	Read:	ADPU	CSEL	IRQE(1)	DLY(1)	CME		CR1(1)	CR0(1)
		Write:								
		Reset:	0	0	0	1	0	0	0	0
$103A	Arm/Reset COP Timer Circuitry Register (COPRST) See page 81.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$103B	EPROM and EEPROM Programming Control Register (PPROG) See page 49.	Read:	ODD	EVEN	ELAT(2)	BYTE	ROW	ERASE	EELAT	EPGM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$103C	Highest Priority I Bit Interrupt and Miscellaneous Register (HPRIO) See page 41.	Read:	RBOOT	SMOD	MDA	IRV(NE)	PSEL3	PSEL2	PSEL1	PSEL0
		Write:								
		Reset:	0	0	0	0	0	1	1	0
$103D	RAM and I/O Mapping Register (INIT) See page 45.	Read:	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0
		Write:								
		Reset:	0	0	0	0	0	0	0	1

= Unimplemented R = Reserved U = Unaffected

I = Indeterminate after reset

Figure 2-7. Register and Control Bit Assignments (Sheet 5 of 6)

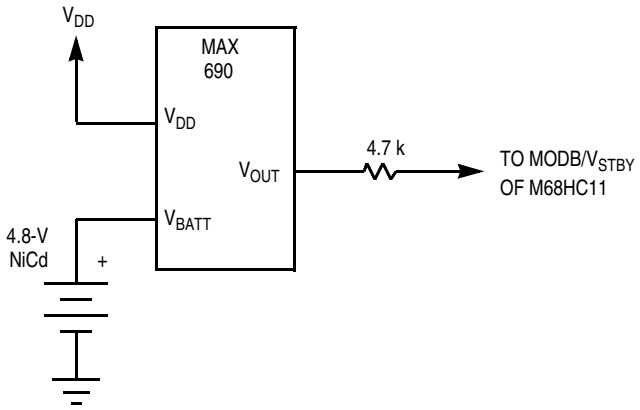


Figure 2-8. RAM Standby MODB/V_{STBY} Connections

The bootloader program is contained in the internal bootstrap ROM. This ROM, which appears as internal memory space at locations $BF00–$BFFF, is enabled only if the MCU is reset in special bootstrap mode.

In expanded modes, the ROM/EPROM/OTPROM (if present) is enabled out of reset and located at the top of the memory map if the ROMON bit in the CONFIG register is set. ROM or EPROM is enabled out of reset in single-chip and bootstrap modes, regardless of the state of ROMON.

For devices with 512 bytes of EEPROM, the EEPROM is located at $B600–$B7FF and has the same read cycle time as the internal ROM. The 512 bytes of EEPROM cannot be remapped to other locations.

For the MC68HC811E2, EEPROM is located at $F800–$FFFF and can be remapped to any 4-Kbyte boundary. EEPROM mapping control bits (EE[3:0] in CONFIG) determine the location of the 2048 bytes of EEPROM and are present only on the MC68HC811E2. Refer to 2.3.3.1 System Configuration Register for a description of the MC68HC811E2 CONFIG register.

EEPROM can be programmed or erased by software and an on-chip charge pump, allowing EEPROM changes using the single V_{DD} supply.

2.3.2 Mode Selection

The four mode variations are selected by the logic states of the MODA and MODB pins during reset. The MODA and MODB logic levels determine the logic state of SMOD and the MDA control bits in the highest priority I-bit interrupt and miscellaneous (HPRIO) register.

After reset is released, the mode select pins no longer influence the MCU operating mode. In single-chip operating mode, the MODA pin is connected to a logic level 0. In expanded mode, MODA is normally connected to V_{DD} through a pullup resistor of 4.7 kΩ. The MODA pin also functions as the load instruction register $\overline{LIR}$ pin when the MCU is not in reset. The open-drain active low $\overline{LIR}$ output pin drives low during the first E cycle of each instruction. The MODB pin also functions as standby power input (V_{STBY}), which allows RAM contents to be maintained in absence of V_{DD}.

Refer to Table 2-1, which is a summary of mode pin operation, the mode control bits, and the four operating modes.

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 HPRIO 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 HPRIO 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)

Table 5-4. Interrupt and Reset Vector Assignments

Vector Address	Interrupt Source	CCR Mask Bit	Local Mask
FFC0, C1 – FFD4, D5	Reserved	—	—
FFD6, D7	SCI serial system • SCI receive data register full • SCI receiver overrun • SCI transmit data register empty • SCI transmit complete • SCI idle line detect	I	RIE RIE TIE TCIE ILIE
FFD8, D9	SPI serial transfer complete	I	SPIE
FFDA, DB	Pulse accumulator input edge	I	PAII
FFDC, DD	Pulse accumulator overflow	I	PAOVI
FFDE, DF	Timer overflow	I	TOI
FFE0, E1	Timer input capture 4/output compare 5	I	I4/O5I
FFE2, E3	Timer output compare 4	I	OC4I
FFE4, E5	Timer output compare 3	I	OC3I
FFE6, E7	Timer output compare 2	I	OC2I
FFE8, E9	Timer output compare 1	I	OC1I
FFEA, EB	Timer input capture 3	I	IC3I
FFEC, ED	Timer input capture 2	I	IC2I
FFEE, EF	Timer input capture 1	I	IC1I
FFF0, F1	Real-time interrupt	I	RTII
FFF2, F3	$\overline{\text{IRQ}}$ (external pin)	I	None
FFF4, F5	$\overline{\text{XIRQ}}$ pin	X	None
FFF6, F7	Software interrupt	None	None
FFF8, F9	Illegal opcode trap	None	None
FFFA, FB	COP failure	None	NOCOP
FFFC, FD	Clock monitor fail	None	CME
FFFE, FF	$\overline{\text{RESET}}$	None	None

5.5.1 Interrupt Recognition and Register Stacking

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

5.5.4 Software Interrupt (SWI)

SWI is an instruction, and thus cannot be interrupted until complete. SWI is not inhibited by the global mask bits in the CCR. Because execution of SWI sets the I mask bit, once an SWI interrupt begins, other interrupts are inhibited until SWI is complete, or until user software clears the I bit in the CCR.

5.5.5 Maskable Interrupts

The maskable interrupt structure of the MCU can be extended to include additional external interrupt sources through the $\overline{\text{IRQ}}$ pin. The default configuration of this pin is a low-level sensitive wired-OR network. When an event triggers an interrupt, a software accessible interrupt flag is set. When enabled, this flag causes a constant request for interrupt service. After the flag is cleared, the service request is released.

5.5.6 Reset and Interrupt Processing

Figure 5-5 and Figure 5-6 illustrate the reset and interrupt process. Figure 5-5 illustrates how the CPU begins from a reset and how interrupt detection relates to normal opcode fetches. Figure 5-6 is an expansion of a block in Figure 5-5 and illustrates interrupt priorities. Figure 5-7 shows the resolution of interrupt sources within the SCI subsystem.

5.6 Low-Power Operation

Both stop mode and wait mode suspend CPU operation until a reset or interrupt occurs. Wait mode suspends processing and reduces power consumption to an intermediate level. Stop mode turns off all on-chip clocks and reduces power consumption to an absolute minimum while retaining the contents of the entire RAM array.

5.6.1 Wait Mode

The WAI opcode places the MCU in wait mode, during which the CPU registers are stacked and CPU processing is suspended until a qualified interrupt is detected. The interrupt can be an external $\overline{\text{IRQ}}$, an $\overline{\text{XIRQ}}$, or any of the internally generated interrupts, such as the timer or serial interrupts. The on-chip crystal oscillator remains active throughout the wait standby period.

The reduction of power in the wait condition depends on how many internal clock signals driving on-chip peripheral functions can be shut down. The CPU is always shut down during wait. While in the wait state, the address/data bus repeatedly runs read cycles to the address where the CCR contents were stacked. The MCU leaves the wait state when it senses any interrupt that has not been masked.

The free-running timer system is shut down only if the I bit is set to 1 and the COP system is disabled by NOCOP being set to 1. Several other systems also can be in a reduced power-consumption state depending on the state of software-controlled configuration control bits. Power consumption by the analog-to-digital (A/D) converter is not affected significantly by the wait condition. However, the A/D converter current can be eliminated by writing the ADPU bit to 0. The SPI system is enabled or disabled by the SPE control bit. The SCI transmitter is enabled or disabled by the TE bit, and the SCI receiver is enabled or disabled by the RE bit. Therefore, the power consumption in wait is dependent on the particular application.

6.2 Port A

Port A shares functions with the timer system and has:

- Three input-only pins
- Three output-only pins
- Two bidirectional I/O pins

Address:	$1000							
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Reset:	I	0	0	0	I	I	I	I
Alternate function:	PAI	OC2	OC3	OC4	IC4/OC5	IC1	IC2	IC3
And/or:	OC1	OC1	OC1	OC1	OC1	—	—	—

I = Indeterminate after reset

Figure 6-1. Port A Data Register (PORTA)

Address:	$1026							
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRA7	PAEWN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0
Reset:	0	0	0	0	0	0	0	0

Figure 6-2. Pulse Accumulator Control Register (PACTL)

DDRA7 — Data Direction for Port A Bit 7

Overridden if an output compare function is configured to control the PA7 pin

0 = Input
1 = Output

The pulse accumulator uses port A bit 7 as the PAI input, but the pin can also be used as general-purpose I/O or as an output compare.

NOTE

Even when port A bit 7 is configured as an output, the pin still drives the input to the pulse accumulator.

PAEN — Pulse Accumulator System Enable Bit

Refer to Chapter 9 Timing Systems.

PAMOD — Pulse Accumulator Mode Bit

Refer to Chapter 9 Timing Systems.

PEDGE — Pulse Accumulator Edge Control Bit

Refer to Chapter 9 Timing Systems.

DDRA3 — Data Direction for Port A Bit 3

This bit is overridden if an output compare function is configured to control the PA3 pin.

0 = Input
1 = Output

I4/O5 — Input Capture 4/Output Compare 5 Bit

Refer to Chapter 9 Timing Systems.

RTR[1:0] — RTI Interrupt Rate Select Bits

Refer to Chapter 9 Timing Systems.

Table 9-1. Timer Summary

Control Bits PR1, PR0	XTAL Frequencies			
	4.0 MHz	8.0 MHz	12.0 MHz	Other Rates
	1.0 MHz	2.0 MHz	3.0 MHz	(E)
	1000 ns	500 ns	333 ns	(1/E)
	Main Timer Count Rates			
0 0 1 count — overflow —	1000 ns 65.536 ms	500 ns 32.768 ms	333 ns 21.845 ms	(E/1) ($E/2^{16}$)
0 1 1 count — overflow —	4.0 µs 262.14 ms	2.0 µs 131.07 ms	1.333 µs 87.381 ms	(E/4) ($E/2^{18}$)
1 0 1 count — overflow —	8.0 µs 524.29 ms	4.0 µs 262.14 ms	2.667 µs 174.76 ms	(E/8) ($E/2^{19}$)
1 1 1 count — overflow —	16.0 µs 1.049 s	8.0 µs 524.29 ms	5.333 µs 349.52 ms	(E/16) ($E/2^{20}$)

9.2 Timer Structure

Figure 9-2 shows the capture/compare system block diagram. The port A pin control block includes logic for timer functions and for general-purpose I/O. For pins PA3, PA2, PA1, and PA0, this block contains both the edge-detection logic and the control logic that enables the selection of which edge triggers an input capture. The digital level on PA[3:0] can be read at any time (read PORTA register), even if the pin is being used for the input capture function. Pins PA[6:3] are used for either general-purpose I/O, or as output compare pins. When one of these pins is being used for an output compare function, it cannot be written directly as if it were a general-purpose output. Each of the output compare functions (OC[5:2]) is related to one of the port A output pins. Output compare one (OC1) has extra control logic, allowing it optional control of any combination of the PA[7:3] pins. The PA7 pin can be used as a general-purpose I/O pin, as an input to the pulse accumulator, or as an OC1 output pin.

9.3 Input Capture

The input capture function records the time an external event occurs by latching the value of the free-running counter when a selected edge is detected at the associated timer input pin. Software can store latched values and use them to compute the periodicity and duration of events. For example, by storing the times of successive edges of an incoming signal, software can determine the period and pulse width of a signal. To measure period, two successive edges of the same polarity are captured. To measure pulse width, two alternate polarity edges are captured.

In most cases, input capture edges are asynchronous to the internal timer counter, which is clocked relative to an internal clock (PH2). These asynchronous capture requests are synchronized to PH2 so that the latching occurs on the opposite half cycle of PH2 from when the timer counter is being incremented. This synchronization process introduces a delay from when the edge occurs to when the counter value is detected. Because these delays offset each other when the time between two edges is being measured, the delay can be ignored. When an input capture is being used with an output compare, there is a similar delay between the actual compare point and when the output pin changes state.

9.5.2 Timer Interrupt Flag Register 2

Bits of this register indicate the occurrence of timer system events. 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.

Address: $1025

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	TOF	RTIF	PAOVF	PAIF				
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 9-22. 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 Interrupt Flag
The RTIF status bit is automatically set to 1 at the end of every RTI period. To clear RTIF, write a byte to TFLG2 with bit 6 set.

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

9.5.3 Pulse Accumulator Control Register

Bits RTR[1:0] of this register select the rate for the RTI system. The remaining bits control the pulse accumulator and IC4/OC5 functions.

Address: $1026

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0
Reset:	0	0	0	0	0	0	0	0

Figure 9-23. Pulse Accumulator Control Register (PACTL)

DDRA7 — Data Direction for Port A Bit 7
Refer to Chapter 6 Parallel Input/Output (I/O) Ports.

PAEN — Pulse Accumulator System Enable Bit
Refer to 9.7 Pulse Accumulator.

PAMOD — Pulse Accumulator Mode Bit
Refer to 9.7 Pulse Accumulator.

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

1. 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
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.

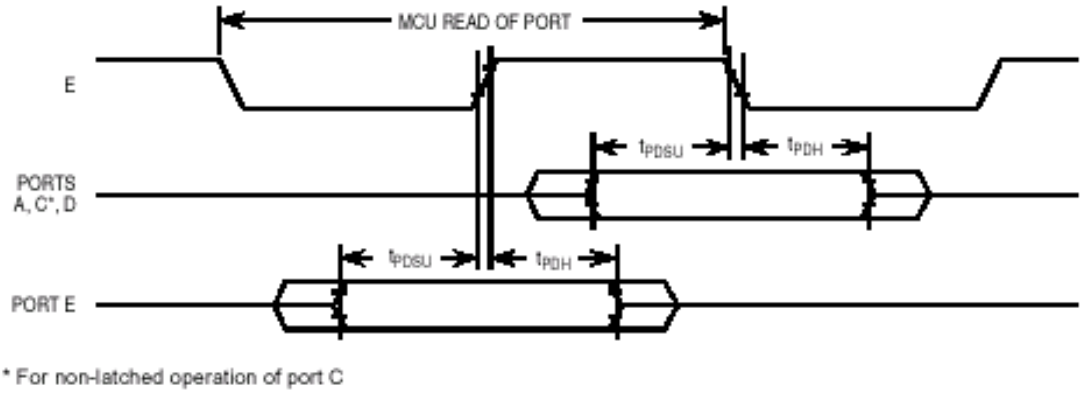


Figure 10-7. Port Read Timing Diagram

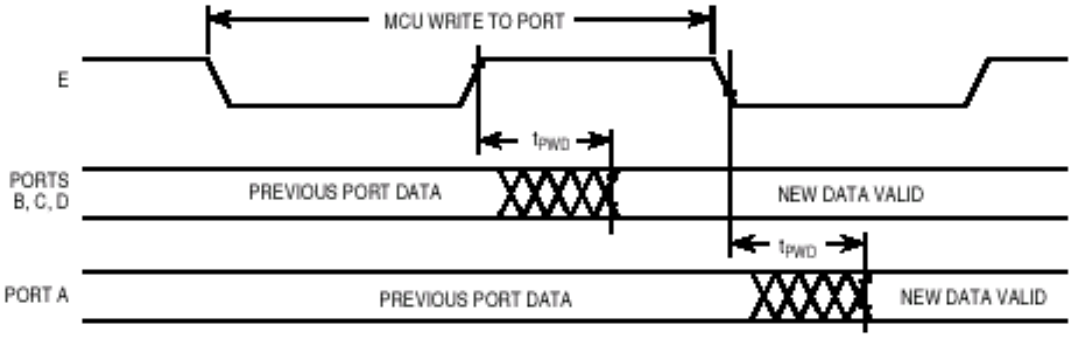


Figure 10-8. Port Write Timing Diagram

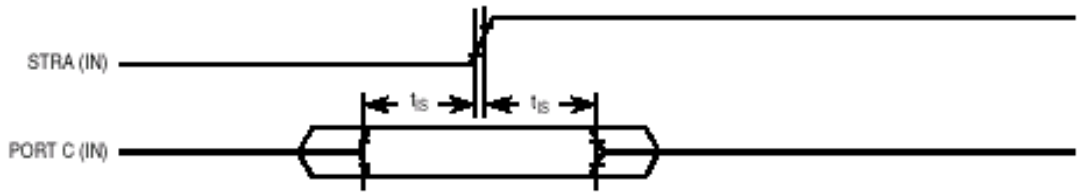


Figure 10-9. Simple Input Strobe Timing Diagram

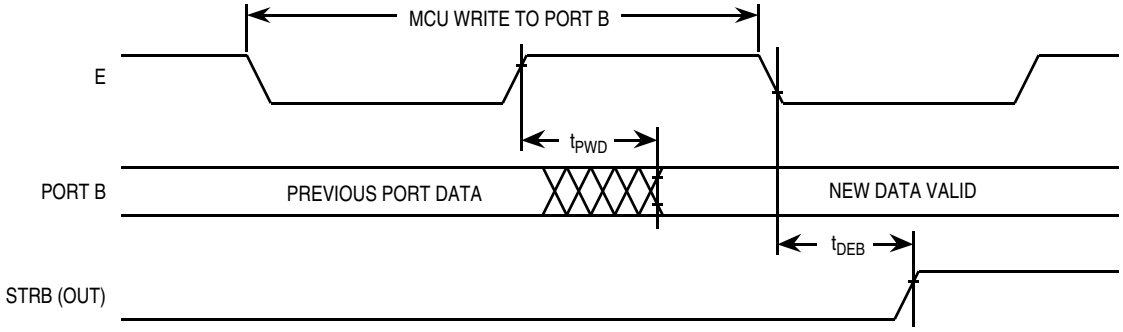
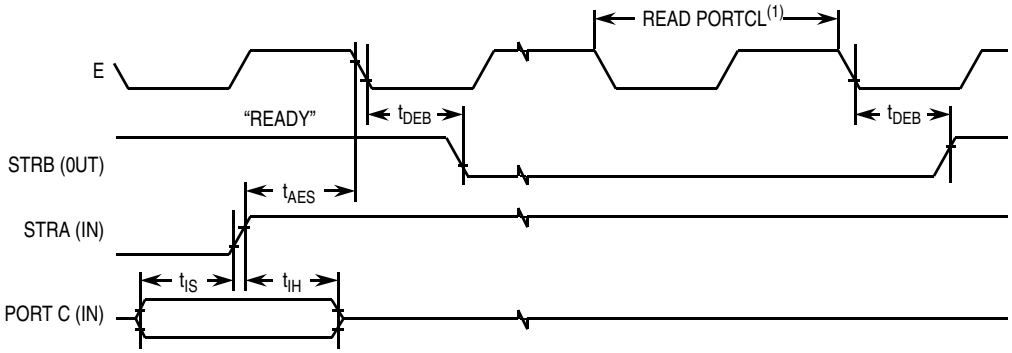


Figure 10-10. Simple Output Strobe Timing Diagram



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

Figure 10-11. Port C Input Handshake Timing Diagram

Appendix A
Development Support

A.1 Introduction

This section provides information on the development support offered for the E-series devices.

A.2 M68HC11 E-Series Development Tools

Device	Package	Emulation Module[1] [2]	Flex Cable[1] [2]	MMDS11 Target Head[1] [2]	SPGMR Programming Adapter[3]
MC68HC11E9 MC68HC711E9	52 FN	M68EM11E20	M68CBL11C	M68TC11E20FN52	M68PA11E20FN52
	52 PB	M68EM11E20	M68CBL11C	M68TC11E20PB52	M68PA11E20PB52
	56 B	M68EM11E20	M68CBL11B	M68TC11E20B56	M68PA11E20B56
	64 FU	M68EM11E20	M68CBL11C	M68TC11E20FU64	M68PA11E20FU64
MC68HC11E20 MC68HC711E20	52 FN	M68EM11E20	M68CBL11C	M68TC11E20FN52	M68PA11E20FN52
	64 FU	M68EM11E20	M68CBL11C	M68TC11E20FU64	M68PA11E20FU64
MC68HC811E2	48 P	M68EM11E20	M68CBL11B	M68TB11E20P48	M68PA11A8P48
	52 FN	M68EM11E20	M68CBL11C	M68TC11E20FN52	M68PA11E20FN52

1. Each MMDS11 system consists of a system console (M68MMDS11), an emulation module, a flex cable, and a target head.
2. A complete EVS consists of a platform board (M68HC11PFB), an emulation module, a flex cable, and a target head.
3. Each SPGMR system consists of a universal serial programmer (M68SPGMR11) and a programming adapter. It can be used alone or in conjunction with the MMDS11.

A.3 EVS — Evaluation System

The EVS is an economical tool for designing, debugging, and evaluating target systems based on the M68HC11. EVS features include:

- Monitor/debugger firmware
- One-line assembler/disassembler
- Host computer download capability
- Dual memory maps:
  - 64-Kbyte monitor map that includes 16 Kbytes of monitor EPROM
  - M68HC11 E-series user map that includes 64 Kbytes of emulation RAM
- MCU extension input/output (I/O) port for single-chip, expanded, and special-test operation modes
- RS-232C terminal and host I/O ports
- Logic analyzer connector

The duplicator program in EEPROM clears the DWOM control bit to change port D (thus, TxD) of U3 to normal driven outputs. This configuration will prevent interference due to R9 when TxD from the target MCU (U6) becomes active. Series resistor R9 demonstrates how TxD of U3 can drive RxD of U3[1] and later TxD of U6 can drive RxD of U3 without a destructive conflict between the TxD output buffers.

As the target MCU (U6) leaves reset, its mode pins select bootstrap mode so the bootloader firmware begins executing. A break is sent out the TxD pin of U6. At this time, the TxD pin of U3 is at a driven high so R9 acts as a pullup resistor for TxD of the target MCU (U6). The break character sent from U6 is received by U3 so the duplicator program that is running in the EEPROM of the master MCU knows that the target MCU is ready to accept a bootloaded program.

The master MCU sends a leading $FF character to set the baud rate in the target MCU. Next, the master MCU passes a 3-instruction program to the target MCU and pauses so the bootstrap program in the target MCU will stop the loading process and jump to the start of the downloaded program. This sequence demonstrates the variable-length download feature of the MC68HC711E9 bootloader.

The short program downloaded to the target MCU clears the DWOM bit to change its TxD pin to a normal driven CMOS output and jumps to the EPROM programming utility in the bootstrap ROM of the target MCU.

Note that the small downloaded program did not have to set up the SCI or initialize any parameters for the EPROM programming process. The bootstrap software that ran prior to the loaded program left the SCI turned on and configured in a way that was compatible with the SCI in the master MCU (the duplicator program in the master MCU also did not have to set up the SCI for the same reason). The programming time and starting address for EPROM programming in the target MCU were also set to default values by the bootloader software before jumping to the start of the downloaded program.

Before the EPROM in the target MCU can be programmed, the V_{PP} power supply must be available at the $\overline{\text{XIRQ}}/V_{PPE}$ pin of the target MCU. The duplicator program running in the master MCU monitors this voltage (for presence or absence, not level) at PE7 through resistor divider R14–RI5. The PE7 input was chosen because the internal circuitry for port E pins can tolerate voltages slightly higher than V_{DD}; therefore, resistors R14 and R15 are less critical. No data to be programmed is passed to the target MCU until the master MCU senses that V_{PP} has been stable for about 200 ms.

When V_{PP} is ready, the master MCU turns on the red LED (light-emitting diode) and begins passing data to the target MCU. EPROM Programming Utility explains the activity as data is sent from the master MCU to the target MCU and programmed into the EPROM of the target. The master MCU in the EVBU corresponds to the HOST in the programming utility description and the "PROGRAM utility in MCU" is running in the bootstrap ROM of the target MCU.

Each byte of data sent to the target is programmed and then the programmed location is read and sent back to the master for verification. If any byte fails, the red and green LEDs are turned off, and the programming operation is aborted. If the entire 12 Kbytes are programmed and verified successfully, the red LED is turned off, and the green LED is turned on to indicate success. The programming of all 12 Kbytes takes about 30 seconds.

After a programming operation, the V_{PP} switch (S2) should be turned off before the EVBU power is turned off.

Listing 3. MC68HC711E9 Bootloader ROM

```
213
214                 ***************************************************
215                 * Boot ROM revision level in ASCII
216                 *       (ORG     $BFD1)
217 BFD1 41                FCC     "A"
218                 ***************************************************
219                 * Mask set I.D. ($0000 FOR EPROM PARTS)
220                 *       (ORG     $BFD2)
221 BFD2 0000                FDB     $0000
222                 ***************************************************
223                 * '711E9 I.D. - Can be used to determine MCU type
224                 *       (ORG     $BFD4)
225 BFD4 71E9                FDB     $71E9
226
227                 ***************************************************
228                 * VECTORS - point to RAM for pseudo-vector JUMPs
229
230 BFD6 00C4                 FDB     $100-60         SCI
231 BFD8 00C7                 FDB     $100-57         SPI
232 BFDA 00CA                 FDB     $100-54         PULSE ACCUM INPUT EDGE
233 BFDC 00CD                 FDB     $100-51         PULSE ACCUM OVERFLOW
234 BFDE 00D0                 FDB     $100-48         TIMER OVERFLOW
235 BFE0 00D3                 FDB     $100-45         TIMER OUTPUT COMPARE 5
236 BFE2 00D6                 FDB     $100-42         TIMER OUTPUT COMPARE 4
237 BFE4 00D9                 FDB     $100-39         TIMER OUTPUT COMPARE 3
238 BFE6 00DC                 FDB     $100-36         TIMER OUTPUT COMPARE 2
239 BFE8 00DF                 FDB     $100-33         TIMER OUTPUT COMPARE 1
240 BFEA 00E2                 FDB     $100-30         TIMER INPUT CAPTURE 3
241 BFEC 00E5                 FDB     $100-27         TIMER INPUT CAPTURE 2
242 BFEE 00E8                 FDB     $100-24         TIMER INPUT CAPTURE 1
243 BFF0 00EB                 FDB     $100-21         REAL TIME INT
244 BFF2 00EE                 FDB     $100-18         IRQ
245 BFF4 00F1                 FDB     $100-15         XIRQ
246 BFF6 00F4                 FDB     $100-12         SWI
247 BFF8 00F7                 FDB     $100-9          ILLEGAL OP-CODE
248 BFFA 00FA                 FDB     $100-6          COP FAIL
249 BFFC 00FD                 FDB     $100-3          CLOCK MONITOR
250 BFFE BF54                 FDB     BEGIN           RESET
251 C000                      END
Symbol Table:

Symbol Name     Value  Def.#  Line Number Cross Reference

BAUD             002B *00037  00160 00180
BAUDOK           BF8A *00183  00178
BEGIN            BF54 *00155  00250
DELAYF           021B *00061  00163
DELAYS           0DB0 *00060  00181
DONEIT           BF47 *00142  00124
EEPMEND          B7FF *00050
EEPMSTR          B600 *00049  00175
ELAT             0020 *00043  00125 00128
EPGM             0001 *00044  00128
EPRMEND          FFFF *00053
EPRMSTR          D000 *00052  00206
```

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.