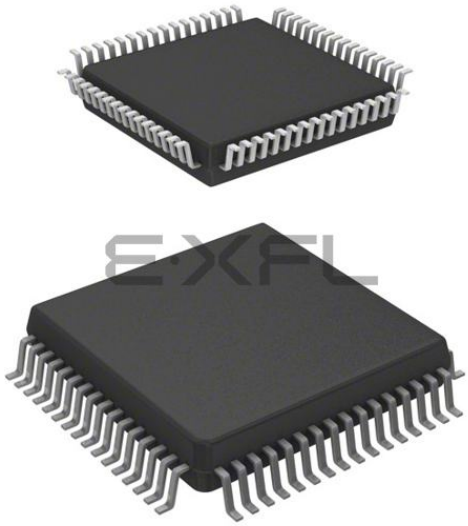


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	HC11
Core Size	8-Bit
Speed	4MHz
Connectivity	SCI, SPI
Peripherals	POR, WDT
Number of I/O	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	64-QFP
Supplier Device Package	64-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68711e20cfue4

Chapter 2
Operating Modes and On-Chip Memory

2.1 Introduction

This section contains information about the operating modes and the on-chip memory for M68HC11 E-series MCUs. Except for a few minor differences, operation is identical for all devices in the E series. Differences are noted where necessary.

2.2 Operating Modes

The values of the mode select inputs MODB and MODA during reset determine the operating mode. Single-chip and expanded multiplexed are the normal modes.

- In single-chip mode only on-chip memory is available.
- Expanded mode, however, allows access to external memory.

Each of the two normal modes is paired with a special mode:

- Bootstrap, a variation of the single-chip mode, is a special mode that executes a bootloader program in an internal bootstrap ROM.
- Test is a special mode that allows privileged access to internal resources.

2.2.1 Single-Chip Mode

In single-chip mode, ports B and C and strobe pins A (STRA) and B (STRB) are available for general-purpose parallel input/output (I/O). In this mode, all software needed to control the MCU is contained in internal resources. If present, read-only memory (ROM) and/or erasable, programmable read-only memory (EPROM) will always be enabled out of reset, ensuring that the reset and interrupt vectors will be available at locations $FFC0–$FFFF.

NOTE

For the MC68HC811E2, the vector locations are the same; however, they are contained in the 2048-byte EEPROM array.

2.2.2 Expanded Mode

In expanded operating mode, the MCU can access the full 64-Kbyte address space. The space includes:

- The same on-chip memory addresses used for single-chip mode
- Addresses for external peripherals and memory devices

The expansion bus is made up of ports B and C, and control signals AS (address strobe) and R/$\overline{W}$ (read/write). R/$\overline{W}$ and AS allow the low-order address and the 8-bit data bus to be multiplexed on the same pins. During the first half of each bus cycle address information is present. During the second half of each bus cycle the pins become the bidirectional data bus. AS is an active-high latch enable signal for an external address latch. Address information is allowed through the transparent latch while AS is high and is latched when AS drives low.

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.

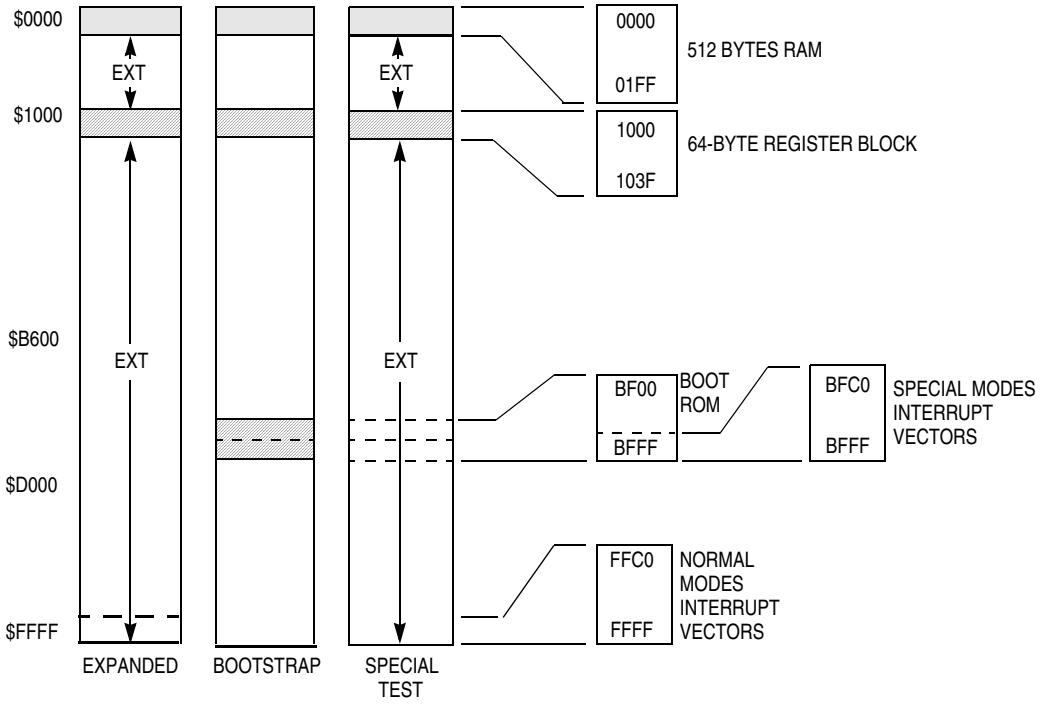


Figure 2-2. Memory Map for MC68HC11E0

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)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$103E	Reserved		R	R	R	R	R	R	R	R
$103F	System Configuration Register (CONFIG) See page 43.	Read:					NOSEC	NOCOP	ROMON	EEON
		Write:								
		Reset:	0	0	0	0	U	U	1	U
$103F	System Configuration Register (CONFIG)(3) See page 43.	Read:	EE3	EE2	EE1	EE0	NOSEC	NOCOP		EEON
		Write:								
		Reset:	1	1	1	1	U	U	1	1

1. Can be written only once in first 64 cycles out of reset in normal modes or at any time during special modes.
2. MC68HC711E9 only
3. MC68HC811E2 only

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

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

2.3.1 RAM and Input/Output Mapping

Hardware priority is built into RAM and I/O mapping. Registers have priority over RAM and RAM has priority over ROM. When a lower priority resource is mapped at the same location as a higher priority resource, a read/write of a location results in a read/write of the higher priority resource only. For example, if both the register block and the RAM are mapped to the same location, only the register block will be accessed. If RAM and ROM are located at the same position, RAM has priority.

The fully static RAM can be used to store instructions, variables, and temporary data. The direct addressing mode can access RAM locations using a 1-byte address operand, saving program memory space and execution time, depending on the application.

RAM contents can be preserved during periods of processor inactivity by two methods, both of which reduce power consumption. They are:

1. In the software-based stop mode, the clocks are stopped while V_{DD} powers the MCU. Because power supply current is directly related to operating frequency in CMOS integrated circuits, only a very small amount of leakage exists when the clocks are stopped.
2. In the second method, the MODB/V_{STBY} pin can supply RAM power from a battery backup or from a second power supply. Figure 2-8 shows a typical standby voltage circuit for a standard 5-volt device. Adjustments to the circuit must be made for devices that operate at lower voltages. Using the MODB/V_{STBY} pin may require external hardware, but can be justified when a significant amount of external circuitry is operating from V_{DD}. If V_{STBY} is used to maintain RAM contents, reset must be held low whenever V_{DD} is below normal operating level. Refer to Chapter 5 Resets and Interrupts.

Address: $1036

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	MBE		ELAT	EXCOL	EXROW	T1	T0	PGM
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

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

3.4 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.

5.2.6 Configuration Control Register

Address:	$103F							
	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	EE3	EE2	EE1	EE0	NOSEC	NOCOP	ROMON	EEON
Reset:	0	0	0	0	1	1	1	1

Figure 5-3. Configuration Control Register (CONFIG)

EE[3:0] — EEPROM Mapping Bits
EE[3:0] apply only to MC68HC811E2. Refer to Chapter 2 Operating Modes and On-Chip Memory.

NOSEC — Security Mode Disable Bit
Refer to Chapter 2 Operating Modes and On-Chip Memory.

NOCOP — COP System Disable Bit
0 = COP enabled (forces reset on timeout)
1 = COP disabled (does not force reset on timeout)

ROMON — ROM (EPROM) Enable Bit
Refer to Chapter 2 Operating Modes and On-Chip Memory.

EEON — EEPROM Enable Bit
Refer to Chapter 2 Operating Modes and On-Chip Memory.

5.3 Effects of Reset

When a reset condition is recognized, the internal registers and control bits are forced to an initial state. Depending on the cause of the reset and the operating mode, the reset vector can be fetched from any of six possible locations. Refer to Table 5-2.

Table 5-2. Reset Cause, Reset Vector, and Operating Mode

Cause of Reset	Normal Mode Vector	Special Test or Bootstrap
POR or $\overline{\text{RESET}}$ pin	$FFFE, FFFF	$BFFE, $BFFF
Clock monitor failure	$FFFC, FFFD	$BFFC, $BFFD
COP Watchdog Timeout	$FFFA, FFFB	$BFFA, $BFFB

These initial states then control on-chip peripheral systems to force them to known startup states, as described in the following subsections.

5.3.1 Central Processor Unit (CPU)

After reset, the central processor unit (CPU) fetches the restart vector from the appropriate address during the first three cycles and begins executing instructions. The stack pointer and other CPU registers are indeterminate immediately after reset; however, the X and I interrupt mask bits in the condition code register (CCR) are set to mask any interrupt requests. Also, the S bit in the CCR is set to inhibit stop mode.

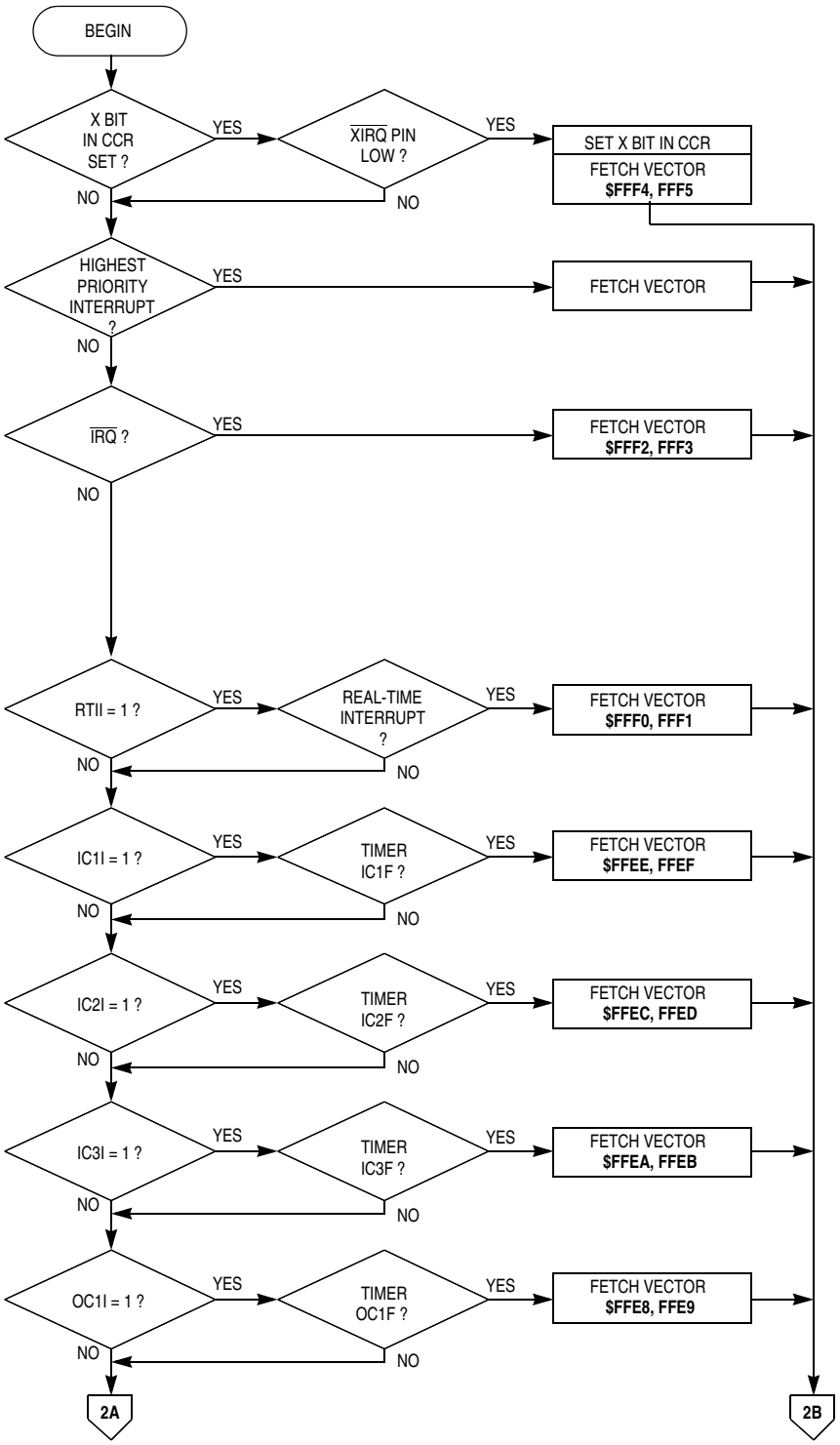


Figure 5-6. Interrupt Priority Resolution (Sheet 1 of 2)

7.4 Receive Operation

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

7.5 Wakeup Feature

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

Two methods of wakeup are available:

- Idle-line wakeup
- Address-mark wakeup

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

7.5.1 Idle-Line Wakeup

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

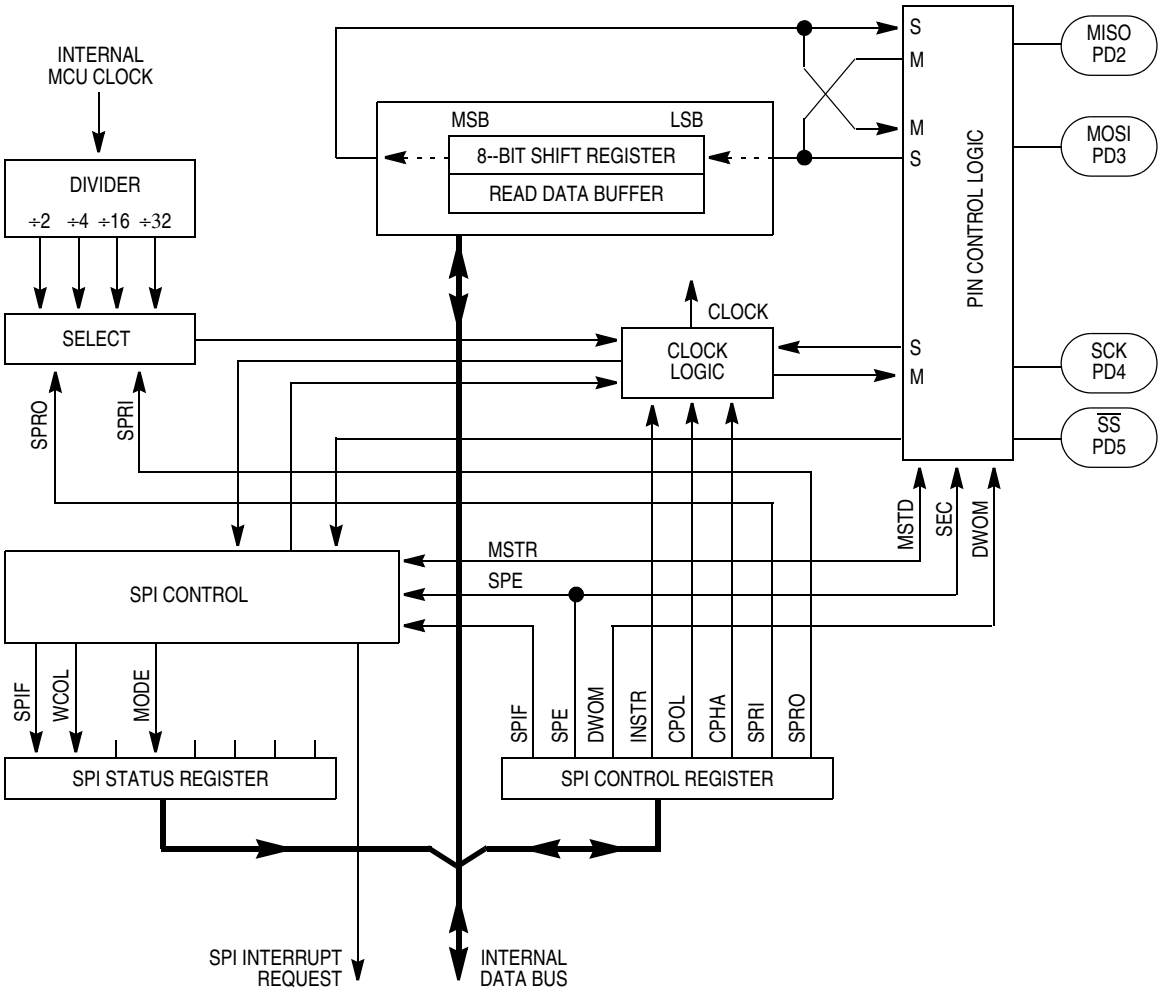


Figure 8-1. SPI Block Diagram

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.

FOC[1:5] — Force Output Comparison Bit

When the FOC bit associated with an output compare circuit is set, the output compare circuit immediately performs the action it is programmed to do when an output match occurs.

0 = Not affected
1 = Output x action occurs

Bits [2:0] — Unimplemented

Always read 0

9.4.3 Output Compare Mask Register

Use OC1M with OC1 to specify the bits of port A that are affected by a successful OC1 compare. The bits of the OC1M register correspond to PA[7:3].

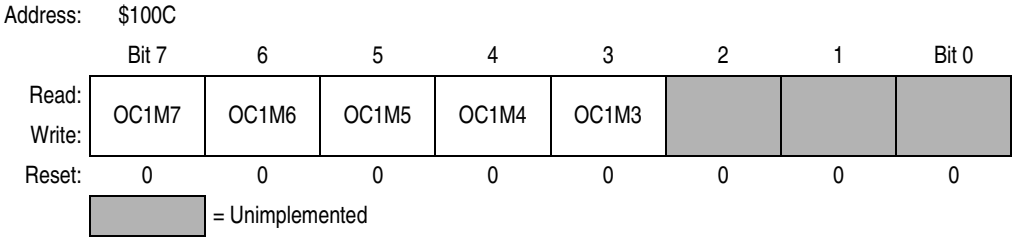


Figure 9-13. Output Compare 1 Mask Register (OC1M)

OC1M[7:3] — Output Compare Masks

0 = OC1 disabled
1 = OC1 enabled to control the corresponding pin of port A

Bits [2:0] — Unimplemented

Always read 0

9.4.4 Output Compare Data Register

Use this register with OC1 to specify the data that is to be stored on the affected pin of port A after a successful OC1 compare. When a successful OC1 compare occurs, a data bit in OC1D is stored in the corresponding bit of port A for each bit that is set in OC1M.

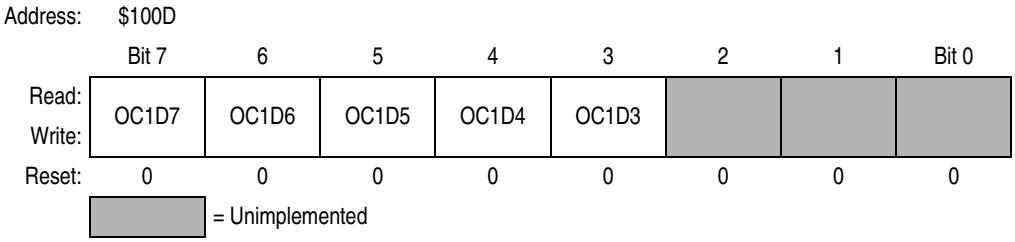


Figure 9-14. Output Compare 1 Data Register (OC1D)

If OC1Mx is set, data in OC1Dx is output to port A bit x on successful OC1 compares.

Bits [2:0] — Unimplemented

Always read 0

10.9 Control Timing

Characteristic[1] [2]	Symbol	1.0 MHz		2.0 MHz		3.0 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of operation	f_o	dc	1.0	dc	2.0	dc	3.0	MHz
E-clock period	t_{CYC}	1000	—	500	—	333	—	ns
Crystal frequency	f_{XTAL}	—	4.0	—	8.0	—	12.0	MHz
External oscillator frequency	4 f_o	dc	4.0	dc	8.0	dc	12.0	MHz
Processor control setup time t_{PCSU} = 1/4 t_{CYC}+ 50 ns	t_{PCSU}	300	—	175	—	133	—	ns
Reset input pulse width To guarantee external reset vector Minimum input time (can be pre-empted by internal reset)	PW_{RSTL}	8 1	— —	8 1	— —	8 1	— —	t_{CYC}
Mode programming setup time	t_{MPS}	2	—	2	—	2	—	t_{CYC}
Mode programming hold time	t_{MPH}	10	—	10	—	10	—	ns
Interrupt pulse width, $\overline{IRQ}$ edge-sensitive mode PW_{IRQ} = t_{CYC} + 20 ns	PW_{IRQ}	1020	—	520	—	353	—	ns
Wait recovery startup time	t_{WRS}	—	4	—	4	—	4	t_{CYC}
Timer pulse width input capture pulse accumulator input PW_{TIM} = t_{CYC} + 20 ns	PW_{TIM}	1020	—	520	—	353	—	ns

1. V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, T_A = T_L to T_H, all timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted
2. $\overline{RESET}$ is recognized during the first clock cycle it is held low. Internal circuitry then drives the pin low for four clock cycles, releases the pin, and samples the pin level two cycles later to determine the source of the interrupt. Refer to Chapter 5 Resets and Interrupts for further detail.

10.13 Analog-to-Digital Converter Characteristics

Characteristic[1]	Parameter[2]	Min	Absolute	2.0 MHz Max	3.0 MHz Max	Unit
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%.

Description	Temperature	Frequency	MC Order Number
20 Kbytes custom ROM	0°C to +70°°C	3 MHz	MC68HC11E20FN3
	–40°C to +85°C	2 MHz	MC68HC11E20CFN2
		3 MHz	MC68HC11E20CFN3
	–40°C to +105°C	2 MHz	MC68HC11E20VFN2
	–40°C to +125°C	2 MHz	MC68HC11E20MFN2

64-pin quad flat pack (QFP)

Custom ROM	0°C to +70°°C	3 MHz	MC68HC11E9FU3
	–40°C to +85°C	2 MHz	MC68HC11E9CFU2
		3 MHz	MC68HC11E9CFU3
	–40°C to +105°C	2 MHz	MC68HC11E9VFU2
	–40°C to +125°C	2 MHz	MC68HC11E9MFU2

64-pin quad flat pack (continued)

20 Kbytes Custom ROM	0°C to +70°°C	3 MHz	MC68HC11E20FU3
	–40°C to +85°C	2 MHz	MC68HC11E20CFU2
		3 MHz	MC68HC11E20CFU3
	–40°C to +105°C	2 MHz	MC68HC11E20VFU2
	–40°C to +125°C	2 MHz	MC68HC11E20MFU2

52-pin thin quad flat pack (10 mm x 10 mm)

Custom ROM	0°C to +70°°C	3 MHz	MC68HC11E9PB3
	–40°C to +85°C	2 MHz	MC68HC11E9CPB2
		3 MHz	MC68HC11E9CPB3
	–40°C to +105°C	2 MHz	MC68HC11E9VPB2
	–40°C to +125°C	2 MHz	MC68HC11E9MPB2

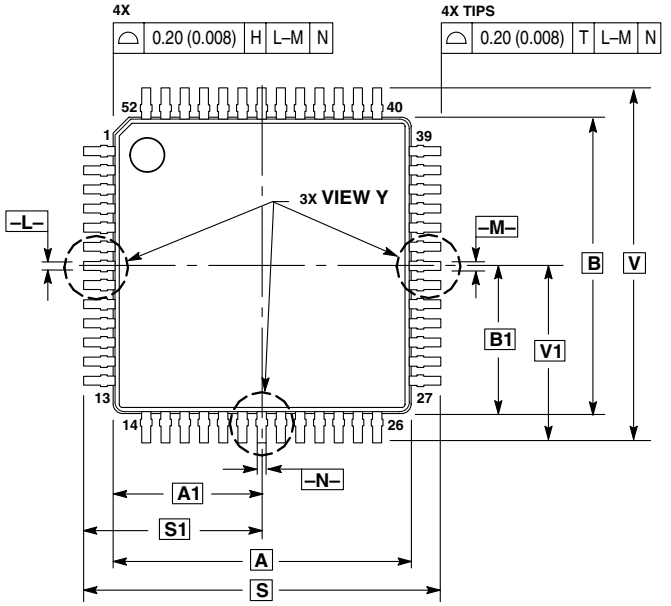
56-pin dual in-line package with 0.70-inch lead spacing (SDIP)

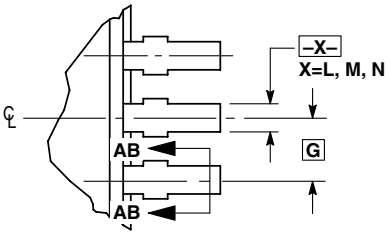
Custom ROM	0°C to +70°°C	3 MHz	MC68HC11E9B3
	–40°C to +85°C	2 MHz	MC68HC11E9CB2
		3 MHz	MC68HC11E9CB3
	–40°C to +105°C	2 MHz	MC68HC11E9VB2
	–40°C to +125°C	2 MHz	MC68HC11E9MB2

11.4 Extended Voltage Device Ordering Information (3.0 Vdc to 5.5 Vdc)

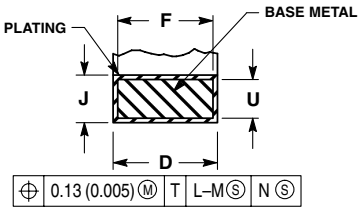
Description	Temperature	Frequency	MC Order Number
52-pin plastic leaded chip carrier (PLCC)			
Custom ROM	–20°C to +70°C	2 MHz	MC68L11E9FN2 MC68L11E20FN2
No ROM		2 MHz	MC68L11E1FN2
No ROM, no EEPROM		2 MHz	MC68L11E0FN2
64-pin quad flat pack (QFP)			
Custom ROM	–20°C to +70°C	2 MHz	MC68L11E9FU2 MC68L11E20FU2
No ROM		2 MHz	MC68L11E1FU2
No ROM, no EEPROM		2 MHz	MC68L11E0FU2
52-pin thin quad flat pack (10 mm x 10 mm)			
Custom ROM	–20°C to +70°C	2 MHz	MC68L11E9PB2
No ROM		2 MHz	MC68L11E1PB2
No ROM, no EEPROM		2 MHz	MC68L11E0PB2
56-pin dual in-line package with 0.70-inch lead spacing (SDIP)			
Custom ROM	–20°C to +70°C	2 MHz	MC68L11E9B2
No ROM		2 MHz	MC68L11E1B2
No ROM, no EEPROM		2 MHz	MC68L11E0B2

11.8 52-Pin Thin Quad Flat Pack (Case 848D)

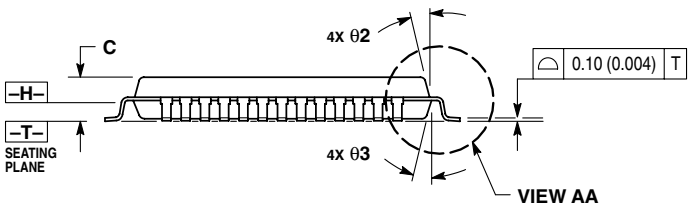


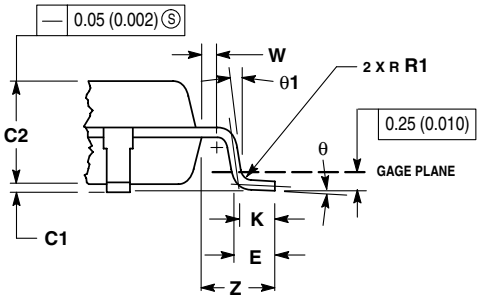


VIEW Y



SECTION AB–AB
ROTATED 90° CLOCKWISE





VIEW AA

NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM PLANE –H– IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS –L–, –M– AND –N– TO BE DETERMINED AT DATUM PLANE –H–.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE –T–.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE -H-.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED 0.46 (0.018). MINIMUM SPACE BETWEEN PROTRUSION AND ADJACENT LEAD OR PROTRUSION 0.07 (0.003).

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	10.00 BSC		0.394 BSC	
A1	5.00 BSC		0.197 BSC	
B	10.00 BSC		0.394 BSC	
B1	5.00 BSC		0.197 BSC	
C	—	1.70	—	0.067
C1	0.05	0.20	0.002	0.008
C2	1.30	1.50	0.051	0.059
D	0.20	0.40	0.008	0.016
E	0.45	0.75	0.018	0.030
F	0.22	0.35	0.009	0.014
G	0.65 BSC		0.026 BSC	
J	0.07	0.20	0.003	0.008
K	0.50 REF		0.020 REF	
R1	0.08	0.20	0.003	0.008
S	12.00 BSC		0.472 BSC	
S1	6.00 BSC		0.236 BSC	
U	0.09	0.16	0.004	0.006
V	12.00 BSC		0.472 BSC	
V1	6.00 BSC		0.236 BSC	
W	0.20 REF		0.008 REF	
Z	1.00 REF		0.039 REF	
θ	0°	7°	0°	7°
θ1	0°	—	0°	—
θ2	12° REF		12° REF	
θ3	5°	13°	5°	13°

The alternate vector locations are achieved by simply driving address bit A14 low during all vector fetches if SMOD = 1. For special test mode, the alternate vector locations assure that the reset vector can be fetched from external memory space so the test system can control MCU operation. In special bootstrap mode, the small boot ROM is enabled in the memory map by RBOOT = 1 so the reset vector will be fetched from this ROM and the bootloader firmware will control MCU operation.

RBOOT is reset to 1 in bootstrap mode to enable the small boot ROM. In the other three modes, RBOOT is reset to 0 to keep the boot ROM out of the memory map. While in special test mode, SMOD = 1, which allows the RBOOT control bit to be written to 1 by software to enable the boot ROM for testing purposes.

Boot ROM Firmware

The main program in the boot ROM is the bootloader, which is automatically executed as a result of resetting the MCU in bootstrap mode. Some newer versions of the M68HC11 Family have additional utility programs that can be called from a downloaded program. One utility is available to program EPROM or OTP versions of the M68HC11. A second utility allows the contents of memory locations to be uploaded to a host computer. In the MC68HC711K4 boot ROM, a section of code is used by Freescale for stress testing the on-chip EEPROM. These test and utility programs are similar to self-test ROM programs in other MCUs except that the boot ROM does not use valuable space in the normal memory map.

Bootstrap firmware is also involved in an optional EEPROM security function on some versions of the M68HC11. This EEPROM security feature prevents a software pirate from seeing what is in the on-chip EEPROM. The secured state is invoked by programming the no security (NOSEC) EEPROM bit in the CONFIG register. Once this NOSEC bit is programmed to 0, the MCU will ignore the mode A pin and always come out of reset in normal single-chip mode or special bootstrap mode, depending on the state of the mode B pin. Normal single-chip mode is the usual way a secured part would be used. Special bootstrap mode is used to disengage the security function (only after the contents of EEPROM and RAM have been erased). Refer to the *M68HC11 Reference Manual*, Freescale document order number M68HC11RM/AD, for additional information on the security mode and complete listings of the boot ROMs that support the EEPROM security functions.

Automatic Selection of Baud Rate

The bootloader program in the MC68HC711E9 accommodates either of two baud rates.

- The higher of these baud rates (7812 baud at a 2-MHz E-clock rate) is used in systems that operate from a binary frequency crystal such as 2^{23} Hz (8.389 MHz). At this crystal frequency, the baud rate is 8192 baud, which was used extensively in automotive applications.
- The second baud rate available to the M68HC11 bootloader is 1200 baud at a 2-MHz E-clock rate. Some of the newest versions of the M68HC11, including the MC68HC11F1 and MC68HC117K4, accommodate other baud rates using the same differentiation technique explained here. Refer to the reference numbers in square brackets in Figure 2 during the following explanation.

NOTE

Software can change some aspects of the memory map after reset.

Listing 1. MCU-to-MCU Duplicator Program

```
 82 B666
 83 B666 150482   DUNPRG     BCLR   PORTB (RESET+RED) Red OFF, apply reset
 84 B669 20FE                BRA    *                 Done so just hang
 85 B66B
 86               ************************************************
 87               * Subroutine to get & send an SCI char.  Also
 88               *  advances pointer (X).
 89               ************************************************
 90 B66B A600     SEND1      LDAA   0,X               Get a character
 91 B66D 132E80FC TRDYLP     BRCLR  SCSR TDRE TRDYLP Wait for TDRE
 92 B671 972F                STAA   SCDR              Send character
 93 B673 08                  INX                     Advance pointer
 94 B674 39                  RTS                     ** Return **
 95
 96               ************************************************
 97               * Program to be bootloaded to target '711E9
 98               ************************************************
 99 B675 8604     BLPROG     LDAA   #$04              Pattern for DWOM off, no SPI
100 B677 B71028              STAA   $1028             Turns off DWOM in target MCU
101               * NOTE: Can't use direct addressing in target MCU because
102               *       regs are located at $1000.
103 B67A 7EBF00              JMP    PROGRAM           Jumps to EPROM prog routine
104 B67D          ENDBPR     EQU    *

Symbol Table:
Symbol Name     Value  Def.#  Line Number Cross Reference

BEGIN            B600 *00029
BLLOOP           B616 *00038  00040
BLPROG           B675 *00099  00037
DATALP           B648 *00068  00079
DLYLP            B620 *00046  00047
DLYLP2           B637 *00059  00063
DUNPRG           B666 *00083  00076
ENDBPR           B67D *00104  00039
EPSTRT           D000 *00023  00055 00066
GREEN            0001 *00015  00075 00081
INIT             103D *00009  00029
PORTB            0004 *00011  00033 00058 00061 00075 00081 00083
PORTE            000A *00016  00059
PROGRAM          BF00 *00022  00103
RDRF             0020 *00020  00034 00053 00071
RED              0002 *00014  00058 00061 00075 00083
RESET            0080 *00013  00032 00083
SCDR             002F *00021  00036 00049 00054 00072 00092
SCSR             002E *00017  00034 00048 00053 00071 00091
SEND1            B66B *00090  00038 00067 00070
SPCR             0028 *00010  00031
TDRE             0080 *00019  00091
TRDYLP           B66D *00091  00091
VERF             B64F *00071  00069 00071
VERFOK           B65F *00078  00074
WT4BRK           B60B *00034  00034
WT4FF            B627 *00053  00053
WT4VPP           B630 *00057  00060
```

Listing 3. MC68HC711E9 Bootloader ROM

```
107                * This routine uses 2 bytes of stack space
108                * Routine does not return. Reset to exit.
109                ************************************************
110 BF13           PRGROUT  EQU    *
111 BF13 3C                PSHX                    Save program delay constant
112 BF14 CE1000            LDX    #$1000           Point to internal registers
113 BF17
114                * Send $FF to indicate ready for program data
115
116 BF17 1F2E80FC          BRCLR  SCSR,X $80 *     Wait for TDRE
117 BF1B 86FF              LDAA   #$FF
118 BF1D A72F              STAA   SCDAT,X
119
120 BF1F           WAIT1    EQU    *
121 BF1F 1F2E20FC          BRCLR  SCSR,X $20 *     Wait for RDRF
122 BF23 E62F              LDAB   SCDAT,X          Get received byte
123 BF25 18E100            CMPB   $0,Y             See if already programmed
124 BF28 271D              BEQ    DONEIT           If so, skip prog cycle
125 BF2A 8620              LDAA   #ELAT            Put EPROM in prog mode
126 BF2C A73B              STAA   PPROG,X
127 BF2E 18E700            STAB   0,Y              Write the data
128 BF31 8621              LDAA   #ELAT+EPGM
129 BF33 A73B              STAA   PPROG,X          Turn on prog voltage
130 BF35 32                PULA                    Pull delay constant
131 BF36 33                PULB                    into D-reg
132 BF37 37                PSHB                    But also keep delay
133 BF38 36                PSHA                    keep delay on stack
134 BF39 E30E              ADDD   TCNT,X           Delay const + present TCNT
135 BF3B ED16              STD    TOC1,X           Schedule OC1 (2ms delay)
136 BF3D 8680              LDAA   #OC1F
137 BF3F A723              STAA   TFLG1,X          Clear any previous flag
138
139 BF41 1F2380FC          BRCLR  TFLG1,X OC1F * Wait for delay to expire
140 BF45 6F3B              CLR    PPROG,X          Turn off prog voltage
141                *
142 BF47           DONEIT   EQU    *
143 BF47 1F2E80FC          BRCLR  SCSR,X $80 *     Wait for TDRE
144 BF4B 18A600            LDAA   $0,Y             Read from EPROM and...
145 BF4E A72F              STAA   SCDAT,X          Xmit for verify
146 BF50 1808              INY                     Point at next location
147 BF52 20CB              BRA    WAIT1            Back to top for next
148                * Loops indefinitely as long as more data sent.
149
150                ************************************************
151                * Main bootloader starts here
152                ************************************************
153                * RESET vector points to here
154
155 BF54           BEGIN    EQU    *
156 BF54 8E01FF            LDS    #RAMEND          Initialize stack pntr
157 BF57 CE1000            LDX    #$1000           Point at internal regs
158 BF5A 1C2820            BSET   SPCR,X $20       Select port D wire-OR mode
159 BF5D CCA20C            LDD    #$A20C           BAUD in A, SCCR2 in B
160 BF60 A72B              STAA   BAUD,X           SCPx = ÷4, SCRx = ÷4
161                * Writing 1 to MSB of BAUD resets count chain
```

Programming Procedure

Once you have obtained PCbug11, use this step-by-step procedure to program your MC68HC711E9 part.

Step 1

- Before applying power to the EVBU, remove the jumper from J7 and place it across J3 to ground the MODB pin.
- Place a jumper across J4 to ground the MODA pin. This will force the EVBU into special bootstrap mode on power up.
- Remove the resident MC68HC11E9 MCU from the EVBU.
- Place your MC68HC711E9 in the open socket with the notched corner of the part aligned with the notch on the PLCC socket.
- Connect the EVBU to one of your PC COM ports. Apply +5 volts to V_{DD} and ground to GND on the power connector of your EVBU.

Also take note of P4 connector pin 18. In step 5, you will connect a +12-volt (at most +12.5 volts) programming voltage through a 100-Ω current limiting resistor to the XIRQ pin. Do not connect this programming voltage until you are instructed to do so in step 5.

Step 2

- From a DOS command line prompt, start PCbug11 with
  - C:\PCBUG11\> PCBUG11 –E PORT = 1 with the EVBU connected to COM1
  - C:\PCBUG11\> PCBUG11 –E PORT = 2 with the EVBU connected to COM2

PCbug11 only supports COM ports 1 and 2. If you have made the proper connections and have a high quality cable, you should quickly get a PCbug11 command prompt. If you do receive a Comms fault error, check your cable and board connections. Most PCbug11 communications problems can be traced to poorly made cables or bad board connections.

Step 3

- PCbug11 defaults to base 10 for its input parameters; change this to hexadecimal by typing

CONTROL BASE HEX

Step 4

- You must declare the addresses of the EPROM array to PCbug11. To do this, type:
  EPROM D000 FFFF

Step 5

You are now ready to download your program into the EPROM.

- Connect +12 volts (at most +12.5 volts) through a 100-Ω current limiting resistor to P4 connector pin 18, the XIRQ* pin.
- At the PCbug11 command prompt type: LOADS C:\MYPROG\ISHERE.S19

Substitute the name of your program into the command above. Use a full path name if your program is not located in the same directory as PCbug11.