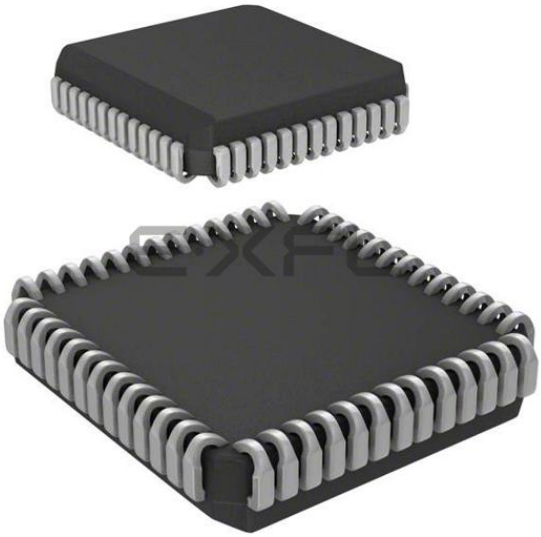


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	HC11
Core Size	8-Bit
Speed	2MHz
Connectivity	SCI, SPI
Peripherals	POR, WDT
Number of I/O	38
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	52-LCC (J-Lead)
Supplier Device Package	52-PLCC (19.1x19.1)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc11e0cfne2

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	

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 µs 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).

Address: $1039

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	ADPU	CSEL	IRQE(1)	DLY(1)	CME		CR1(1)	CR0(1)
Reset:	0	0	0	1	0	0	0	0

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

= Unimplemented

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.

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.

PORTCL is used in the handshake clearing mechanism. When an active edge occurs on the STRA pin, port C data is latched into the PORTCL register. Reads of this register return the last value latched into PORTCL and clear STAF flag (following a read of PIOC with STAF set).

Address: $1007

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
Reset:	0	0	0	0	0	0	0	0

Figure 6-6. Port C Data Direction Register (DDRC)

DDRC[7:0] — Port C Data Direction Bits

In the 3-state variation of output handshake mode, clear the corresponding DDRC bits. Refer to Figure 10-13. 3-State Variation of Output Handshake Timing Diagram (STRA Enables Output Buffer).

0 = Input
1 = Output

6.5 Port D

In all modes, port D bits [5:0] can be used either for general-purpose I/O or with the serial communications interface (SCI) and serial peripheral interface (SPI) subsystems. During reset, port D pins PD[5:0] are configured as high-impedance inputs (DDRD bits cleared).

Address: $1008

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	0	0	PD5	PD4	PD3	PD2	PD1	PD0
Reset:	—	—	I	I	I	I	I	I
Alternate Function:	—	—	PD5 $\overline{SS}$	PD4 SCK	PD3 MOSI	PD2 MISO	PD1 Tx	PD0 RxD

I = Indeterminate after reset

Figure 6-7. Port D Data Register (PORTD)

Address: $1009

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:			DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 6-8. Port D Data Direction Register (DDRD)

Bits [7:6] — Unimplemented

Always read 0

DDRD[5:0] — Port D Data Direction Bits

When DDRD bit 5 is 1 and MSTR = 1 in SPCR, PD5/$\overline{SS}$ is a general-purpose output and mode fault logic is disabled.

0 = Input
1 = Output

7.7.4 Serial Communication Status Register

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

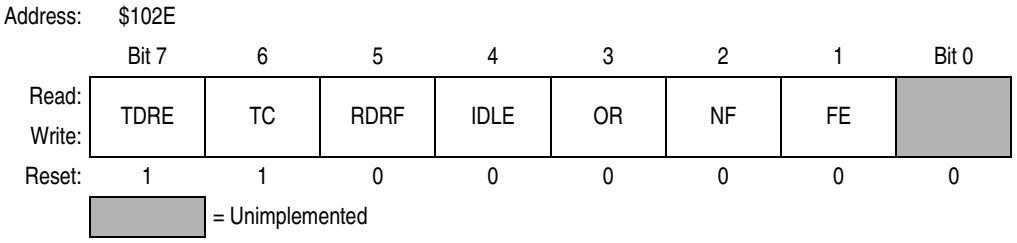


Figure 7-6. Serial Communications Status Register (SCSR)

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
0 = 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

FE — Framing Error Flag

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

0 = Stop bit detected
1 = Zero detected

Bit 0 — Unimplemented

Always reads 0

7.7.5 Baud Rate Register

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

Address: $102B

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

U = Unaffected

Figure 7-7. Baud Rate Register (BAUD)

TCLR — Clear Baud Rate Counter Bit (Test)

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

NOTE

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

RCKB — SCI Baud Rate Clock Check Bit (Test)

See Table 7-1.

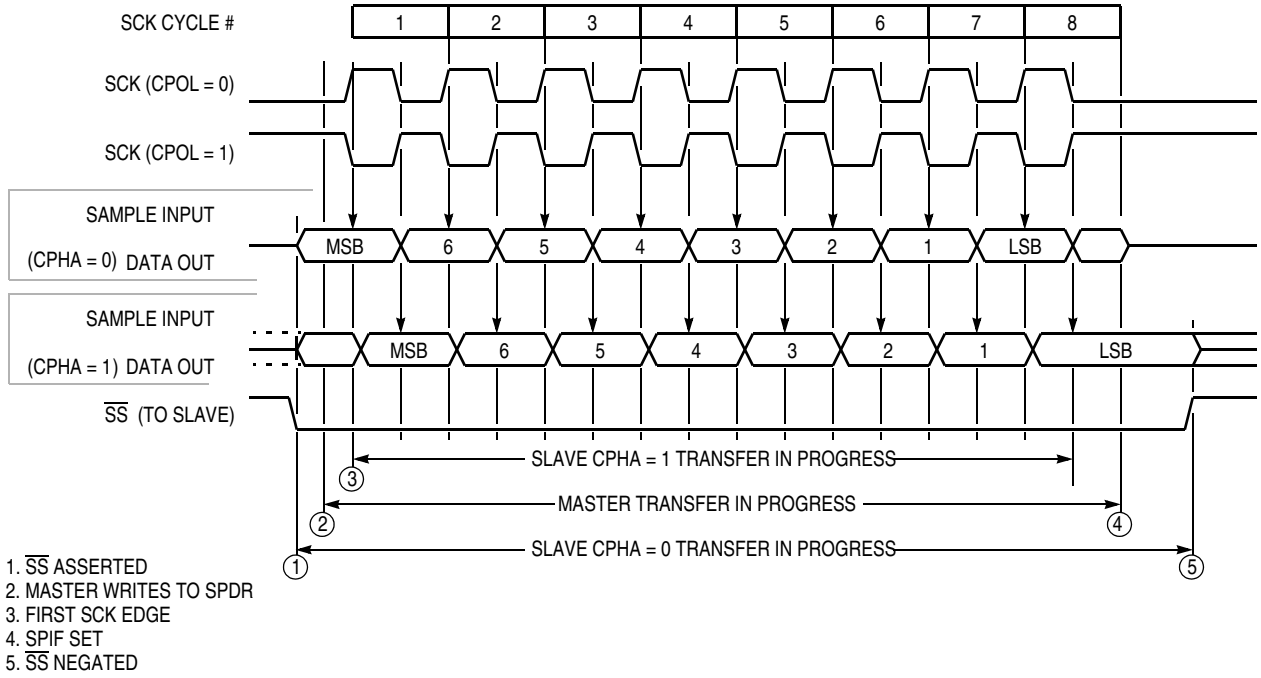


Figure 8-2. SPI Transfer Format

8.5 SPI Signals

This subsection contains descriptions of the four SPI signals:

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

Any SPI output line must have its corresponding data direction bit in DDRD register set. If the DDR bit is clear, that line is disconnected from the SPI logic and becomes a general-purpose input. All SPI input lines are forced to act as inputs regardless of the state of the corresponding DDR bits in DDRD register.

8.5.1 Master In/Slave Out

MISO is one of two unidirectional serial data signals. It is an input to a master device and an output from a slave device. The MISO line of a slave device is placed in the high-impedance state if the slave device is not selected.

8.5.2 Master Out/Slave In

The MOSI line is the second of the two unidirectional serial data signals. It is an output from a master device and an input to a slave device. The master device places data on the MOSI line a half-cycle before the clock edge that the slave device uses to latch the data.

9.4.9 Timer Interrupt Mask 2 Register

Use this 8-bit register to enable or inhibit timer overflow and real-time interrupts. The timer prescaler control bits are included in this register.

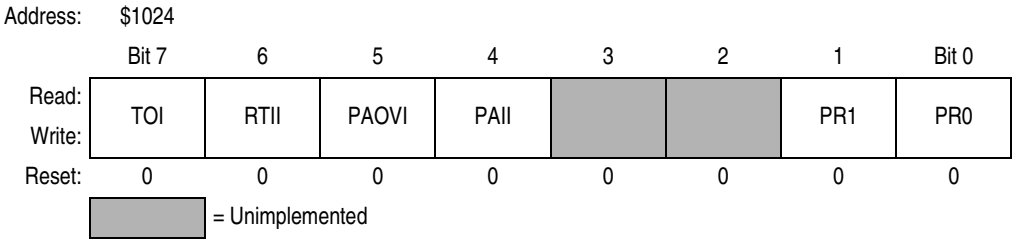


Figure 9-19. Timer Interrupt Mask 2 Register (TMSK2)

TOI — Timer Overflow Interrupt Enable Bit

0 = TOF interrupts disabled
1 = Interrupt requested when TOF is set to 1

RTII — Real-Time Interrupt Enable Bit

Refer to 9.5 Real-Time Interrupt (RTI).

PAOVI — Pulse Accumulator Overflow Interrupt Enable Bit

Refer to 9.7.3 Pulse Accumulator Status and Interrupt Bits.

PAII — Pulse Accumulator Input Edge Interrupt Enable Bit

Refer to 9.7.3 Pulse Accumulator Status and Interrupt Bits.

Bits [3:2] — Unimplemented

Always read 0

PR[1:0] — Timer Prescaler Select Bits

These bits are used to select the prescaler divide-by ratio. In normal modes, PR[1:0] can be written only once, and the write must be within 64 cycles after reset. Refer to Table 9-1 and Table 9-4 for specific timing values.

Table 9-4. Timer Prescale

PR[1:0]	Prescaler
0 0	1
0 1	4
1 0	8
1 1	16

NOTE

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

9.7.1 Pulse Accumulator Control Register

Four of this register's bits control an 8-bit pulse accumulator system. Another bit enables either the OC5 function or the IC4 function, while two other bits select the rate for the real-time interrupt system.

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 9-25. 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

0 = Pulse accumulator disabled
1 = Pulse accumulator enabled

PAMOD — Pulse Accumulator Mode Bit

0 = Event counter
1 = Gated time accumulation

PEDGE — Pulse Accumulator Edge Control Bit

This bit has different meanings depending on the state of the PAMOD bit, as shown in Table 9-7.

Table 9-7. Pulse Accumulator Edge Control

PAMOD	PEDGE	Action on Clock
0	0	PAI falling edge increments the counter.
0	1	PAI rising edge increments the counter.
1	0	A 0 on PAI inhibits counting.
1	1	A 1 on PAI inhibits counting.

DDRA3 — Data Direction for Port A Bit 3

Refer to Chapter 6 Parallel Input/Output (I/O) Ports.

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

0 = Output compare 5 function enable (no IC4)
1 = Input capture 4 function enable (no OC5)

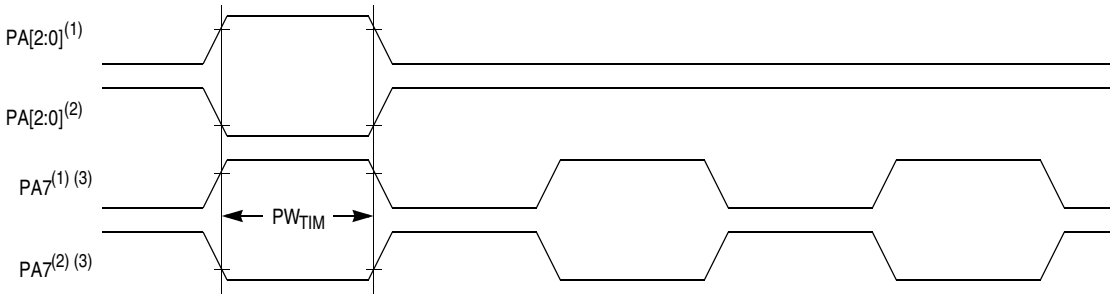
RTR[1:0] — RTI Interrupt Rate Select Bits

Refer to 9.5 Real-Time Interrupt (RTI).

10.10 MC68L11E9/E20 Control Timing

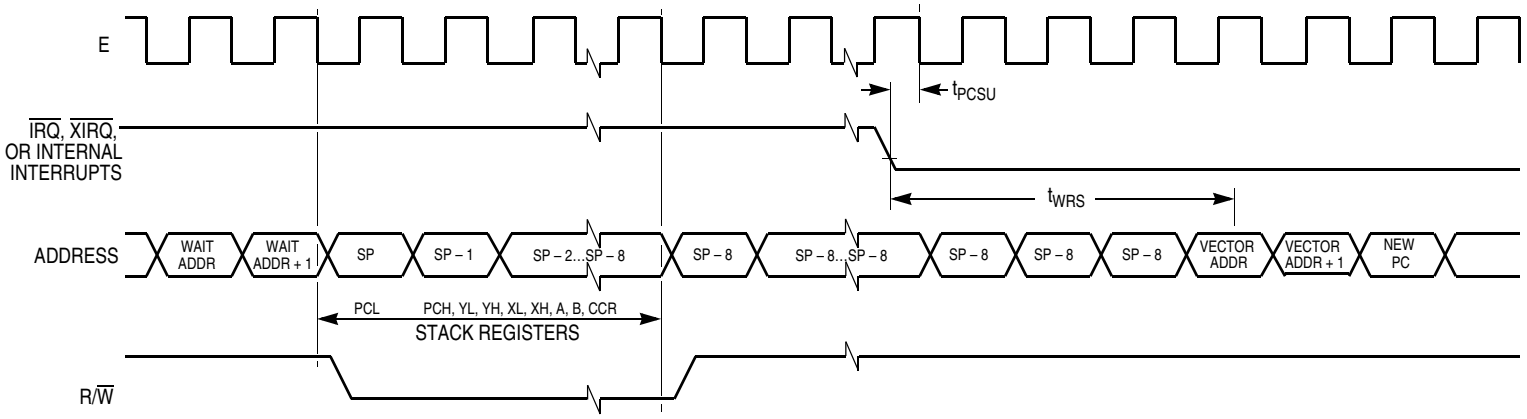
Characteristic[1] [2]	Symbol	1.0 MHz		2.0 MHz		Unit
		Min	Max	Min	Max	
Frequency of operation	f_o	dc	1.0	dc	2.0	MHz
E-clock period	t_{CYC}	1000	—	500	—	ns
Crystal frequency	f_{XTAL}	—	4.0	—	8.0	MHz
External oscillator frequency	4 f_o	dc	4.0	dc	8.0	MHz
Processor control setup time t_{PCSU} = 1/4 t_{CYC}+ 75 ns	t_{PCSU}	325	—	200	—	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	— —	t_{CYC}
Mode programming setup time	t_{MPS}	2	—	2	—	t_{CYC}
Mode programming hold time	t_{MPH}	10	—	10	—	ns
Interrupt pulse width, $\overline{IRQ}$ edge-sensitive mode PW_{IRQ} = t_{CYC} + 20 ns	PW_{IRQ}	1020	—	520	—	ns
Wait recovery startup time	t_{WRS}	—	4	—	4	t_{CYC}
Timer pulse width input capture pulse accumulator input PW_{TIM} = t_{CYC} + 20 ns	PW_{TIM}	1020	—	520	—	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. $\overline{RESET}$ 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.



Notes:
1. Rising edge sensitive input
2. Falling edge sensitive input
3. Maximum pulse accumulator clocking rate is E-clock frequency divided by 2.

Figure 10-2. Timer Inputs



Note: RESET also causes recovery from WAIT.

Figure 10-5. WAIT Recovery from Interrupt Timing Diagram

10.19 EEPROM Characteristics

Characteristic[1]	Temperature Range			Unit
	–40 to 85°C	–40 to 105°C	–40 to 125°C	
Programming time[2] < 1.0 MHz, RCO enabled 1.0 to 2.0 MHz, RCO disabled ≥ 2.0 MHz (or anytime RCO enabled)	 10 20 10	 15 Must use RCO 15	 20 Must use RCO 20	ms
Erase time[2] Byte, row, and bulk	10	10	10	ms
Write/erase endurance	10,000	10,000	10,000	Cycles
Data retention	10	10	10	Years

1. V_{DD} = 5.0 Vdc ±10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H
2. The RC oscillator (RCO) must be enabled (by setting the CSEL bit in the OPTION register) for EEPROM programming and erasure when the E-clock frequency is below 1.0 MHz.

10.20 MC68L11E9/E20 EEPROM Characteristics

Characteristic[1]	Temperature Range –20 to 70°C	Unit
Programming time[2] 3 V, E ≤ 2.0 MHz, RCO enabled 5 V, E ≤ 2.0 MHz, RCO enabled	 25 10	ms
Erase time[2] (byte, row, and bulk) 3 V, E ≤ 2.0 MHz, RCO enabled 5 V, E ≤ 2.0 MHz, RCO enabled	 25 10	ms
Write/erase endurance	10,000	Cycles
Data retention	10	Years

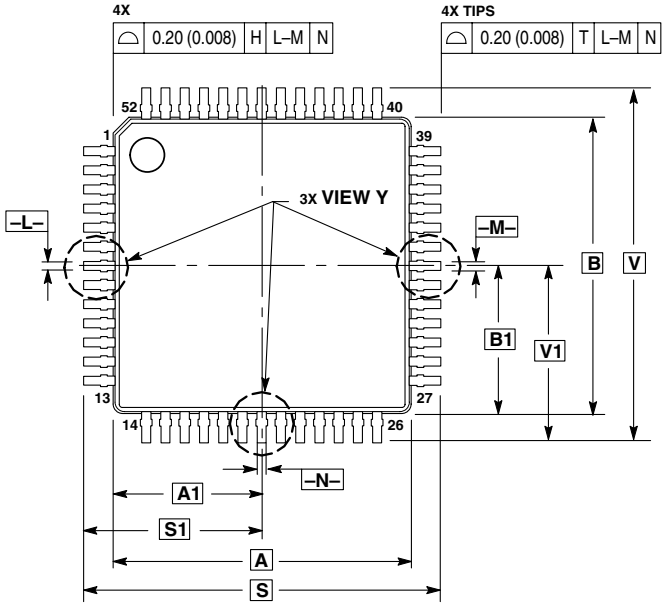
1. V_{DD} = 3.0 Vdc to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H
2. The RC oscillator (RCO) must be enabled (by setting the CSEL bit in the OPTION register) for EEPROM programming and erasure.

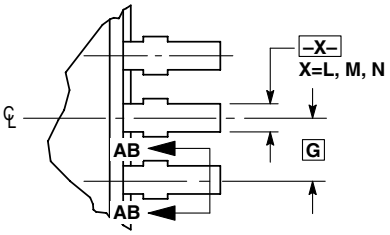
10.21 EPROM Characteristics

Characteristics[1]	Symbol	Min	Typ	Max	Unit
Programming voltage[2]	V_{PPE}	11.75	12.25	12.75	V
Programming current[3]	I_{PPE}	—	3	10	mA
Programming time	t_{EPROG}	2	2	4	ms

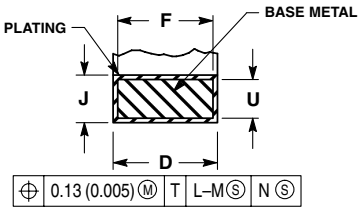
1. V_{DD} = 5.0 Vdc ± 10%
2. During EPROM programming of the MC68HC711E9 device, the V_{PPE} pin circuitry may latch-up and be damaged if the input current is not limited to 10 mA. For more information please refer to MC68HC711E9 8-Bit Microcontroller Unit Mask Set Errata 3 (Freescale document order number 68HC711E9MSE3.
3. Typically, a 1-kΩ series resistor is sufficient to limit the programming current for the MC68HC711E9. A 100-Ω series resistor is sufficient to limit the programming current for the MC68HC711E20.

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

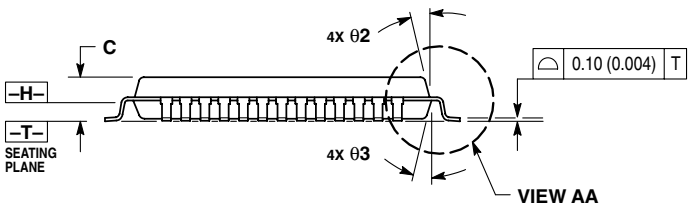


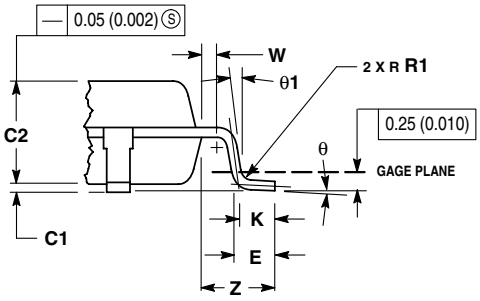


VIEW Y



SECTION AB–AB
ROTATED 90° CLOCKWISE

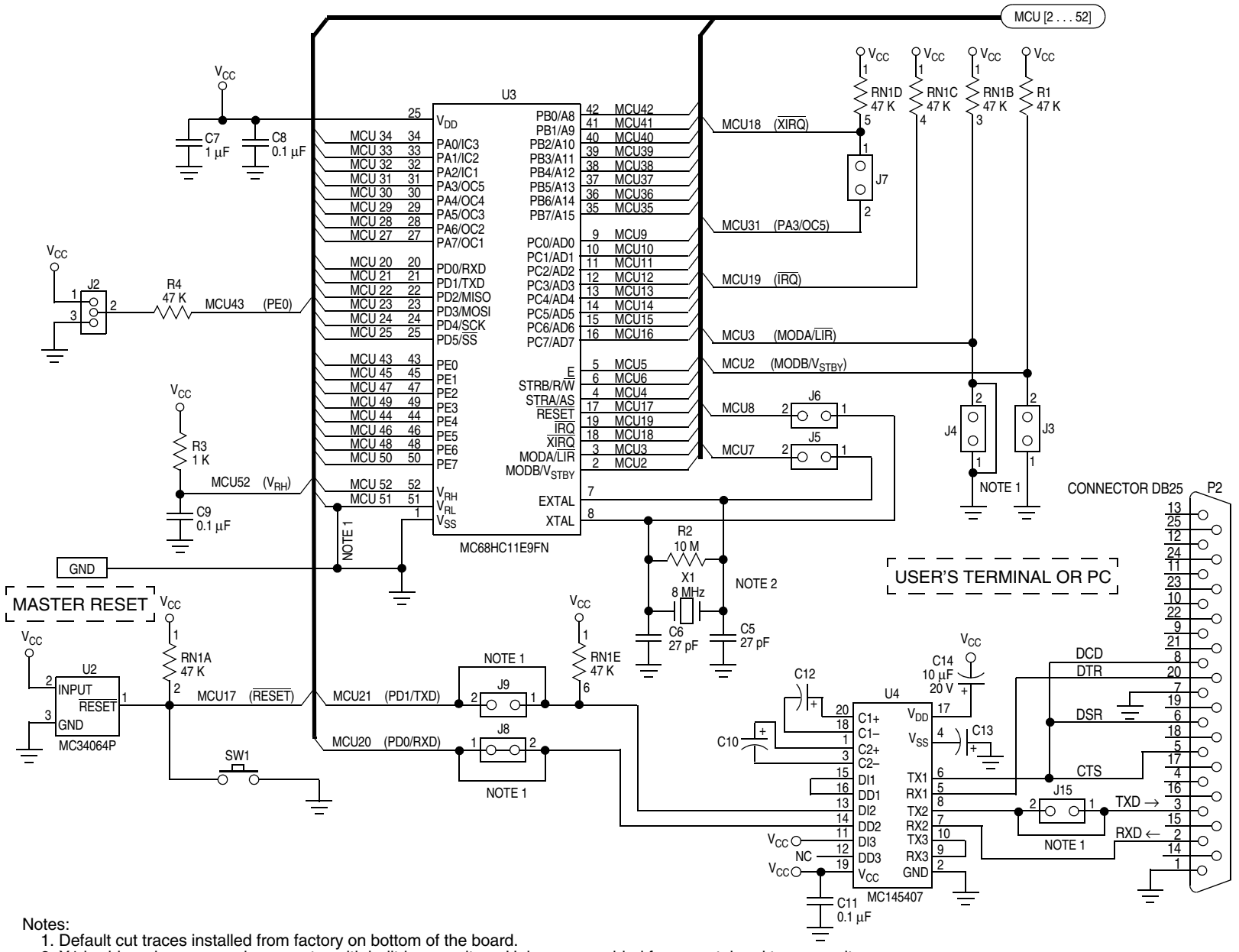




VIEW AA

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°



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 RxD 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.

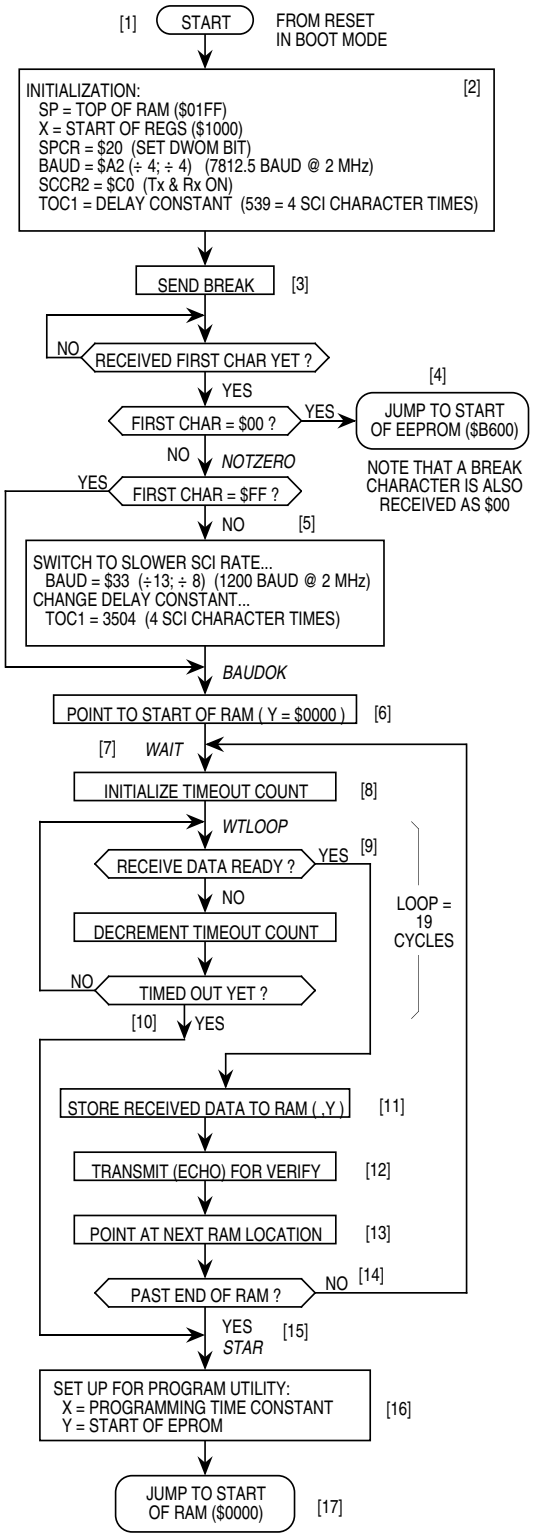


Figure 3. MC68HC711E9 Bootloader Flowchart

```
              Errors: None
              Labels: 28
Last Program Address: $B67C
Last Storage Address: $0000
       Program Bytes: $007D  125
       Storage Bytes: $0000  0
```

Driving Boot Mode from a Personal Computer

In this example, a personal computer is used as the host to drive the bootloader of an MC68HC711E9. An M68HC11 EVBU is used for the target MC68HC711E9. A large program is transferred from the personal computer into the EPROM of the target MC68HC711E9.

Hardware

Figure 7 shows a small modification to the EVBU to accommodate the 12-volt (nominal) EPROM programming voltage. The $\overline{\text{XIRQ}}$ pin is connected to a pullup resistor, two jumpers, and the 60-pin connectors, P4 and P5. The object of the modification is to isolate the $\overline{\text{XIRQ}}$ pin and then connect it to the programming power supply. Carefully cut the trace on the solder side of the EVBU as indicated in Figure 7. This disconnects the pullup resistor RN1 D from $\overline{\text{XIRQ}}$ but leaves P4–18, P5–18, and jumpers J7 and J14 connected so the EVBU can still be used for other purposes after programming is done. Remove any fabricated jumpers from J7 and J14. The EVBU normally has a jumper at J7 to support the trace function

Figure 8 shows a small circuit that is added to the wire-wrap area of the EVBU. The 3-terminal jumper allows the $\overline{\text{XIRQ}}$ line to be connected to either the programming power supply or to a substitute pullup resistor for $\overline{\text{XIRQ}}$. The 100-ohm resistor is a current limiter to protect the 12-volt input of the MCU. The resistor and LED connected to P5 pin 9 (port C bit 0) is an optional indicator that lights when programming is complete.

Software

BASIC was chosen as the programming language due to its readability and availability in parallel versions on both the IBM® PC and the Macintosh®. The program demonstrates several programming techniques for use with an M68HC11 and is not necessarily intended to be a finished, commercial program. For example, there is little error checking, and the user interface is elementary. A complete listing of the BASIC program is included in Listing 2. BASIC Program for Personal Computer with moderate comments. The following paragraphs include a more detailed discussion of the program as it pertains to communicating with and programming the target MC68HC711E9. Lines 25–45 initialize and define the variables and array used in the program. Changes to this section would allow for other programs to be downloaded.

® IBM is a registered trademark of International Business Machines.
® Macintosh is a registered trademark of Apple Computers, Inc.

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.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

M68HC11E
Rev. 5.1, 07/2005

freescale™
semiconductor