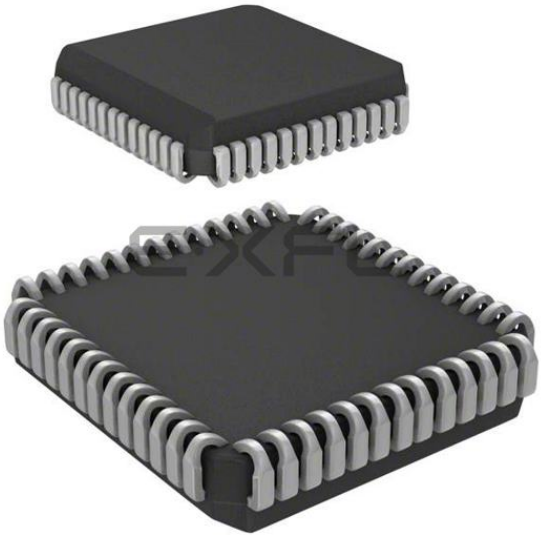


Welcome to E-XFL.COM

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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	52-LCC (J-Lead)
Supplier Device Package	52-PLCC (19.1x19.1)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc11e0mfne2

Revision History

Date	Revision Level	Description	Page Number(s)
May, 2001	3.1	2.3.3.1 System Configuration Register — Addition to NOCOP bit description	44
		Added 10.21 EPROM Characteristics	175
June, 2001	3.2	10.21 EPROM Characteristics — For clarity, addition to note 2 following the table	175
December, 2001	3.3	7.7.2 Serial Communications Control Register 1 — SCCR1 bit 4 (M) description corrected	110
July, 2002	4	10.7 MC68L11E9/E20 DC Electrical Characteristics — Title changed to include the MC68L11E20	153
		10.8 MC68L11E9/E20 Supply Currents and Power Dissipation — Title changed to include the MC68L11E20	154
		10.10 MC68L11E9/E20 Control Timing — Title changed to include the MC68L11E20	157
		10.12 MC68L11E9/E20 Peripheral Port Timing — Title changed to include the MC68L11E20	163
		10.14 MC68L11E9/E20 Analog-to-Digital Converter Characteristics — Title changed to include the MC68L11E20	167
		10.16 MC68L11E9/E20 Expansion Bus Timing Characteristics — Title changed to include the MC68L11E20	169
		10.18 MC68L11E9/E20 Serial Peirpheral Interface Characteristics — Title changed to include the MC68L11E20	172
		— Title changed to include the MC68L11E20	175
		11.4 Extended Voltage Device Ordering Information (3.0 Vdc to 5.5 Vdc) — Updated table to include MC68L1120	181
June, 2003	5	Format updated to current publications standards	Throughout
		1.4.6 Non-Maskable Interrupt (XIRQ/VPPE) — Added Caution note pertaining to EPROM programming of the MC68HC711E9 device only.	23
		6.4 Port C — Clarified description of DDRC[7:0] bits	100
		10.21 EPROM Characteristics — Added note pertaining to EPROM programming of the MC68HC711E9 device only.	175
July, 2005	5.1	Updated to meet Freescale identity guidelines.	Throughout

Chapter 9
Timing Systems

Chapter 10
Electrical Characteristics

Chapter 1
General Description

1.1 Introduction

This document contains a detailed description of the M68HC11 E series of 8-bit microcontroller units (MCUs). These MCUs all combine the M68HC11 central processor unit (CPU) with high-performance, on-chip peripherals.

The E series is comprised of many devices with various configurations of:

- Random-access memory (RAM)
- Read-only memory (ROM)
- Erasable programmable read-only memory (EPROM)
- Electrically erasable programmable read-only memory (EEPROM)
- Several low-voltage devices are also available.

With the exception of a few minor differences, the operation of all E-series MCUs is identical. A fully static design and high-density complementary metal-oxide semiconductor (HCMOS) fabrication process allow the E-series devices to operate at frequencies from 3 MHz to dc with very low power consumption.

1.2 Features

Features of the E-series devices include:

- M68HC11 CPU
- Power-saving stop and wait modes
- Low-voltage devices available (3.0–5.5 Vdc)
- 0, 256, 512, or 768 bytes of on-chip RAM, data retained during standby
- 0, 12, or 20 Kbytes of on-chip ROM or EPROM
- 0, 512, or 2048 bytes of on-chip EEPROM with block protect for security
- 2048 bytes of EEPROM with selectable base address in the MC68HC811E2
- Asynchronous non-return-to-zero (NRZ) serial communications interface (SCI)
- Additional baud rates available on MC68HC(7)11E20
- Synchronous serial peripheral interface (SPI)
- 8-channel, 8-bit analog-to-digital (A/D) converter
- 16-bit timer system:
  - Three input capture (IC) channels
  - Four output compare (OC) channels
  - One additional channel, selectable as fourth IC or fifth OC
- 8-bit pulse accumulator
- Real-time interrupt circuit

1.4.15 Port D

Pins PD5–PD0 can be used for general-purpose I/O signals. These pins alternately serve as the serial communication interface (SCI) and serial peripheral interface (SPI) signals when those subsystems are enabled.

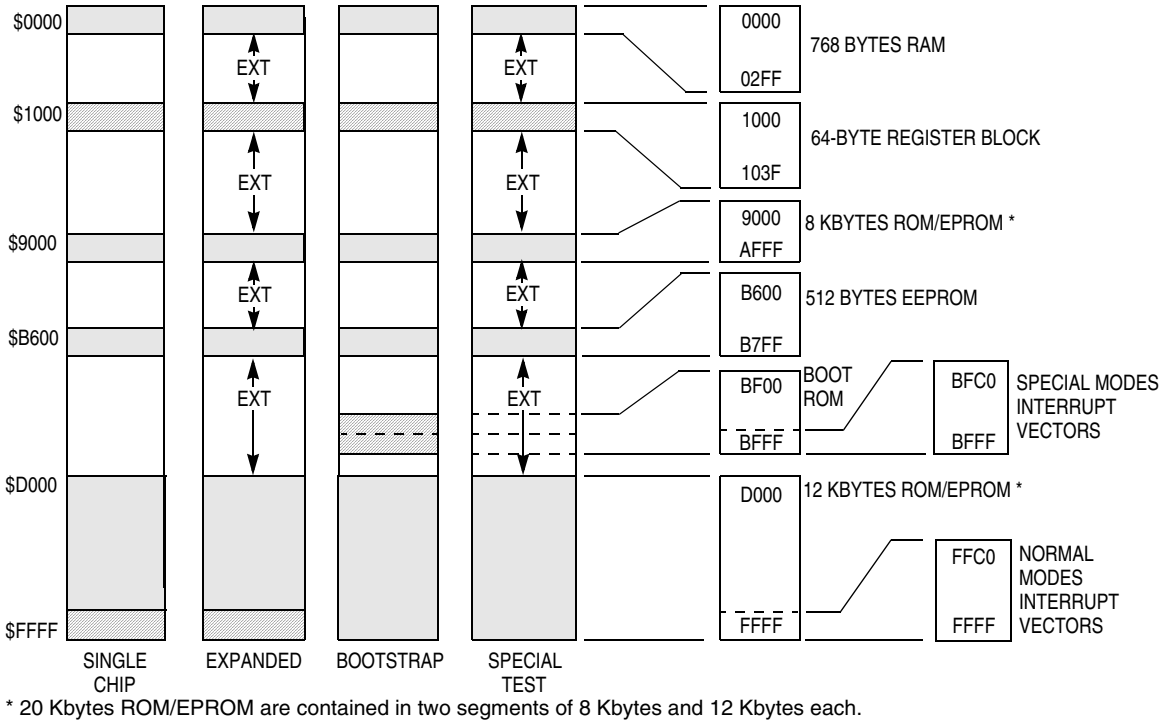
- PD0 is the receive data input (RxD) signal for the SCI.
- PD1 is the transmit data output (TxD) signal for the SCI.
- PD5–PD2 are dedicated to the SPI:
  - PD2 is the master in/slave out (MISO) signal.
  - PD3 is the master out/slave in (MOSI) signal.
  - PD4 is the serial clock (SCK) signal.
  - PD5 is the slave select ($\overline{SS}$) input.

1.4.16 Port E

Use port E for general-purpose or analog-to-digital (A/D) inputs.

CAUTION

If high accuracy is required for A/D conversions, avoid reading port E during sampling, as small disturbances can reduce the accuracy of that result.



* 20 Kbytes ROM/EPROM are contained in two segments of 8 Kbytes and 12 Kbytes each.

Figure 2-5. Memory Map for MC68HC(7)11E20

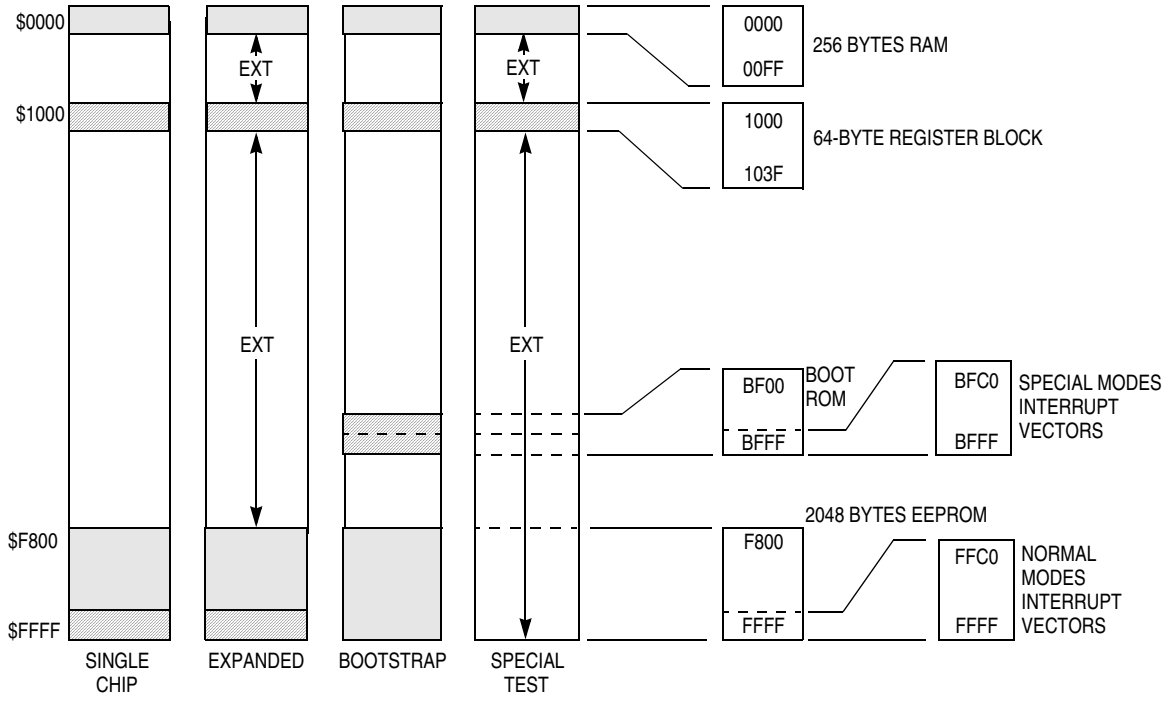


Figure 2-6. Memory Map for MC68HC811E2

2.4.1 Programming an Individual EPROM Address

- In this method, the MCU programs its own EPROM by controlling the PPROG register (EPROG in MC68HC711E20). Use these procedures to program the EPROM through the MCU with:
- The ROMON bit set in the CONFIG register
- The 12-volt nominal programming voltage present on the $\overline{\text{XIRQ}}/V_{PPE}$ pin
- The $\overline{\text{IRQ}}$ pin must be pulled high.

NOTE

Any operating mode can be used.

This example applies to all devices with EPROM/OTPROM except for the MC68HC711E20.

```
EPROG   LDAB    #$20
        STAB    $103B   Set ELAT bit in (EPGM = 0) to enable
                        EPROM latches.
        STAA    $0,X    Store data to EPROM address
        LDAB    #$21
        STAB    $103B   Set EPGM bit with ELAT = 1 to enable
                        EPROM programming voltage
        JSR     DLYEP   Delay 2-4 ms
        CLR     $103B   Turn off programming voltage and set
                        to READ mode
```

This example applies only to MC68HC711E20.

```
EPROG   LDAB    #$20
        STAB    $1036   Set ELAT bit (EPGM = 0) to enable
                        EPROM latches.
        STAA    $0,X    Store data to EPROM address
        LDAB    #$21
        STAB    $1036   Set EPGM bit with ELAT = 1 to enable
                        EPROM programming voltage
        JSR     DLYEP   Delay 2-4 ms
        CLR     $1036   Turn off programming voltage and set
                        to READ mode
```

2.4.2 Programming the EPROM with Downloaded Data

When using this method, the EPROM is programmed by software while in the special test or bootstrap modes. User-developed software can be uploaded through the SCI or a ROM-resident EPROM programming utility can be used. The 12-volt nominal programming voltage must be present on the $\overline{\text{XIRQ}}/V_{PPE}$ pin. To use the resident utility, bootload a 3-byte program consisting of a single jump instruction to $BF00. $BF00 is the starting address of a resident EPROM programming utility. The utility program sets the X and Y index registers to default values, then receives programming data from an external host, and puts it in EPROM. The value in IX determines programming delay time. The value in IY is a pointer to the first address in EPROM to be programmed (default = $D000).

When the utility program is ready to receive programming data, it sends the host the $FF character. Then it waits. When the host sees the $FF character, the EPROM programming data is sent, starting with the first location in the EPROM array. After the last byte to be programmed is sent and the corresponding verification data is returned, the programming operation is terminated by resetting the MCU.

For more information, Freescale application note AN1060 entitled M68HC11 Bootstrap Mode has been included at the back of this document.

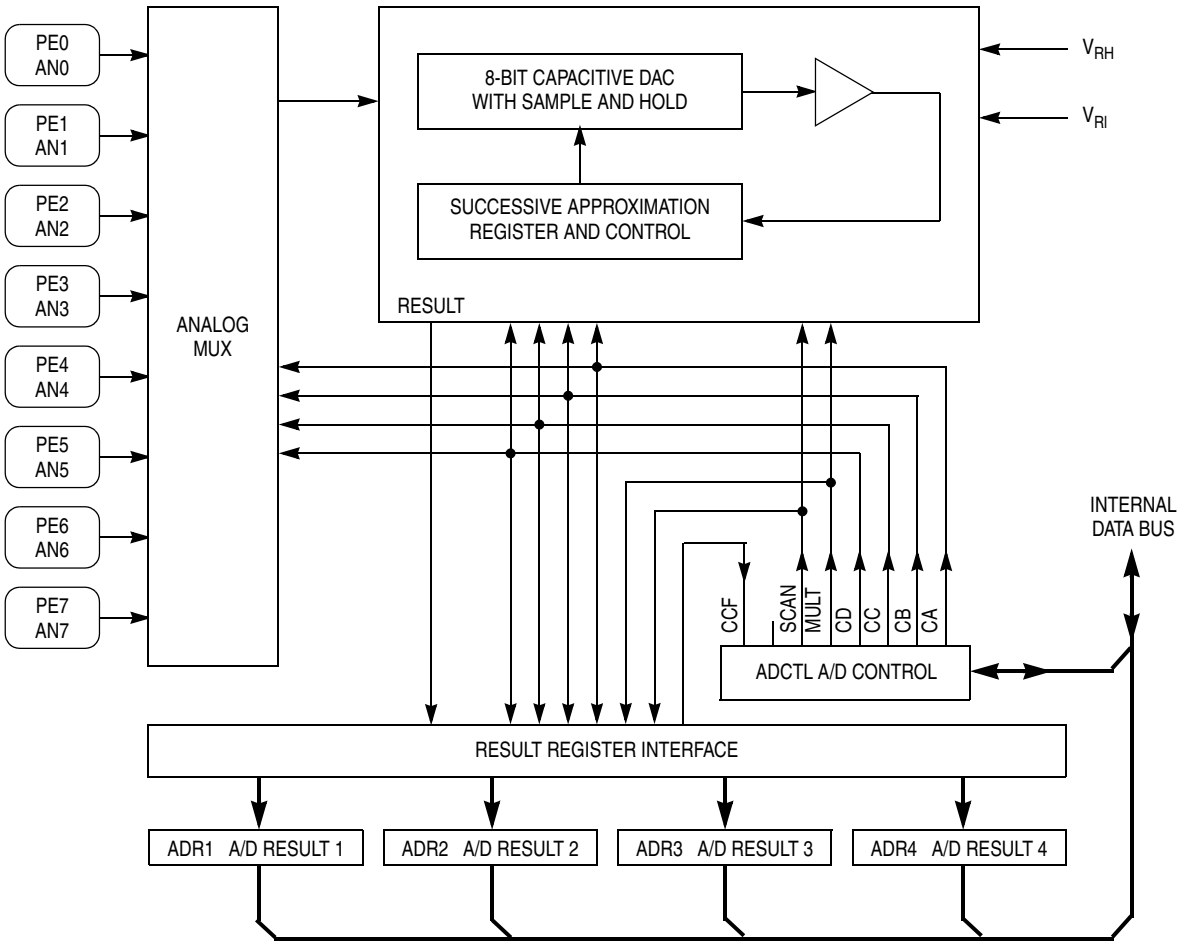
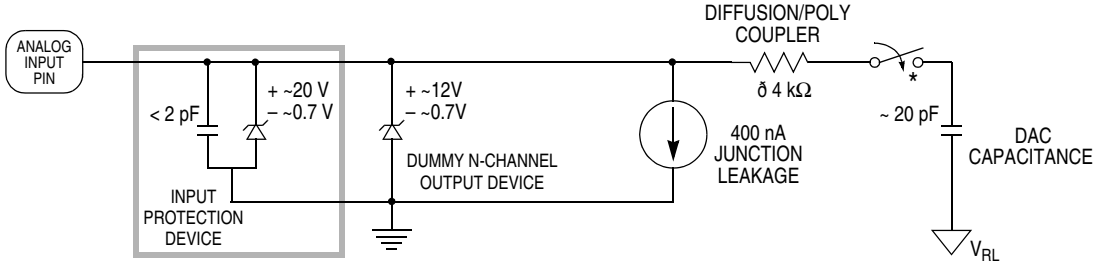


Figure 3-1. A/D Converter Block Diagram



* THIS ANALOG SWITCH IS CLOSED ONLY DURING THE 12-CYCLE SAMPLE TIME.

Figure 3-2. Electrical Model of an A/D Input Pin (Sample Mode)

4.4 Opcodes and Operands

The M68HC11 Family of microcontrollers uses 8-bit opcodes. Each opcode identifies a particular instruction and associated addressing mode to the CPU. Several opcodes are required to provide each instruction with a range of addressing capabilities. Only 256 opcodes would be available if the range of values were restricted to the number able to be expressed in 8-bit binary numbers.

A 4-page opcode map has been implemented to expand the number of instructions. An additional byte, called a prebyte, directs the processor from page 0 of the opcode map to one of the other three pages. As its name implies, the additional byte precedes the opcode.

A complete instruction consists of a prebyte, if any, an opcode, and zero, one, two, or three operands. The operands contain information the CPU needs for executing the instruction. Complete instructions can be from one to five bytes long.

4.5 Addressing Modes

Six addressing modes can be used to access memory:

- Immediate
- Direct
- Extended
- Indexed
- Inherent
- Relative

These modes are detailed in the following paragraphs. All modes except inherent mode use an effective address. The effective address is the memory address from which the argument is fetched or stored or the address from which execution is to proceed. The effective address can be specified within an instruction, or it can be calculated.

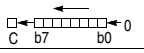
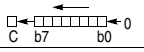
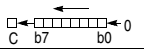
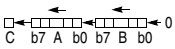
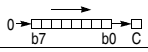
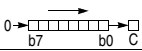
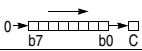
4.5.1 Immediate

In the immediate addressing mode, an argument is contained in the byte(s) immediately following the opcode. The number of bytes following the opcode matches the size of the register or memory location being operated on. There are 2-, 3-, and 4- (if prebyte is required) byte immediate instructions. The effective address is the address of the byte following the instruction.

4.5.2 Direct

In the direct addressing mode, the low-order byte of the operand address is contained in a single byte following the opcode, and the high-order byte of the address is assumed to be $00. Addresses $00–$FF are thus accessed directly, using 2-byte instructions. Execution time is reduced by eliminating the additional memory access required for the high-order address byte. In most applications, this 256-byte area is reserved for frequently referenced data. In M68HC11 MCUs, the memory map can be configured for combinations of internal registers, RAM, or external memory to occupy these addresses.

Table 4-2. Instruction Set (Sheet 4 of 7)

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
INCB	Increment Accumulator B	B + 1 ⇒ B	B INH	5C	—	2	—	—	—	—	Δ	Δ	Δ	—
INS	Increment Stack Pointer	SP + 1 ⇒ SP	INH	31	—	3	—	—	—	—	—	—	—	—
INX	Increment Index Register X	IX + 1 ⇒ IX	INH	08	—	3	—	—	—	—	—	Δ	—	—
INY	Increment Index Register Y	IY + 1 ⇒ IY	INH	18 08	—	4	—	—	—	—	—	Δ	—	—
JMP (opr)	Jump	See Figure 3–2	EXT IND,X IND,Y	7E 6E 18 6E	hh ll ff ff	3 3 4	—	—	—	—	—	—	—	—
JSR (opr)	Jump to Subroutine	See Figure 3–2	DIR EXT IND,X IND,Y	9D BD AD 18 AD	dd hh ll ff ff	5 6 6 7	—	—	—	—	—	—	—	—
LDAA (opr)	Load Accumulator A	M ⇒ A	A IMM A DIR A EXT A IND,X A IND,Y	86 96 B6 A6 18 A6	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
LDAB (opr)	Load Accumulator B	M ⇒ B	B IMM B DIR B EXT B IND,X B IND,Y	C6 D6 F6 E6 18 E6	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
LDD (opr)	Load Double Accumulator D	M ⇒ A,M + 1 ⇒ B	IMM DIR EXT IND,X IND,Y	CC DC FC EC 18 EC	jj kk dd hh ll ff ff	3 4 5 5 6	—	—	—	—	Δ	Δ	0	—
LDS (opr)	Load Stack Pointer	M : M + 1 ⇒ SP	IMM DIR EXT IND,X IND,Y	8E 9E BE AE 18 AE	jj kk dd hh ll ff ff	3 4 5 5 6	—	—	—	—	Δ	Δ	0	—
LDX (opr)	Load Index Register X	M : M + 1 ⇒ IX	IMM DIR EXT IND,X IND,Y	CE DE FE EE CD EE	jj kk dd hh ll ff ff	3 4 5 5 6	—	—	—	—	Δ	Δ	0	—
LDY (opr)	Load Index Register Y	M : M + 1 ⇒ IY	IMM DIR EXT IND,X IND,Y	18 CE 18 DE 18 FE 1A EE 18 EE	jj kk dd hh ll ff ff	4 5 6 6 6	—	—	—	—	Δ	Δ	0	—
LSL (opr)	Logical Shift Left		EXT IND,X IND,Y	78 68 18 68	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
LSLA	Logical Shift Left A		A INH	48	—	2	—	—	—	—	Δ	Δ	Δ	Δ
LSLB	Logical Shift Left B		B INH	58	—	2	—	—	—	—	Δ	Δ	Δ	Δ
LSLD	Logical Shift Left Double		INH	05	—	3	—	—	—	—	Δ	Δ	Δ	Δ
LSR (opr)	Logical Shift Right		EXT IND,X IND,Y	74 64 18 64	hh ll ff ff	6 6 7	—	—	—	—	0	Δ	Δ	Δ
LSRA	Logical Shift Right A		A INH	44	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRB	Logical Shift Right B		B INH	54	—	2	—	—	—	—	0	Δ	Δ	Δ

5.2.2 External Reset ($\overline{\text{RESET}}$)

The CPU distinguishes between internal and external reset conditions by sensing whether the reset pin rises to a logic 1 in less than two E-clock cycles after an internal device releases reset. When a reset condition is sensed, the $\overline{\text{RESET}}$ pin is driven low by an internal device for four E-clock cycles, then released. Two E-clock cycles later it is sampled. If the pin is still held low, the CPU assumes that an external reset has occurred. If the pin is high, it indicates that the reset was initiated internally by either the COP system or the clock monitor.

CAUTION

Do not connect an external resistor capacitor (RC) power-up delay circuit to the reset pin of M68HC11 devices because the circuit charge time constant can cause the device to misinterpret the type of reset that occurred.

5.2.3 Computer Operating Properly (COP) Reset

The MCU includes a COP system to help protect against software failures. When the COP is enabled, the software is responsible for keeping a free-running watchdog timer from timing out. When the software is no longer being executed in the intended sequence, a system reset is initiated.

The state of the NOCOP bit in the CONFIG register determines whether the COP system is enabled or disabled. To change the enable status of the COP system, change the contents of the CONFIG register and then perform a system reset. In the special test and bootstrap operating modes, the COP system is initially inhibited by the disable resets (DISR) control bit in the TEST1 register. The DISR bit can subsequently be written to 0 to enable COP resets.

The COP timer rate control bits CR[1:0] in the OPTION register determine the COP timeout period. The system E clock is divided by 2^{15} and then further scaled by a factor shown in Table 5-1. After reset, these bits are 0, which selects the fastest timeout period. In normal operating modes, these bits can be written only once within 64 bus cycles after reset.

Table 5-1. COP Timer Rate Select

CR[1:0]	Divide E/2^{15} By	XTAL = 4.0 MHz Timeout – 0 ms, + 32.8 ms	XTAL = 8.0 MHz Timeout – 0 ms, + 16.4 ms	XTAL = 12.0 MHz Timeout – 0 ms, + 10.9 ms	XTAL = 16.0 MHz Timeout – 0 ms, + 8.2 ms
0 0	1	32.768 ms	16.384 ms	10.923 ms	8.19 ms
0 1	4	131.072 ms	65.536 ms	43.691 ms	32.8 ms
1 0	16	524.28 ms	262.14 ms	174.76 ms	131 ms
1 1	64	2.098 s	1.049 s	699.05 ms	524 ms
	E =	1.0 MHz	2.0 MHz	3.0 MHz	4.0 MHz

7.5.2 Address-Mark Wakeup

The serial characters in this type of wakeup consist of seven (eight if M = 1) information bits and an MSB, which indicates an address character (when set to 1, or mark). The first character of each message is an addressing character (MSB = 1). All receivers in the system evaluate this character to determine if the remainder of the message is directed toward this particular receiver. As soon as a receiver determines that a message is not intended for it, the receiver activates the RWU function by using a software write to set the RWU bit. Because setting RWU inhibits receiver-related flags, there is no further software overhead for the rest of this message.

When the next message begins, its first character has its MSB set, which automatically clears the RWU bit and enables normal character reception. The first character whose MSB is set is also the first character to be received after wakeup because RWU gets cleared before the stop bit for that frame is serially received. This type of wakeup allows messages to include gaps of idle time, unlike the idle-line method, but there is a loss of efficiency because of the extra bit time for each character (address bit) required for all characters.

7.6 SCI Error Detection

Three error conditions – SCDR overrun, received bit noise, and framing – can occur during generation of SCI system interrupts. Three bits (OR, NF, and FE) in the serial communications status register (SCSR) indicate if one of these error conditions exists.

The overrun error (OR) bit is set when the next byte is ready to be transferred from the receive shift register to the SCDR and the SCDR is already full (RDRF bit is set). When an overrun error occurs, the data that caused the overrun is lost and the data that was already in SCDR is not disturbed. The OR is cleared when the SCSR is read (with OR set), followed by a read of the SCDR.

The noise flag (NF) bit is set if there is noise on any of the received bits, including the start and stop bits. The NF bit is not set until the RDRF flag is set. The NF bit is cleared when the SCSR is read (with FE equal to 1) followed by a read of the SCDR.

When no stop bit is detected in the received data character, the framing error (FE) bit is set. FE is set at the same time as the RDRF. If the byte received causes both framing and overrun errors, the processor only recognizes the overrun error. The framing error flag inhibits further transfer of data into the SCDR until it is cleared. The FE bit is cleared when the SCSR is read (with FE equal to 1) followed by a read of the SCDR.

7.7 SCI Registers

Five addressable registers are associated with the SCI:

- Four control and status registers:
  - Serial communications control register 1 (SCCR1)
  - Serial communications control register 2 (SCCR2)
  - Baud rate register (BAUD)
  - Serial communications status register (SCSR)
- One data register:
  - Serial communications data register (SCDR)

The SCI registers are the same for all M68HC11 E-series devices with one exception. The SCI system for MC68HC(7)11E20 contains an extra bit in the BAUD register that provides a greater selection of baud prescaler rates. Refer to 7.7.5 Baud Rate Register, Figure 7-8, and Figure 7-9.

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.

10.3 Functional Operating Range

Rating	Symbol	Value	Unit
Operating temperature range MC68HC(7)11Ex MC68HC(7)11ExC MC68HC(7)11ExV MC68HC(7)11ExM MC68HC811E2 MC68HC811E2C MC68HC811E2V MC68HC811E2M MC68L11Ex	T_A	T_L to T_H 0 to +70 –40 to +85 –40 to +105 –40 to +125 0 to +70 –40 to +85 –40 to +105 –40 to +125 –20 to +70	°C
Operating voltage range	V_{DD}	5.0 ± 10%	V

10.4 Thermal Characteristics

Characteristic	Symbol	Value	Unit
Average junction temperature	T_J	$T_A + (P_D \times \Theta_{JA})$	°C
Ambient temperature	T_A	User-determined	°C
Package thermal resistance (junction-to-ambient) 48-pin plastic DIP (MC68HC811E2 only) 56-pin plastic SDIP 52-pin plastic leaded chip carrier 52-pin plastic thin quad flat pack (TQFP) 64-pin quad flat pack	Θ_{JA}	 50 50 50 85 85	°C/W
Total power dissipation(1)	P_D	$P_{INT} + P_{I/O}$ $K / T_J + 273°C$	W
Device internal power dissipation	P_{INT}	$I_{DD} \times V_{DD}$	W
I/O pin power dissipation(2)	$P_{I/O}$	User-determined	W
A constant(3)	K	$P_D \times (T_A + 273°C)$ $+ \Theta_{JA} \times P_D^2$	W/°C

1. This is an approximate value, neglecting $P_{I/O}$.
2. For most applications, $P_{I/O} \leq P_{INT}$ and can be neglected.
3. K is a constant pertaining to the device. Solve for K with a known T_A and a measured P_D (at equilibrium). Use this value of K to solve for P_D and T_J iteratively for any value of T_A.

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%.

10.17 Serial Peripheral Interface Timing Characteristics

Num	Characteristic[1]	Symbol	E9		E20		Unit
			Min	Max	Min	Max	
	Frequency of operation E clock	f_o	dc	3.0	dc	3.0	MHz
	E-clock period	t_{CYC}	333	—	333	—	ns
	Operating frequency Master Slave	$f_{op(m)}$ $f_{op(s)}$	f_o/32 dc	f_o/2 f_o	f_o/128 dc	f_o/2 f_o	MHz
1	Cycle time Master Slave	$t_{CYC(m)}$ $t_{CYC(s)}$	2 1	32 —	2 1	128 —	t_{CYC}
2	Enable lead time[2] Slave	$t_{lead(s)}$	1	—	1	—	t_{CYC}
3	Enable lag time[2] Slave	$t_{lag(s)}$	1	—	1	—	t_{CYC}
4	Clock (SCK) high time Master Slave	$t_{w(SCKH)m}$ $t_{w(SCKH)s}$	t_{CYC}–25 1/2 t_{CYC}–25	16 t_{CYC} —	t_{CYC}–25 1/2 t_{CYC}–25	64 t_{CYC} —	ns
5	Clock (SCK) low time Master Slave	$t_{w(SCKL)m}$ $t_{w(SCKL)s}$	t_{CYC}–25 1/2 t_{CYC}–25	16 t_{CYC} —	t_{CYC}–25 1/2 t_{CYC}–25	64 t_{CYC} —	ns
6	Data setup time (inputs) Master Slave	$t_{su(m)}$ $t_{su(s)}$	30 30	— —	30 30	— —	ns
7	Data hold time (inputs) Master Slave	$t_{h(m)}$ $t_{h(s)}$	30 30	— —	30 30	— —	ns
8	Slave access time CPHA = 0 CPHA = 1	t_a	0 0	40 40	0 0	40 40	ns
9	Disable time (hold time to high-impedance state) Slave	t_{dis}	—	50	—	50	ns
10	Data valid[3] (after enable edge)	t_v	—	50	—	50	ns
11	Data hold time (outputs) (after enable edge)	t_{ho}	0	—	0	—	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. Time to data active from high-impedance state
3. Assumes 200 pF load on SCK, MOSI, and MISO pins

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.

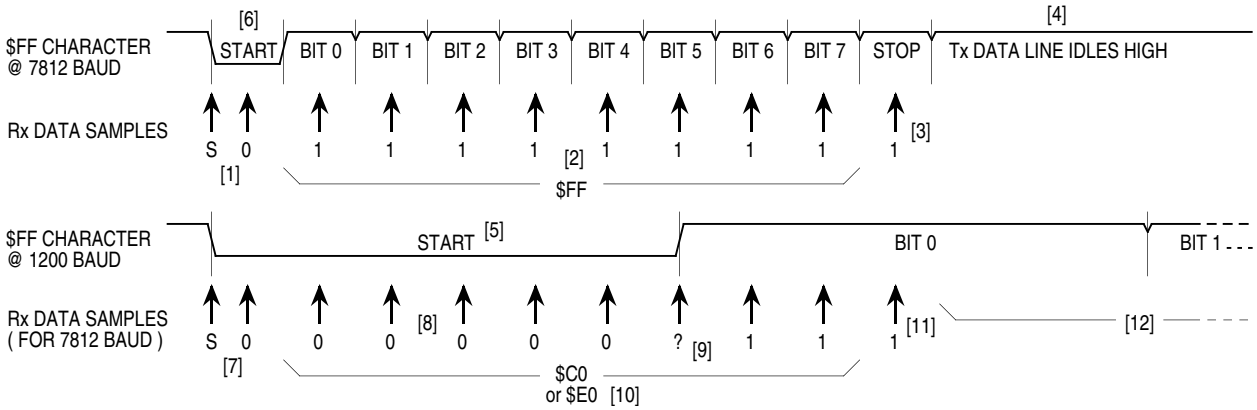


Figure 2. Automatic Detection of Baud Rate

Samples taken at [7] detect the failing edge of the start bit and verify it is a logic 0. Samples taken at the middle of what the receiver interprets as the first five bit times [8] detect logic 0s. The sample taken at the middle of what the receiver interprets as bit 5 [9] may detect either a 0 or a 1 because the receive data has a rising transition at about this time. The samples for bits 6 and 7 detect 1s, causing the receiver to think the received character was $C0 or $E0 [10] at 7812 baud instead of the $FF which was sent at 1200 baud. The stop bit sample detects a 1 as expected [11], but this detection is actually in the middle of bit 0 of the 1200 baud $FF character. The SCI receiver is not confused by the rest of the 1200 baud $FF character because the receive data line is high [12] just as it would be for the idle condition. If a character other than $FF is sent as the first character, an SCI receive error could result.

Main Bootloader Program

Figure 3 is a flowchart of the main bootloader program in the MC68HC711E9. This bootloader demonstrates the most important features of the bootloaders used on all M68HC11 Family members. For complete listings of other M68HC11 versions, refer to Listing 3. MC68HC711E9 Bootloader ROM at the end of this application note, and to **Appendix B** of the *M68HC11 Reference Manual*, Freescale document order number M68HC11RM/AD.

The reset vector in the boot ROM points to the start [1] of this program. The initialization block [2] establishes starting conditions and sets up the SCI and port D. The stack pointer is set because there are push and pull instructions in the bootloader program. The X index register is pointed at the start of the register block ($1000) so indexed addressing can be used. Indexed addressing takes one less byte of ROM space than extended instructions, and bit manipulation instructions are not available in extended addressing forms. The port D wire-OR mode (DWOM) bit in the serial peripheral interface control register (SPCR) is set to configure port D for wired-OR operation to minimize potential conflicts with external systems that use the PD1/TxD pin as an input. The baud rate for the SCI is initially set to 7812 baud at a 2-MHz E-clock rate but can automatically switch to 1200 baud based on the first character received. The SCI receiver and transmitter are enabled. The receiver is required by the bootloading process, and the transmitter is used to transmit data back to the host computer for optional verification. The last item in the initialization is to set an intercharacter delay constant used to terminate the download when the host computer stops sending data to the MC68HC711E9. This delay constant is stored in the timer output compare 1 (TOC1) register, but the on-chip timer is not used in the bootloader program. This example

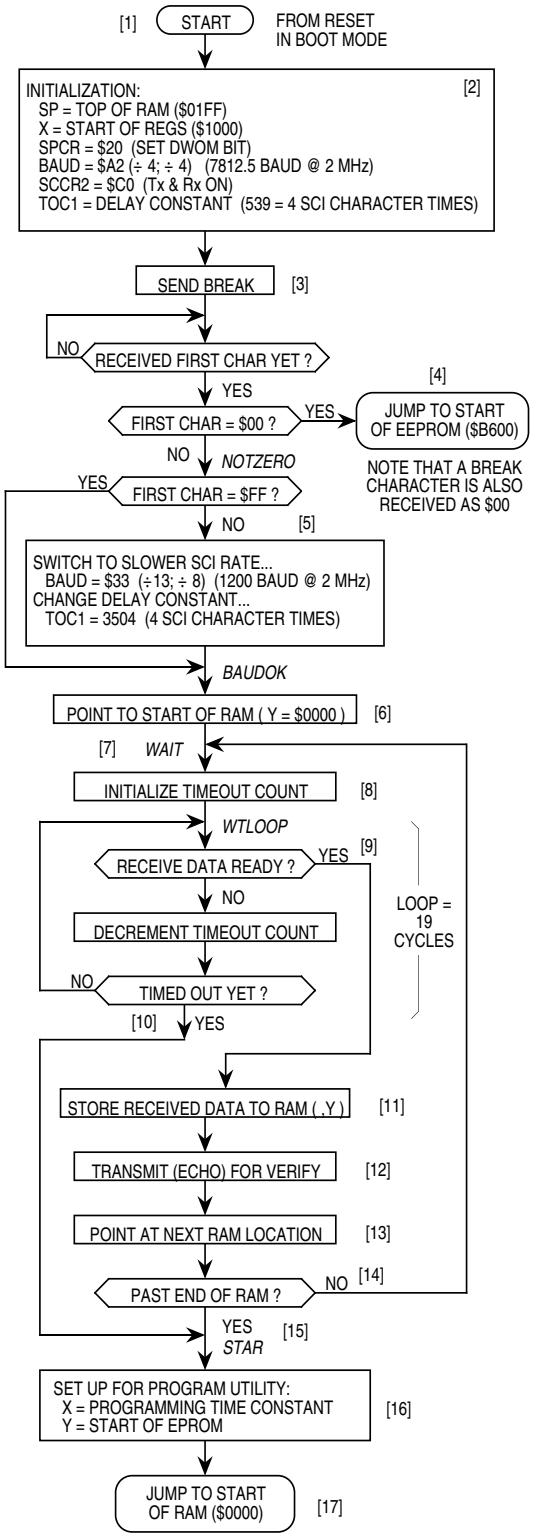


Figure 3. MC68HC711E9 Bootloader Flowchart

Operation

Configure the EVBU for boot mode operation by putting a jumper at J3. Ensure that the trace command jumper at J7 is not installed because this would connect the 12-V programming voltage to the OC5 output of the MCU.

Connect the EVBU to its dc power supply. When it is time to program the MCU EPROM, turn on the 12-volt programming power supply to the new circuitry in the wire-wrap area.

Connect the EVBU serial port to the appropriate serial port on the host system. For the Macintosh, this is the modem port with a modem cable. For the MS-DOS® computer, it is connected to COM1 with a straight through or modem cable. Power up the host system and start the BASIC program. If the program has not been compiled, this is accomplished from within the appropriate BASIC compiler or interpreter. Power up the EVBU.

Answer the prompt for filename with either a [RETURN] to accept the default shown or by typing in a new filename and pressing [RETURN].

The program will inform the user that it is working on converting the file from S records to binary. This process will take from 30 seconds to a few minutes, depending on the computer.

A prompt reading, "Comm port open?" will appear at the end of the file conversion. This is the last chance to ensure that everything is properly configured on the EVBU. Pressing [RETURN] will send the bootcode to the target MC68HC711E9. The program then informs the user that the bootload code is being sent to the target, and the results of the echoing of this code are displayed on the screen.

Another prompt reading "Programming is ready to begin. Are you?" will appear. Turn on the 12-volt programming power supply and press [RETURN] to start the actual programming of the target EPROM.

A count of the byte being verified will be updated continually on the screen as the programming progresses. Any failures will be flagged as they occur.

When programming is complete, a message will be displayed as well as a prompt requesting the user to press [RETURN] to quit.

Turn off the 12-volt programming power supply before turning off 5 volts to the EVBU.

® MS-DOS is a registered trademark of Microsoft Corporation in the United States and oth175190er countries.