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	Active
Core Processor	HC11
Core Size	8-Bit
Speed	3MHz
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	0°C ~ 70°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/pro/item?MUrl=&PartUrl=mc68hcp11e0fn

MC68HC11E Family

Data Sheet

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

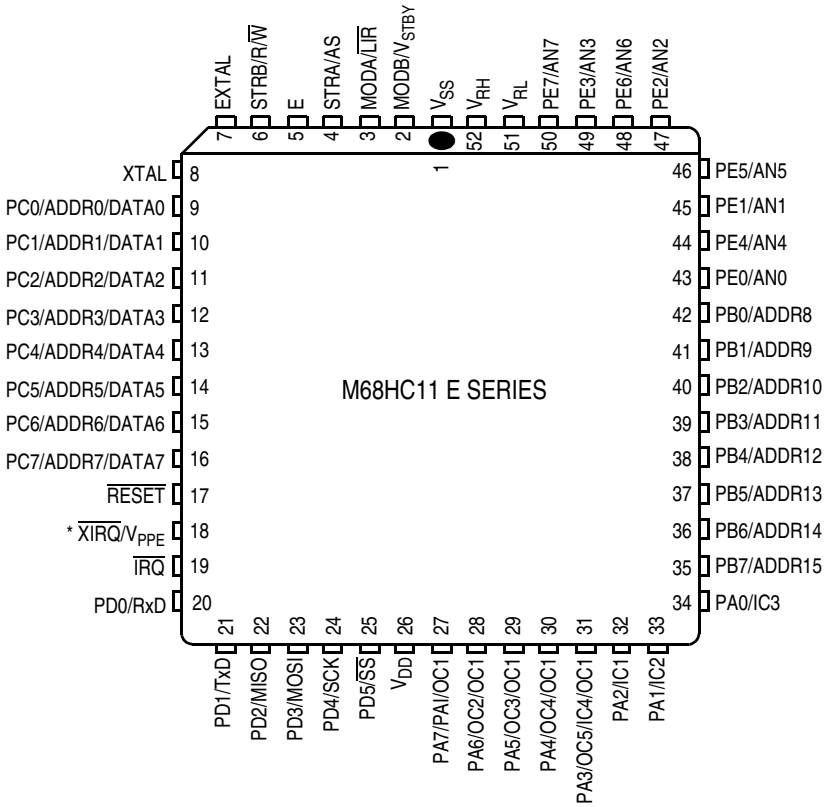
http://freescale.com/

The following revision history table summarizes changes contained in this document. For your convenience, the page number designators have been linked to the appropriate location.

Chapter 11
Ordering Information and Mechanical Specifications

Appendix A
Development Support

Appendix B
EVBU Schematic



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

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

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	

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.

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

PGM — EPROM Programming Voltage Enable Bit

PGM can be read any time and can be written only when ELAT = 1.

0 = Programming voltage to EPROM array disconnected

1 = Programming voltage to EPROM array connected

2.5 EEPROM

Some E-series devices contain 512 bytes of on-chip EEPROM. The MC68HC811E2 contains 2048 bytes of EEPROM with selectable base address. All E-series devices contain the EEPROM-based CONFIG register.

2.5.1 EEPROM and CONFIG Programming and Erasure

The erased state of an EEPROM bit is 1. During a read operation, bit lines are precharged to 1. The floating gate devices of programmed bits conduct and pull the bit lines to 0. Unprogrammed bits remain at the precharged level and are read as ones. Programming a bit to 1 causes no change. Programming a bit to 0 changes the bit so that subsequent reads return 0.

When appropriate bits in the BPROT register are cleared, the PPROG register controls programming and erasing the EEPROM. The PPROG register can be read or written at any time, but logic enforces defined programming and erasing sequences to prevent unintentional changes to EEPROM data. When the EELAT bit in the PPROG register is cleared, the EEPROM can be read as if it were a ROM.

The on-chip charge pump that generates the EEPROM programming voltage from V_{DD} uses MOS capacitors, which are relatively small in value. The efficiency of this charge pump and its drive capability are affected by the level of V_{DD} and the frequency of the driving clock. The load depends on the number of bits being programmed or erased and capacitances in the EEPROM array.

The clock source driving the charge pump is software selectable. When the clock select (CSEL) bit in the OPTION register is 0, the E clock is used; when CSEL is 1, an on-chip resistor-capacitor (RC) oscillator is used.

The EEPROM programming voltage power supply voltage to the EEPROM array is not enabled until there has been a write to PPROG with EELAT set and PGM cleared. This must be followed by a write to a valid EEPROM location or to the CONFIG address, and then a write to PPROG with both the EELAT and EPGM bits set. Any attempt to set both EELAT and EPGM during the same write operation results in neither bit being set.

2.5.1.1 Block Protect Register

This register prevents inadvertent writes to both the CONFIG register and EEPROM. The active bits in this register are initialized to 1 out of reset and can be cleared only during the first 64 E-clock cycles after reset in the normal modes. When these bits are cleared, the associated EEPROM section and the CONFIG register can be programmed or erased. EEPROM is only visible if the EEON bit in the CONFIG register is set. The bits in the BPROT register can be written to 1 at any time to protect EEPROM and the CONFIG register. In test or bootstrap modes, write protection is inhibited and BPROT can be written repeatedly. Address ranges for protected areas of EEPROM differ significantly for the MC68HC811E2. Refer to Figure 2-16.

2.5.1.2 EPROM and EEPROM Programming Control Register

The EPROM and EEPROM programming control register (PPROG) selects and controls the EEPROM programming function. Bits in PPROG enable the programming voltage, control the latching of data to be programmed, and select the method of erasure (for example, byte, row, etc.).

Address: $103B

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

1. MC68HC711E9 only

Figure 2-17. EPROM and EEPROM Programming Control Register (PPROG)

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

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

ELAT — EPROM/OTPROM Latch Control Bit

For the MC68HC711E9, EPGM enables the high voltage necessary for both EPROM/OTPROM and EEPROM programming.

For MC68HC711E9, ELAT and EELAT are mutually exclusive and cannot both equal 1.

0 = EPROM address and data bus configured for normal reads
1 = EPROM address and data bus configured for programming

BYTE — Byte/Other EEPROM Erase Mode Bit

This bit overrides the ROW bit.

0 = Row or bulk erase
1 = Erase only one byte

ROW — Row/All EEPROM Erase Mode Bit

If BYTE is 1, ROW has no meaning.

0 = Bulk erase
1 = Row erase

Table 2-8. EEPROM Erase

BYTE	ROW	Action
0	0	Bulk erase (entire array)
0	1	Row erase (16 bytes)
1	0	Byte erase
1	1	Byte erase

ERASE — Erase Mode Select Bit

0 = Normal read or program mode
1 = Erase mode

EELAT — EEPROM Latch Control Bit

0 = EEPROM address and data bus configured for normal reads and cannot be programmed
1 = EEPROM address and data bus configured for programming or erasing and cannot be read

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

3.1 Introduction

The analog-to-digital (A/D) system, a successive approximation converter, uses an all-capacitive charge redistribution technique to convert analog signals to digital values.

3.2 Overview

The A/D system is an 8-channel, 8-bit, multiplexed-input converter. The converter does not require external sample and hold circuits because of the type of charge redistribution technique used. A/D converter timing can be synchronized to the system E clock or to an internal resistor capacitor (RC) oscillator.

The A/D converter system consists of four functional blocks: multiplexer, analog converter, digital control, and result storage. Refer to Figure 3-1.

3.2.1 Multiplexer

The multiplexer selects one of 16 inputs for conversion. Input selection is controlled by the value of bits CD:CA in the ADCTL register. The eight port E pins are fixed-direction analog inputs to the multiplexer, and additional internal analog signal lines are routed to it.

Port E pins also can be used as digital inputs. Digital reads of port E pins are not recommended during the sample portion of an A/D conversion cycle, when the gate signal to the N-channel input gate is on. Because no P-channel devices are directly connected to either input pins or reference voltage pins, voltages above V_{DD} do not cause a latchup problem, although current should be limited according to maximum ratings. Refer to Figure 3-2, which is a functional diagram of an input pin.

3.2.2 Analog Converter

Conversion of an analog input selected by the multiplexer occurs in this block. It contains a digital-to-analog capacitor (DAC) array, a comparator, and a successive approximation register (SAR). Each conversion is a sequence of eight comparison operations, beginning with the most significant bit (MSB). Each comparison determines the value of a bit in the successive approximation register.

The DAC array performs two functions. It acts as a sample and hold circuit during the entire conversion sequence and provides comparison voltage to the comparator during each successive comparison.

The result of each successive comparison is stored in the SAR. When a conversion sequence is complete, the contents of the SAR are transferred to the appropriate result register.

A charge pump provides switching voltage to the gates of analog switches in the multiplexer. Charge pump output must stabilize between 7 and 8 volts within up to 100 μs before the converter can be used. The charge pump is enabled by the ADPU bit in the OPTION register.

Chapter 4
Central Processor Unit (CPU)

4.1 Introduction

Features of the M68HC11 Family include:

- Central processor unit (CPU) architecture
- Data types
- Addressing modes
- Instruction set
- Special operations such as subroutine calls and interrupts

The CPU is designed to treat all peripheral, input/output (I/O), and memory locations identically as addresses in the 64-Kbyte memory map. This is referred to as memory-mapped I/O. There are no special instructions for I/O that are separate from those used for memory. This architecture also allows accessing an operand from an external memory location with no execution time penalty.

4.2 CPU Registers

M68HC11 CPU registers are an integral part of the CPU and are not addressed as if they were memory locations. The seven registers, discussed in the following paragraphs, are shown in Figure 4-1.

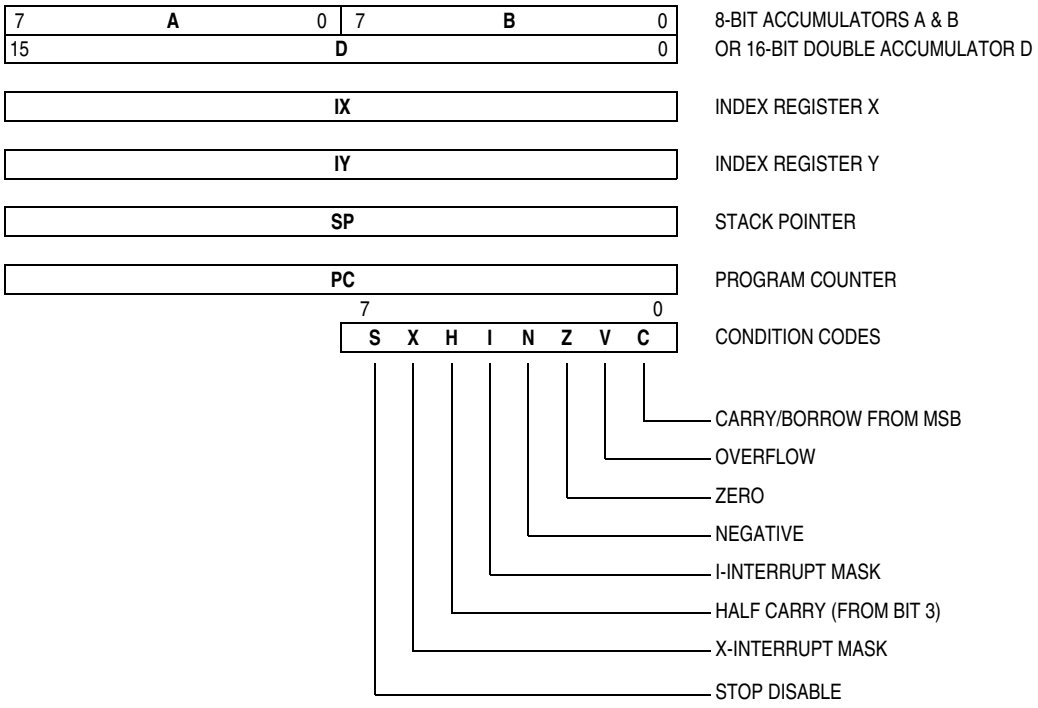


Figure 4-1. Programming Model

5.2.5 System Configuration Options Register

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 mode or at any time in special modes

= Unimplemented

Figure 5-2. System Configuration Options Register (OPTION)

ADPU — Analog-to-Digital Converter Power-Up Bit

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

CSEL — Clock Select Bit

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

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

0 = $\overline{\text{IRQ}}$ is configured for level-sensitive operation.
1 = $\overline{\text{IRQ}}$ is configured for edge-sensitive-only operation.

DLY — Enable Oscillator Startup Delay Bit

Refer to Chapter 2 Operating Modes and On-Chip Memory and Chapter 3 Analog-to-Digital (A/D) Converter.

CME — Clock Monitor Enable Bit

This control bit can be read or written at any time and controls whether or not the internal clock monitor circuit triggers a reset sequence when the system clock is slow or absent. When it is clear, the clock monitor circuit is disabled, and when it is set, the clock monitor circuit is enabled. Reset clears the CME bit.

0 = Clock monitor circuit disabled
1 = Slow or stopped clocks cause reset

Bit 2 — Unimplemented

Always reads 0

CR[1:0] — COP Timer Rate Select Bit

The internal E clock is first divided by 2^{15} before it enters the COP watchdog system. These control bits determine a scaling factor for the watchdog timer. See Table 5-1 for specific timeout settings.

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

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

5.3.10 System

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

5.4 Reset and Interrupt Priority

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

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

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

The maskable interrupt sources have this priority arrangement:

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

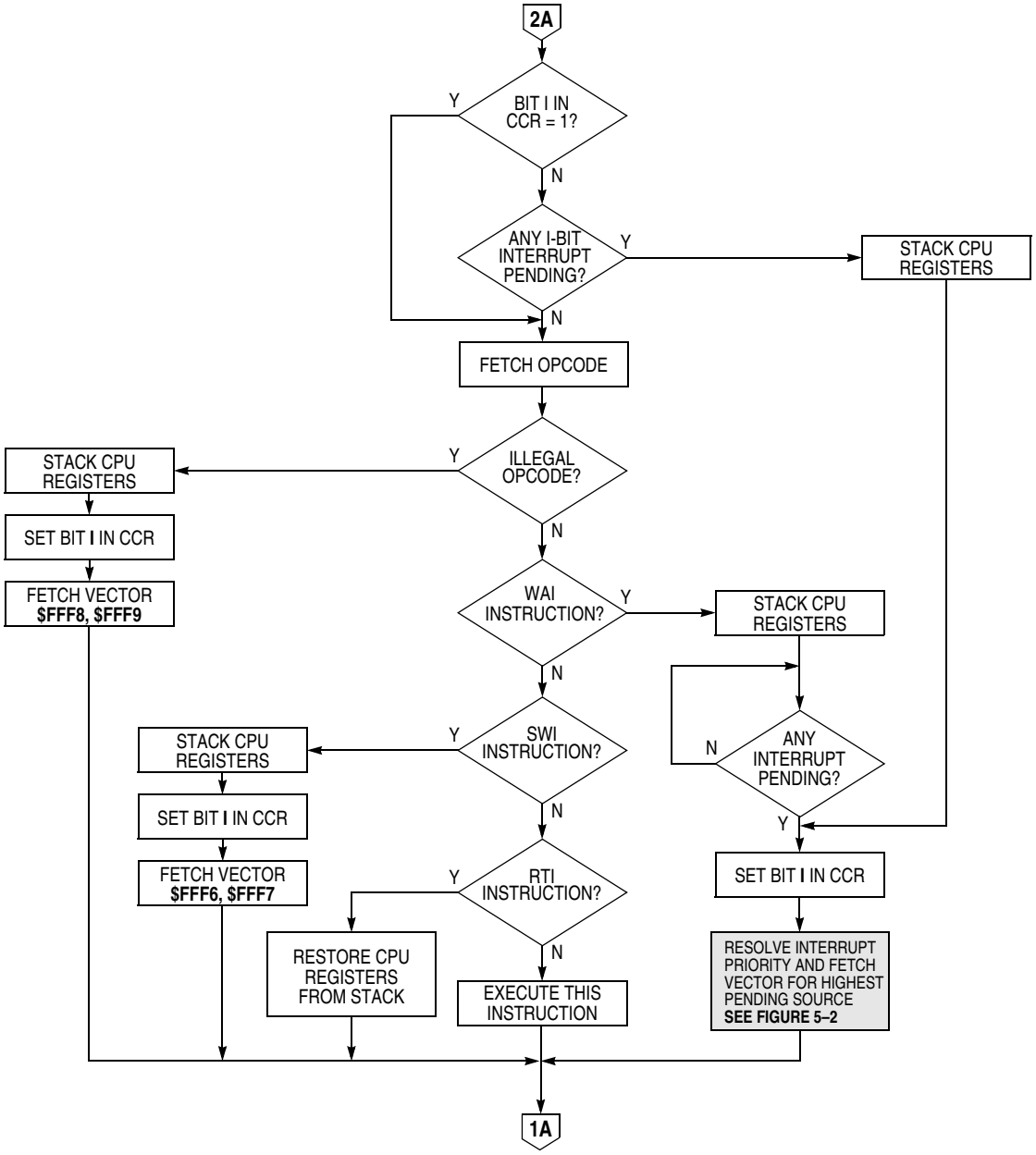


Figure 5-5. Processing Flow Out of Reset (Sheet 2 of 2)

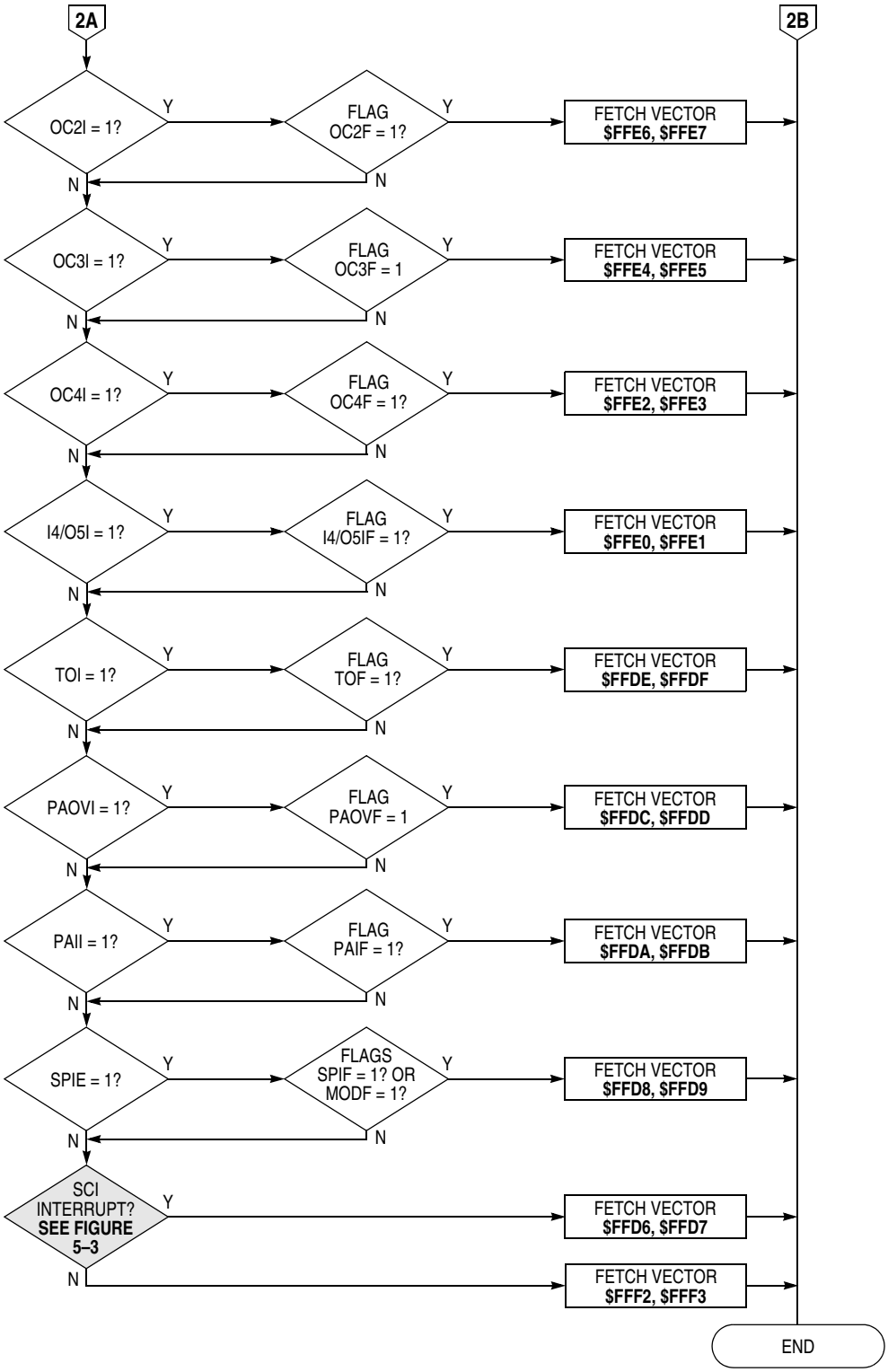


Figure 5-6. Interrupt Priority Resolution (Sheet 2 of 2)

6.3 Port B

In single-chip or bootstrap modes, port B pins are general-purpose outputs. In expanded or special test modes, port B pins are high-order address outputs.

Address: $1004

	Bit 7	6	5	4	3	2	1	Bit 0
Single-chip or bootstrap modes:								
Read: Write:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Reset:	0	0	0	0	0	0	0	0
Expanded or special test modes:								
Read: Write:	ADDR15	ADDR14	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8
Reset:	0	0	0	0	0	0	0	0

Figure 6-3. Port B Data Register (PORTB)

6.4 Port C

In single-chip and bootstrap modes, port C pins reset to high-impedance inputs. (DDRC bits are set to 0.) In expanded and special test modes, port C pins are multiplexed address/data bus and the port C register address is treated as an external memory location.

Address: $1003

	Bit 7	6	5	4	3	2	1	Bit 0
Single-chip or bootstrap modes:								
Read: Write:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Reset:	Indeterminate after reset							
Expanded or special test modes:								
Read: Write:	ADDR7 DATA7	ADDR6 DATA6	ADDR5 DATA5	ADDR4 DATA4	ADDR3 DATA3	ADDR2 DATA2	ADDR1 DATA1	ADDR0 DATA0
Reset:	Indeterminate after reset							

Figure 6-4. Port C Data Register (PORTC)

Address: $1005

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0
Reset:	Indeterminate after reset							

Figure 6-5. Port C Latched Register (PORTCL)

The control and status bits that implement the input capture functions are contained in:

- Pulse accumulator control register (PACTL)
- Timer control 2 register (TCTL2)
- Timer interrupt mask 1 register (TMSK1)
- Timer interrupt flag 2 register (TFLG1)

To configure port A bit 3 as an input capture, clear the DDRA3 bit of the PACTL register. Note that this bit is cleared out of reset. To enable PA3 as the fourth input capture, set the I4/O5 bit in the PACTL register. Otherwise, PA3 is configured as a fifth output compare out of reset, with bit I4/O5 being cleared. If the DDRA3 bit is set (configuring PA3 as an output), and IC4 is enabled, then writes to PA3 cause edges on the pin to result in input captures. Writing to TI4/O5 has no effect when the TI4/O5 register is acting as IC4.

9.3.1 Timer Control Register 2

Use the control bits of this register to program input capture functions to detect a particular edge polarity on the corresponding timer input pin. Each of the input capture functions can be independently configured to detect rising edges only, falling edges only, any edge (rising or falling), or to disable the input capture function. The input capture functions operate independently of each other and can capture the same TCNT value if the input edges are detected within the same timer count cycle.

Address: $1021

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
Reset:	0	0	0	0	0	0	0	0

Figure 9-3. Timer Control Register 2 (TCTL2)

EDGxB and EDGxA — Input Capture Edge Control Bits

There are four pairs of these bits. Each pair is cleared to 0 by reset and must be encoded to configure the corresponding input capture edge detector circuit. IC4 functions only if the I4/O5 bit in the PACTL register is set. Refer to Table 9-2 for timer control configuration.

Table 9-2. Timer Control Configuration

EDGxB	EDGxA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any edge

9.3.2 Timer Input Capture Registers

When an edge has been detected and synchronized, the 16-bit free-running counter value is transferred into the input capture register pair as a single 16-bit parallel transfer. Timer counter value captures and timer counter incrementing occur on opposite half-cycles of the phase 2 clock so that the count value is stable whenever a capture occurs. The timer input capture registers are not affected by reset. Input capture values can be read from a pair of 8-bit read-only registers. A read of the high-order byte of an

Freescale Semiconductor
Application Note

AN1060
Rev. 1.1, 07/2005

M68HC11 Bootstrap Mode

By Jim Sibigtroth
Mike Rhoades
John Langan
Austin, Texas

Introduction

The M68HC11 Family of MCUs (microcontroller units) has a bootstrap mode that allows a user-defined program to be loaded into the internal random-access memory (RAM) by way of the serial communications interface (SCI); the M68HC11 then executes this loaded program. The loaded program can do anything a normal user program can do as well as anything a factory test program can do because protected control bits are accessible in bootstrap mode. Although the bootstrap mode is a single-chip mode of operation, expanded mode resources are accessible because the mode control bits can be changed while operating in the bootstrap mode.

This application note explains the operation and application of the M68HC11 bootstrap mode. Although basic concepts associated with this mode are quite simple, the more subtle implications of these functions require careful consideration. Useful applications of this mode are overlooked due to an incomplete understanding of bootstrap mode. Also, common problems associated with bootstrap mode could be avoided by a more complete understanding of its operation and implications.

Topics discussed in this application note include:

- Basic operation of the M68HC11 bootstrap mode
- General discussion of bootstrap mode uses
- Detailed explanation of on-chip bootstrap logic
- Detailed explanation of bootstrap firmware
- Bootstrap firmware vs. EEPROM security
- Incorporating the bootstrap mode into a system
- Driving bootstrap mode from another M68HC11
- Driving bootstrap mode from a personal computer
- Common bootstrap mode problems
- Variations for specific versions of M68HC11
- Commented listings for selected M68HC11 bootstrap ROMs

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.

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

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

$FF Character Is Required before Loading into RAM

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

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

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

Variable-Length Download

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

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

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

EPROM/OTP Versions of M68HC11 Have an EPROM Emulation Mode

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

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

```
162 BF62 E72D              STAB   SCCR2,X        Rx and Tx Enabled
163 BF64 CC021B            LDD    #DELAYF        Delay for fast baud rate
164 BF67 ED16              STD    TOC1,X         Set as default delay
165
166                * Send BREAK to signal ready for download
167 BF69 1C2D01            BSET   SCCR2,X $01    Set send break bit
168 BF6C 1E0801FC          BRSET  PORTD,X $01 *  Wait for RxD pin to go low
169 BF70 1D2D01            BCLR   SCCR2,X $01    Clear send break bit
170 BF73
171 BF73 1F2E20FC          BRCLR  SCSR,X $20 *   Wait for RDRF
172 BF77 A62F              LDAA   SCDAT,X        Read data
173                * Data will be $00 if BREAK OR $00 received
174 BF79 2603              BNE    NOTZERO        Bypass JMP if not 0
175 BF7B 7EB600            JMP    EEPMSTR        Jump to EEPROM if it was 0
176 BF7E           NOTZERO EQU    *
177 BF7E 81FF              CMPA   #$FF           $FF will be seen as $FF
178 BF80 2708              BEQ    BAUDOK         If baud was correct
179                * Or else change to ÷104 (÷13 & ÷8) 1200 @ 2MHZ
180 BF82 1C2B33            BSET   BAUD,X $33     Works because $22 -> $33
181 BF85 CC0DB0            LDD    #DELAYS        And switch to slower...
182 BF88 ED16              STD    TOC1,X         delay constant
183 BF8A           BAUDOK  EQU    *
184 BF8A 18CE0000          LDY    #RAMSTR        Point at start of RAM
185
186 BF8E           WAIT    EQU    *
187 BF8E EC16              LDD    TOC1,X         Move delay constant to D
188 BF90           WTLOOP  EQU    *
189 BF90 1E2E2007          BRSET  SCSR,X $20 NEWONE  Exit loop if RDRF set
190 BF94 8F                XGDX                  Swap delay count to X
191 BF95 09                DEX                   Decrement count
192 BF96 8F                XGDX                  Swap back to D
193 BF97 26F7              BNE    WTLOOP         Loop if not timed out
194 BF99 200F              BRA    STAR           Quit download on timeout
195
196 BF9B           NEWONE  EQU    *
197 BF9B A62F              LDAA   SCDAT,X        Get received data
198 BF9D 18A700            STAA   $00,Y          Store to next RAM location
199 BFA0 A72F              STAA   SCDAT,X        Transmit it for handshake
200 BFA2 1808              INY                   Point at next RAM location
201 BFA4 188C0200          CPY    #RAMEND+1      See if past end
202 BFA8 26E4              BNE    WAIT           If not, Get another
203
204 BFAA           STAR    EQU    *
205 BFAA CE1068            LDX    #PROGDEL       Init X with programming delay
206 BFAD 18CED000          LDY    #EPRMSTR       Init Y with EPROM start addr
207 BFB1 7E0000            JMP    RAMSTR         ** EXIT to start of RAM **
208 BFB4
209                ************************************************
210                * Block fill unused bytes with zeros
211
212 BFB4 000000000000        BSZ    $BFD1-*
       000000000000
       000000000000
       000000000000
       0000000000
```