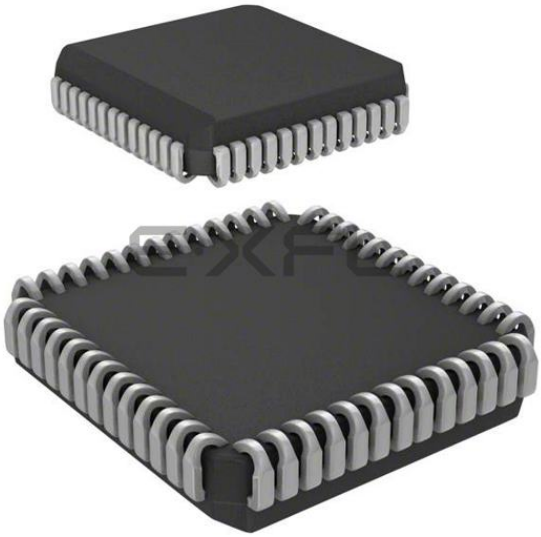

Welcome to E-XFL.COM

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	12KB (12K x 8)
Program Memory Type	OTP
EEPROM Size	512 x 8
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/mc68hc711e9cfne2

List of Sections

Section 1. General Description 23

Section 2. Pin Descriptions 27

Section 3. Central Processor Unit (CPU) 45

Section 4. Operating Modes and On-Chip Memory 65

Section 5. Resets and Interrupts 107

Section 6. Parallel Input/Output (I/O) Ports 133

Section 7. Serial Communications Interface (SCI) 145

Section 8. Serial Peripheral Interface (SPI) 165

Section 9. Timing System 177

Section 10. Analog-to-Digital (A/D) Converter 209

Section 11. Electrical Characteristics 221

Section 12. Mechanical Data 253

Section 13. Ordering Information 261

Appendix A. Development Support 269

Appendix B. EVBU Schematic 275

Section 5. Resets and Interrupts

5.1 Contents 107
5.2 Introduction 108
5.3 Resets 108
5.3.1 Power-On Reset (POR) 109
5.3.2 External Reset (RESET) 109
5.3.3 Computer Operating Properly (COP) Reset 110
5.3.4 Clock Monitor Reset 111
5.3.5 System Configuration Options Register 112
5.3.6 Configuration Control Register 113
5.4 Effects of Reset 114
5.4.1 Central Processor Unit (CPU) 115
5.4.2 Memory Map 115
5.4.3 Timer 115
5.4.4 Real-Time Interrupt (RTI) 116
5.4.5 Pulse Accumulator 116
5.4.6 Computer Operating Properly (COP) 116
5.4.7 Serial Communications Interface (SCI) 116
5.4.8 Serial Peripheral Interface (SPI) 117
5.4.9 Analog-to-Digital (A/D) Converter 117
5.4.10 System 117
5.5 Reset and Interrupt Priority 117
5.5.1 Highest Priority Interrupt and Miscellaneous Register 119
5.6 Interrupts 121
5.6.1 Interrupt Recognition and Register Stacking 122
5.6.2 Non-Maskable Interrupt Request (XIRQ) 123
5.6.3 Illegal Opcode Trap 123
5.6.4 Software Interrupt (SWI) 124
5.6.5 Maskable Interrupts 124
5.6.6 Reset and Interrupt Processing 124
5.7 Low-Power Operation 129
5.7.1 Wait Mode 130
5.7.2 Stop Mode 130

List of Figures

Figure	Title	Page
1-1	M68HC11 E-Series Block Diagram	26
2-1	Pin Assignments for 52-Pin PLCC and CLCC	28
2-2	Pin Assignments for 64-Pin QFP	29
2-3	Pin Assignments for 52-Pin TQFP	30
2-4	Pin Assignments for 56-Pin SDIP	31
2-5	Pin Assignments for 48-Pin DIP (MC68HC811E2)	32
2-6	External Reset Circuit	33
2-7	External Reset Circuit with Delay	33
2-8	Common Parallel Resonant Crystal Connections	35
2-9	External Oscillator Connections	35
3-1	Programming Model	47
3-2	Stacking Operations	49
4-1	Address/Data Demultiplexing	68
4-2	Memory Map for MC68HC11E0	70
4-3	Memory Map for MC68HC11E1	70
4-4	Memory Map for MC68HC(7)11E9	71
4-5	Memory Map for MC68HC(7)11E20	71
4-6	Memory Map for MC68HC811E2	72
4-7	Register and Control Bit Assignments	72
4-8	RAM Standby MODB/V_{STBY} Connections	81
4-9	Highest Priority I-Bit Interrupt and Miscellaneous Register (HPRIO)	83
4-10	System Configuration Register (CONFIG)	87
4-11	MC68HC811E2 System Configuration Register (CONFIG)	87
4-12	RAM and I/O Mapping Register (INIT)	89

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.

3.6 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.

3.6.1 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.

Table 3-2. Instruction Set (Sheet 3 of 7)

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
CMPB (opr)	Compare B to Memory	B – M	B IMM B DIR B EXT B IND,X B IND,Y	C1 D1 F1 E1 18 E1	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
COM (opr)	Ones Complement Memory Byte	$FF – M ⇒ M	EXT IND,X IND,Y	73 63 18 63	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	0	1
COMA	Ones Complement A	$FF – A ⇒ A	A INH	43	—	2	—	—	—	—	Δ	Δ	0	1
COMB	Ones Complement B	$FF – B ⇒ B	B INH	53	—	2	—	—	—	—	Δ	Δ	0	1
CPD (opr)	Compare D to Memory 16-Bit	D – M : M + 1	IMM DIR EXT IND,X IND,Y	1A 83 1A 93 1A B3 1A A3 CD A3	jj kk dd hh ll ff ff	5 6 7 7 7	—	—	—	—	Δ	Δ	Δ	Δ
CPX (opr)	Compare X to Memory 16-Bit	IX – M : M + 1	IMM DIR EXT IND,X IND,Y	8C 9C BC AC CD AC	jj kk dd hh ll ff ff	4 5 6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
CPY (opr)	Compare Y to Memory 16-Bit	IY – M : M + 1	IMM DIR EXT IND,X IND,Y	18 8C 18 9C 18 BC 1A AC 18 AC	jj kk dd hh ll ff ff	5 6 7 7 7	—	—	—	—	Δ	Δ	Δ	Δ
DAA	Decimal Adjust A	Adjust Sum to BCD	INH	19	—	2	—	—	—	—	Δ	Δ	Δ	Δ
DEC (opr)	Decrement Memory Byte	M – 1 ⇒ M	EXT IND,X IND,Y	7A 6A 18 6A	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	—
DECA	Decrement Accumulator A	A – 1 ⇒ A	A INH	4A	—	2	—	—	—	—	Δ	Δ	Δ	—
DECB	Decrement Accumulator B	B – 1 ⇒ B	B INH	5A	—	2	—	—	—	—	Δ	Δ	Δ	—
DES	Decrement Stack Pointer	SP – 1 ⇒ SP	INH	34	—	3	—	—	—	—	—	—	—	—
DEX	Decrement Index Register X	IX – 1 ⇒ IX	INH	09	—	3	—	—	—	—	—	Δ	—	—
DEY	Decrement Index Register Y	IY – 1 ⇒ IY	INH	18 09	—	4	—	—	—	—	—	Δ	—	—
EORA (opr)	Exclusive OR A with Memory	A ⊕ M ⇒ A	A IMM A DIR A EXT A IND,X A IND,Y	88 98 B8 A8 18 A8	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
EORB (opr)	Exclusive OR B with Memory	B ⊕ M ⇒ B	B IMM B DIR B EXT B IND,X B IND,Y	C8 D8 F8 E8 18 E8	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
FDIV	Fractional Divide 16 by 16	D / IX ⇒ IX; r ⇒ D	INH	03	—	41	—	—	—	—	—	Δ	Δ	Δ
IDIV	Integer Divide 16 by 16	D / IX ⇒ IX; r ⇒ D	INH	02	—	41	—	—	—	—	—	Δ	0	Δ
INC (opr)	Increment Memory Byte	M + 1 ⇒ M	EXT IND,X IND,Y	7C 6C 18 6C	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	—
INCA	Increment Accumulator A	A + 1 ⇒ A	A INH	4C	—	2	—	—	—	—	Δ	Δ	Δ	—

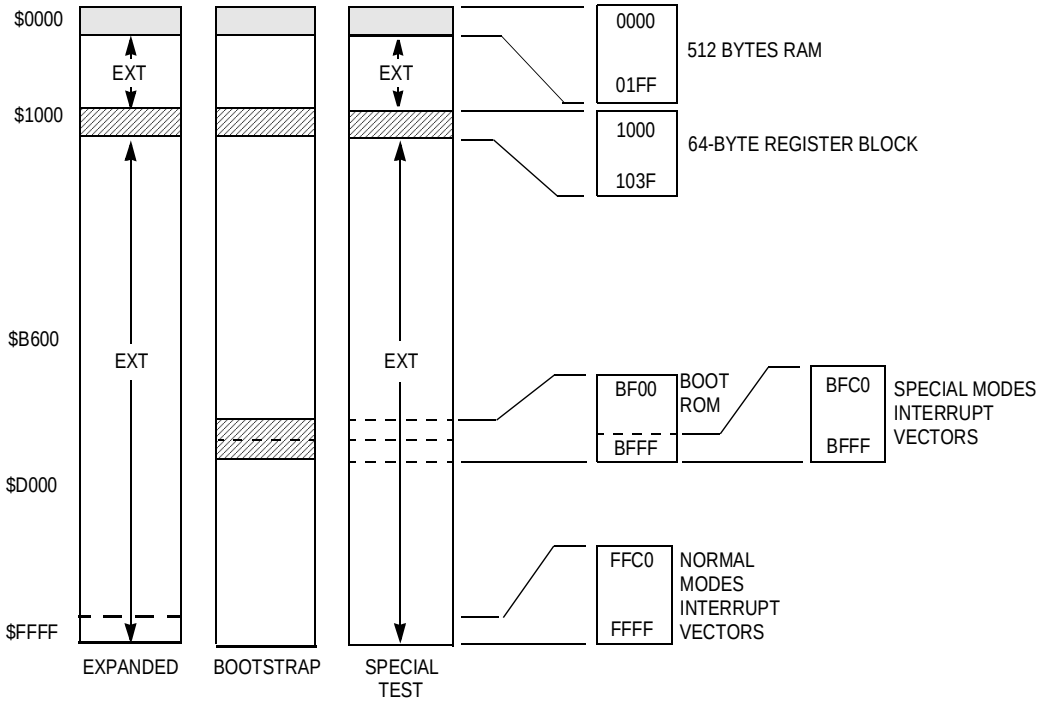

Figure 4-2. Memory Map for MC68HC11E0

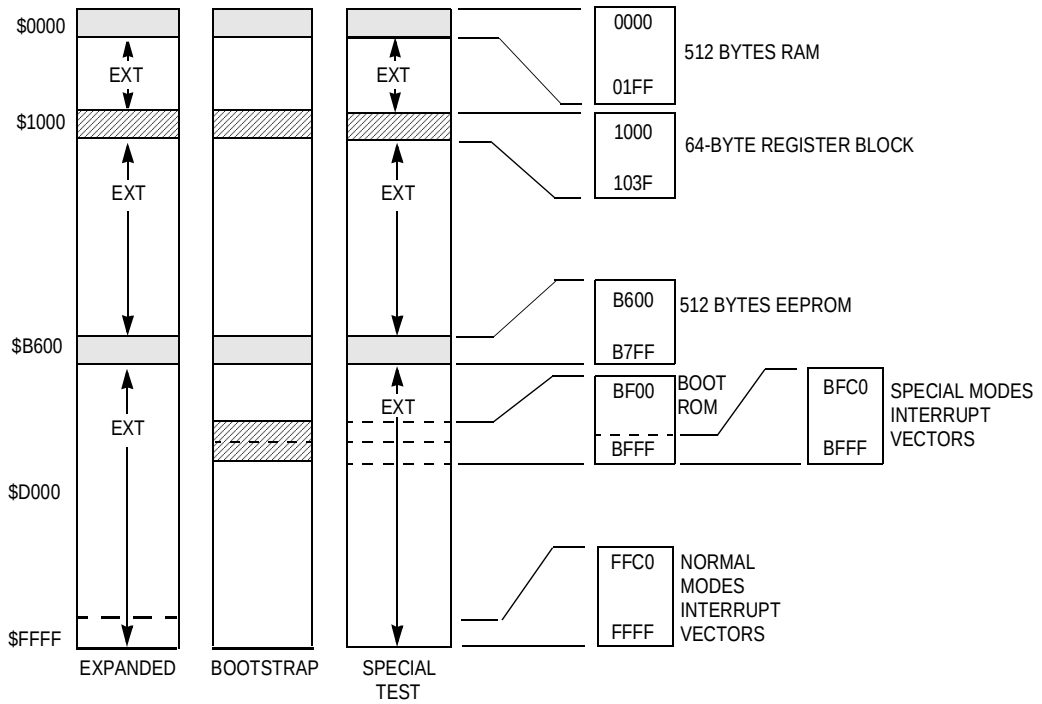

Figure 4-3. Memory Map for MC68HC11E1

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$1003	Port C Data Register (PORTC) See page 136.	Read: Write:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
		Reset:	Indeterminate after reset							
$1004	Port B Data Register (PORTB) See page 136.	Read: Write:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
		Reset:	0	0	0	0	0	0	0	0
$1005	Port C Latched Register (PORTCL) See page 137.	Read: Write:	PCL7	PCL6	PCL5	PCL4	PCL3	PCL2	PCL1	PCL0
		Reset:	Indeterminate after reset							
$1006	Reserved		R	R	R	R	R	R	R	R
$1007	Port C Data Direction Register (DDRC) See page 137.	Read: Write:	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Reset:	0	0	0	0	0	0	0	0
$1008	Port D Data Register (PORTD) See page 138.	Read: Write:	0	0	PD5	PD4	PD3	PD2	PD1	PD0
		Reset:	U	U	I	I	I	I	I	I
$1009	Port D Data Direction Register (DDRD) See page 138.	Read: Write:			DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		Reset:	0	0	0	0	0	0	0	0
$100A	Port E Data Register (PORTE) See page 139.	Read: Write:	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
		Reset:	Indeterminate after reset							
$100B	Timer Compare Force Register (CFORC) See page 190.	Read: Write:	FOC1	FOC2	FOC3	FOC4	FOC5			
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented R = Reserved U = Unaffected

I = Indeterminate after reset

Figure 4-7. Register and Control Bit Assignments (Sheet 2 of 8)

4.6.1.5 EEPROM Byte Erase

This is an example of how to erase a single byte of EEPROM.

```
BYTEE   LDAB    #$16     BYTE = 1, ERASE = 1, EELAT = 1

        STAB    $103B    Set to BYTE erase mode

        STAB    0,X      Write any data to address to be erased

        LDAB    #$17     BYTE = 1, ERASE = 1, EELAT = 1,
                         EPGM = 1

        STAB    $103B    Turn on high voltage

        JSR     DLY10    Delay 10 ms

        CLR     $103B    Turn off high voltage and set
                         to READ mode
```

4.6.1.6 CONFIG Register Programming

Because the CONFIG register is implemented with EEPROM cells, use EEPROM procedures to erase and program this register. The procedure for programming is the same as for programming a byte in the EEPROM array, except that the CONFIG register address is used. CONFIG can be programmed or erased (including byte erase) while the MCU is operating in any mode, provided that PTCON in BPROT is clear.

To change the value in the CONFIG register, complete this procedure.

1. Erase the CONFIG register.
2. Program the new value to the CONFIG address.
3. Initiate reset.

NOTE: *Do not initiate a reset until the procedure is complete.*

4.6.2 EEPROM Security

The optional security feature, available only on ROM-based MCUs, protects the EEPROM and RAM contents from unauthorized access. A program, or a key portion of a program, can be protected against unauthorized duplication. To accomplish this, the protection mechanism restricts operation of protected devices to the single-chip modes. This

5.3.3 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

6.4 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)

6.5 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)

RWU — Receiver Wakeup Control Bit

0 = Normal SCI receiver

1 = Wakeup enabled and receiver interrupts inhibited

SBK — Send Break

At least one character time of break is queued and sent each time SBK is written to 1. As long as the SBK bit is set, break characters are queued and sent. More than one break may be sent if the transmitter is idle at the time the SBK bit is toggled on and off, as the baud rate clock edge could occur between writing the 1 and writing the 0 to SBK.

0 = Break generator off

1 = Break codes generated

7.8.4 Serial Communication Status Register

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

Address: $102E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	TDRE	TC	RDRF	IDLE	OR	NF	FE	
Reset:	1	1	0	0	0	0	0	0

= Unimplemented

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

Table 9-1. Timer Summary

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

9.3 Timer Structure

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

DDRA3 — Data Direction for Port A Bit 3

Refer to **Section 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 **9.6 Real-Time Interrupt (RTI)**.

9.8.2 Pulse Accumulator Count Register

This 8-bit read/write register contains the count of external input events at the PAI input or the accumulated count. The PACNT is readable even if PAI is not active in gated time accumulation mode. The counter is not affected by reset and can be read or written at any time. Counting is synchronized to the internal PH2 clock so that incrementing and reading occur during opposite half cycles.

Address: $1027

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset:	Indeterminate after reset							

Figure 9-26. Pulse Accumulator Count Register (PACNT)

10.3 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 10-1**.

10.3.1 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 10-2**, which is a functional diagram of an input pin.

11.18 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

Appendix A. Development Support

A.1 Contents

A.2 Introduction 269
A.3 Motorola M68HC11 E-Series Development Tools 270
A.4 EVS — Evaluation System 270
A.5 Motorola Modular Development System (MMDS11) 271
A.6 SPGMR11 — Serial Programmer for M68HC11 MCUs 273

A.2 Introduction

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

UPLOAD Utility

The UPLOAD utility subroutine transfers data from the MCU to a host computer system over the SCI serial data link.

NOTE: *Only EPROM versions of the M68HC11 include this utility.*

Verification of EPROM contents is one example of how the UPLOAD utility could be used. Before calling this program, the Y index register is loaded (by user firmware) with the address of the first data byte to be uploaded. If a baud rate other than the current SCI baud rate is to be used for the upload process, the user's firmware must also write to the baud register. The UPLOAD program sends successive bytes of data out the SCI transmitter until a reset is issued (the upload loop is infinite).

For a complete commented listing example of the UPLOAD utility, refer to **Listing 3. MC68HC711E9 Bootloader ROM**.

EPROM Programming Utility

The EPROM programming utility is one way of programming data into the internal EPROM of the MC68HC711E9 MCU. An external 12-V programming power supply is required to program on-chip EPROM. The simplest way to use this utility program is to bootload a 3-byte program consisting of a single jump instruction to the start of the PROGRAM utility program ($BF00). The bootloader program sets the X and Y index registers to default values before jumping to the downloaded program (see [16] at the bottom of **Figure 3**). When the host computer sees the $FF character, data to be programmed into the EPROM is sent, starting with the character for location $D000. After the last byte to be programmed is sent to the MC68HC711E9 and the corresponding verification data is returned to the host, the programming operation is terminated by resetting the MCU.

The number of bytes to be programmed, the first address to be programmed, and the programming time can be controlled by the user if values other than the default values are desired.

Between these times, the bootloader program is executed, which changes the states of some systems and control bits:

- The SCI system is initialized and turned on (Rx and Tx).
- The SCI system has control of the PD0 and PD1 pins.
- Port D outputs are configured for wire-OR operation.
- The stack pointer is initialized to the top of RAM.
- Time has passed (two or more SCI character times).
- Timer has advanced from its reset count value.

Users also forget that bootstrap mode is a special mode. Thus, privileged control bits are accessible, and write protection for some registers is not in effect. The bootstrap ROM is in the memory map. The DISR bit in the TEST1 control register is set, which disables resets from the COP and clock monitor systems.

Since bootstrap is a special mode, these conditions can be changed by software. The bus can even be switched from single-chip mode to expanded mode to gain access to external memories and peripherals.

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.

```
150                 ************************************************
151                 * Main bootloader starts here
152                 ************************************************
153                 * RESET vector points to here
154
155 BF54           BEGIN    EQU    *
156 BF54 8E01FF             LDS    #RAMEND        Initialize stack pntr
157 BF57 CE1000             LDX    #$1000         Point at internal regs
158 BF5A 1C2820             BSET   SPCR,X $20     Select port D wire-OR mode
159 BF5D CCA20C             LDD    #$A20C         BAUD in A, SCCR2 in B
160 BF60 A72B               STAA   BAUD,X         SCPx = ÷4, SCRx = ÷4
161                 * Writing 1 to MSB of BAUD resets count chain
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
```