Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

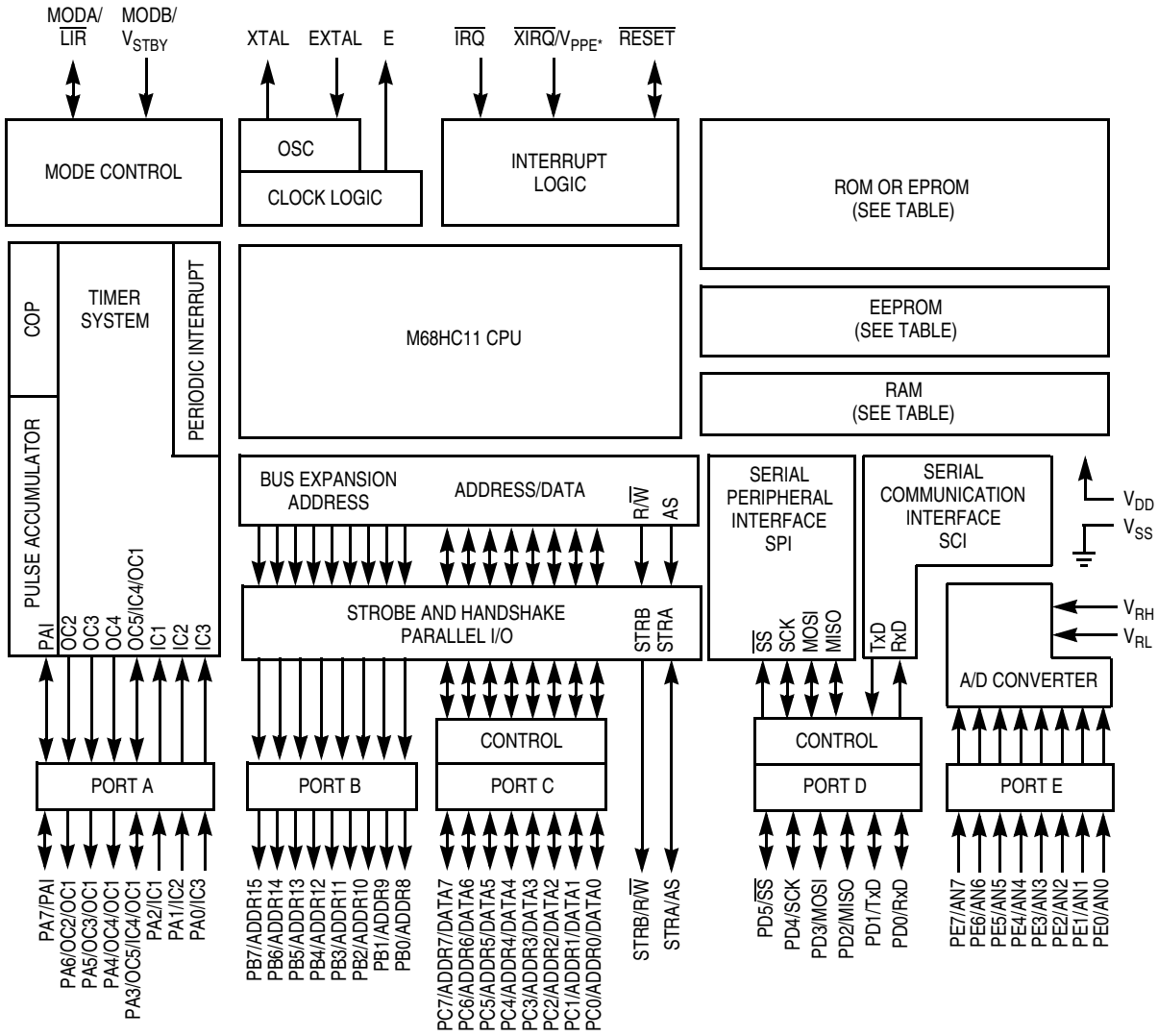
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	4MHz
Connectivity	SCI, SPI
Peripherals	POR, WDT
Number of I/O	38
Program Memory Size	20KB (20K x 8)
Program Memory Type	OTP
EEPROM Size	512 x 8
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	52-LCC (J-Lead)
Supplier Device Package	52-PLCC (19.1x19.1)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc68711e20cfne4

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



DEVICE	RAM	ROM	EPROM	EEPROM
MC68HC11E0	512	—	—	—
MC68HC11E1	512	—	—	512
MC68HC11E9	512	12 K	—	512
MC68HC711E9	512	—	12 K	512
MC68HC11E20	768	20 K	—	512
MC68HC711E20	768	—	20 K	512
MC68HC811E2	256	—	—	2048

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

Figure 1-1. M68HC11 E-Series Block Diagram

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	

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$1019	Timer Output Compare 2 Register Low (TOC2L) See page 134.	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$101A	Timer Output Compare 3 Register High (TOC3H) See page 135.	Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Reset:	1	1	1	1	1	1	1	1
$101B	Timer Output Compare 3 Register Low (TOC3L) See page 135.	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$101C	Timer Output Compare 4 Register High (TOC4H) See page 135.	Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Reset:	1	1	1	1	1	1	1	1
$101D	Timer Output Compare 4 Register Low (TOC4L) See page 135.	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$101E	Timer Input Capture 4/Output Compare 5 Register High (TI4/O5) See page 133.	Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Reset:	1	1	1	1	1	1	1	1
$101F	Timer Input Capture 4/Output Compare 5 Register Low (TI4/O5) See page 133.	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$1020	Timer Control Register 1 (TCTL1) See page 137.	Read: Write:	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5
		Reset:	0	0	0	0	0	0	0	0
$1021	Timer Control Register 2 (TCTL2) See page 131.	Read: Write:	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
		Reset:	0	0	0	0	0	0	0	0
$1022	Timer Interrupt Mask 1 Register (TMSK1) See page 138.	Read: Write:	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I
		Reset:	0	0	0	0	0	0	0	0
$1023	Timer Interrupt Flag 1 (TFLG1) See page 138.	Read: Write:	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F
		Reset:	0	0	0	0	0	0	0	0
$1024	Timer Interrupt Mask 2 Register (TMSK2) See page 139.	Read: Write:	TOI	RTII	PAOVI	PAII			PR1	PR0
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented R = Reserved U = Unaffected

I = Indeterminate after reset

Figure 2-7. Register and Control Bit Assignments (Sheet 3 of 6)

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

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

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

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

2.3.1 RAM and Input/Output Mapping

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

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

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

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

NOSEC — Security Disable Bit

NOSEC is invalid unless the security mask option is specified before the MCU is manufactured. If the security mask option is omitted NOSEC always reads 1. The enhanced security feature is available in the MC68S711E9 MCU. The enhancement to the standard security feature protects the EPROM as well as RAM and EEPROM.

0 = Security enabled
1 = Security disabled

NOCOP — COP System Disable Bit

Refer to Chapter 5 Resets and Interrupts.

1 = COP disabled
0 = COP enabled

ROMON — ROM/EPROM/OTPROM Enable Bit

When this bit is 0, the ROM or EPROM is disabled and that memory space becomes externally addressed. In single-chip mode, ROMON is forced to 1 to enable ROM/EPROM regardless of the state of the ROMON bit.

0 = ROM disabled from the memory map
1 = ROM present in the memory map

EEON — EEPROM Enable Bit

When this bit is 0, the EEPROM is disabled and that memory space becomes externally addressed.

0 = EEPROM removed from the memory map
1 = EEPROM present in the memory map

2.3.3.2 RAM and I/O Mapping Register

The internal registers used to control the operation of the MCU can be relocated on 4-Kbyte boundaries within the memory space with the use of the RAM and I/O mapping register (INIT). This 8-bit special-purpose register can change the default locations of the RAM and control registers within the MCU memory map. It can be written only once within the first 64 E-clock cycles after a reset in normal modes, and then it becomes a read-only register.

Address: $103D

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0
Reset:	0	0	0	0	0	0	0	1

Figure 2-12. RAM and I/O Mapping Register (INIT)

RAM[3:0] — RAM Map Position Bits

These four bits, which specify the upper hexadecimal digit of the RAM address, control position of RAM in the memory map. RAM can be positioned at the beginning of any 4-Kbyte page in the memory map. It is initialized to address $0000 out of reset. Refer to Table 2-4.

REG[3:0] — 64-Byte Register Block Position

These four bits specify the upper hexadecimal digit of the address for the 64-byte block of internal registers. The register block, positioned at the beginning of any 4-Kbyte page in the memory map, is initialized to address $1000 out of reset. Refer to Table 2-5.

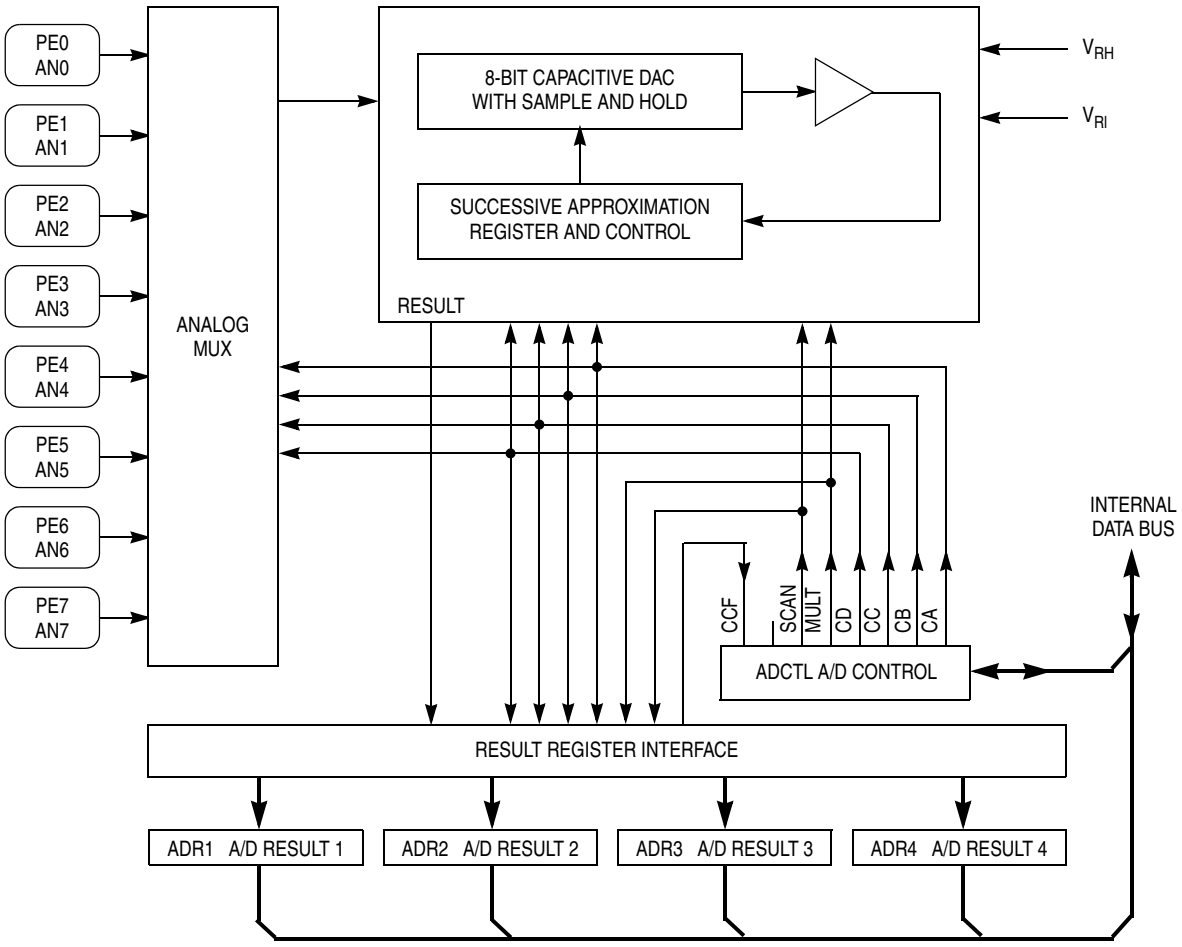
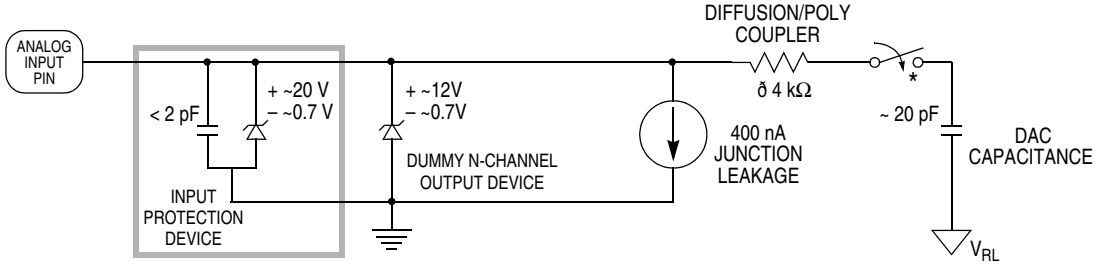


Figure 3-1. A/D Converter Block Diagram



* 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)

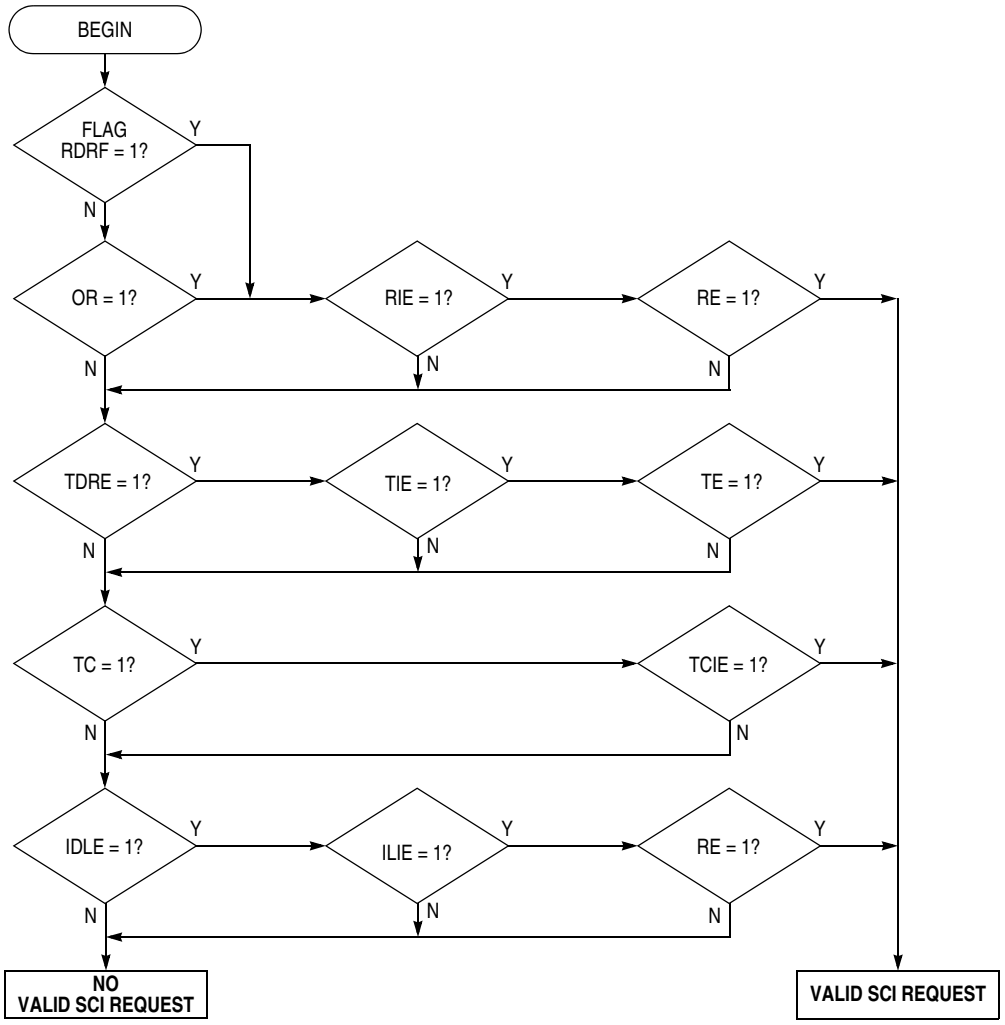


Figure 5-7. Interrupt Source Resolution Within SCI

5.6.2 Stop Mode

Executing the STOP instruction while the S bit in the CCR is equal to 0 places the MCU in stop mode. If the S bit is not 0, the stop opcode is treated as a no-op (NOP). Stop mode offers minimum power consumption because all clocks, including the crystal oscillator, are stopped while in this mode. To exit stop and resume normal processing, a logic low level must be applied to one of the external interrupts ($\overline{\text{IRQ}}$ or $\overline{\text{XIRQ}}$) or to the $\overline{\text{RESET}}$ pin. A pending edge-triggered $\overline{\text{IRQ}}$ can also bring the CPU out of stop.

Because all clocks are stopped in this mode, all internal peripheral functions also stop. The data in the internal RAM is retained as long as V_{DD} power is maintained. The CPU state and I/O pin levels are static and are unchanged by stop. Therefore, when an interrupt comes to restart the system, the MCU resumes processing as if there were no interruption. If reset is used to restart the system, a normal reset sequence results in which all I/O pins and functions are also restored to their initial states.

To use the $\overline{\text{IRQ}}$ pin as a means of recovering from stop, the I bit in the CCR must be clear ($\overline{\text{IRQ}}$ not masked). The $\overline{\text{XIRQ}}$ pin can be used to wake up the MCU from stop regardless of the state of the X bit in the CCR, although the recovery sequence depends on the state of the X bit. If X is set to 0 ($\overline{\text{XIRQ}}$ not

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.

input capture register pair inhibits a new capture transfer for one bus cycle. If a double-byte read instruction, such as load double accumulator D (LDD), is used to read the captured value, coherency is assured. When a new input capture occurs immediately after a high-order byte read, transfer is delayed for an additional cycle but the value is not lost.

Register name: Timer Input Capture 1 Register (High) Address: $1010

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	Indeterminate after reset							

Register name: Timer Input Capture 1 Register (Low) Address: $1011

	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 9-4. Timer Input Capture 1 Register Pair (TIC1)

Register name: Timer Input Capture 2 Register (High) Address: $1012

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	Indeterminate after reset							

Register name: Timer Input Capture 2 Register (Low) Address: $1013

	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 9-5. Timer Input Capture 2 Register Pair (TIC2)

Register name: Timer Input Capture 3 Register (High) Address: $1014

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	Indeterminate after reset							

Register name: Timer Input Capture 3 Register (Low) Address: $1015

	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 9-6. Timer Input Capture 3 Register Pair (TIC3)

9.3.3 Timer Input Capture 4/Output Compare 5 Register

Use TI4/O5 as either an input capture register or an output compare register, depending on the function chosen for the PA3 pin. To enable it as an input capture pin, set the I4/O5 bit in the pulse accumulator control register (PACTL) to logic level 1. To use it as an output compare register, set the I4/O5 bit to a logic level 0. Refer to 9.7 Pulse Accumulator.

Register name: Timer Input Capture 4/Output Compare 5 (High) Address: $101E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	1	1	1	1	1	1	1	1

Register name: Timer Input Capture 4/Output Compare 5 (Low) Address: $101F

	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:	1	1	1	1	1	1	1	1

Figure 9-7. Timer Input Capture 4/Output Compare 5 Register Pair (TI4/O5)

9.4 Output Compare

Use the output compare (OC) function to program an action to occur at a specific time — when the 16-bit counter reaches a specified value. For each of the five output compare functions, there is a separate 16-bit compare register and a dedicated 16-bit comparator. The value in the compare register is compared to the value of the free-running counter on every bus cycle. When the compare register matches the counter value, an output compare status flag is set. The flag can be used to initiate the automatic actions for that output compare function.

To produce a pulse of a specific duration, write a value to the output compare register that represents the time the leading edge of the pulse is to occur. The output compare circuit is configured to set the appropriate output either high or low, depending on the polarity of the pulse being produced. After a match occurs, the output compare register is reprogrammed to change the output pin back to its inactive level at the next match. A value representing the width of the pulse is added to the original value, and then written to the output compare register. Because the pin state changes occur at specific values of the free-running counter, the pulse width can be controlled accurately at the resolution of the free-running counter, independent of software latencies. To generate an output signal of a specific frequency and duty cycle, repeat this pulse-generating procedure.

The five 16-bit read/write output compare registers are: TOC1, TOC2, TOC3, and TOC4, and the TI4/O5. TI4/O5 functions under software control as either IC4 or OC5. Each of the OC registers is set to $FFFF on reset. A value written to an OC register is compared to the free-running counter value during each E-clock cycle. If a match is found, the particular output compare flag is set in timer interrupt flag register 1 (TFLG1). If that particular interrupt is enabled in the timer interrupt mask register 1 (TMSK1), an interrupt is generated. In addition to an interrupt, a specified action can be initiated at one or more timer output pins. For OC[5:2], the pin action is controlled by pairs of bits (OMx and OLx) in the TCTL1 register. The output action is taken on each successful compare, regardless of whether or not the OCxF flag in the TFLG1 register was previously cleared.

Register name: Timer Output Compare 3 Register (High) Address: $101A

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	1	1	1	1	1	1	1	1

Register name: Timer Output Compare 3 Register (Low) Address: $101B

	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:	1	1	1	1	1	1	1	1

Figure 9-10. Timer Output Compare 3 Register Pair (TOC3)

Register name: Timer Output Compare 4 Register (High) Address: $101C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset:	1	1	1	1	1	1	1	1

Register name: Timer Output Compare 4 Register (Low) Address: $101D

	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:	1	1	1	1	1	1	1	1

Figure 9-11. Timer Output Compare 4 Register Pair (TOC4)

9.4.2 Timer Compare Force Register

The CFORC register allows forced early compares. FOC[1:5] correspond to the five output compares. These bits are set for each output compare that is to be forced. The action taken as a result of a forced compare is the same as if there were a match between the OCx register and the free-running counter, except that the corresponding interrupt status flag bits are not set. The forced channels trigger their programmed pin actions to occur at the next timer count transition after the write to CFORC.

The CFORC bits should not be used on an output compare function that is programmed to toggle its output on a successful compare because a normal compare that occurs immediately before or after the force can result in an undesirable operation.

Address: $100B

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	FOC1	FOC2	FOC3	FOC4	FOC5			
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 9-12. Timer Compare Force Register (CFORC)

9.4.9 Timer Interrupt Mask 2 Register

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

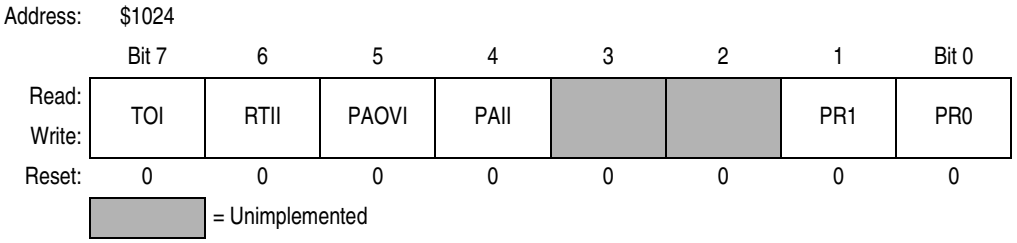


Figure 9-19. Timer Interrupt Mask 2 Register (TMSK2)

TOI — Timer Overflow Interrupt Enable Bit

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

RTII — Real-Time Interrupt Enable Bit

Refer to 9.5 Real-Time Interrupt (RTI).

PAOVI — Pulse Accumulator Overflow Interrupt Enable Bit

Refer to 9.7.3 Pulse Accumulator Status and Interrupt Bits.

PAII — Pulse Accumulator Input Edge Interrupt Enable Bit

Refer to 9.7.3 Pulse Accumulator Status and Interrupt Bits.

Bits [3:2] — Unimplemented

Always read 0

PR[1:0] — Timer Prescaler Select Bits

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

Table 9-4. Timer Prescale

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

NOTE

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

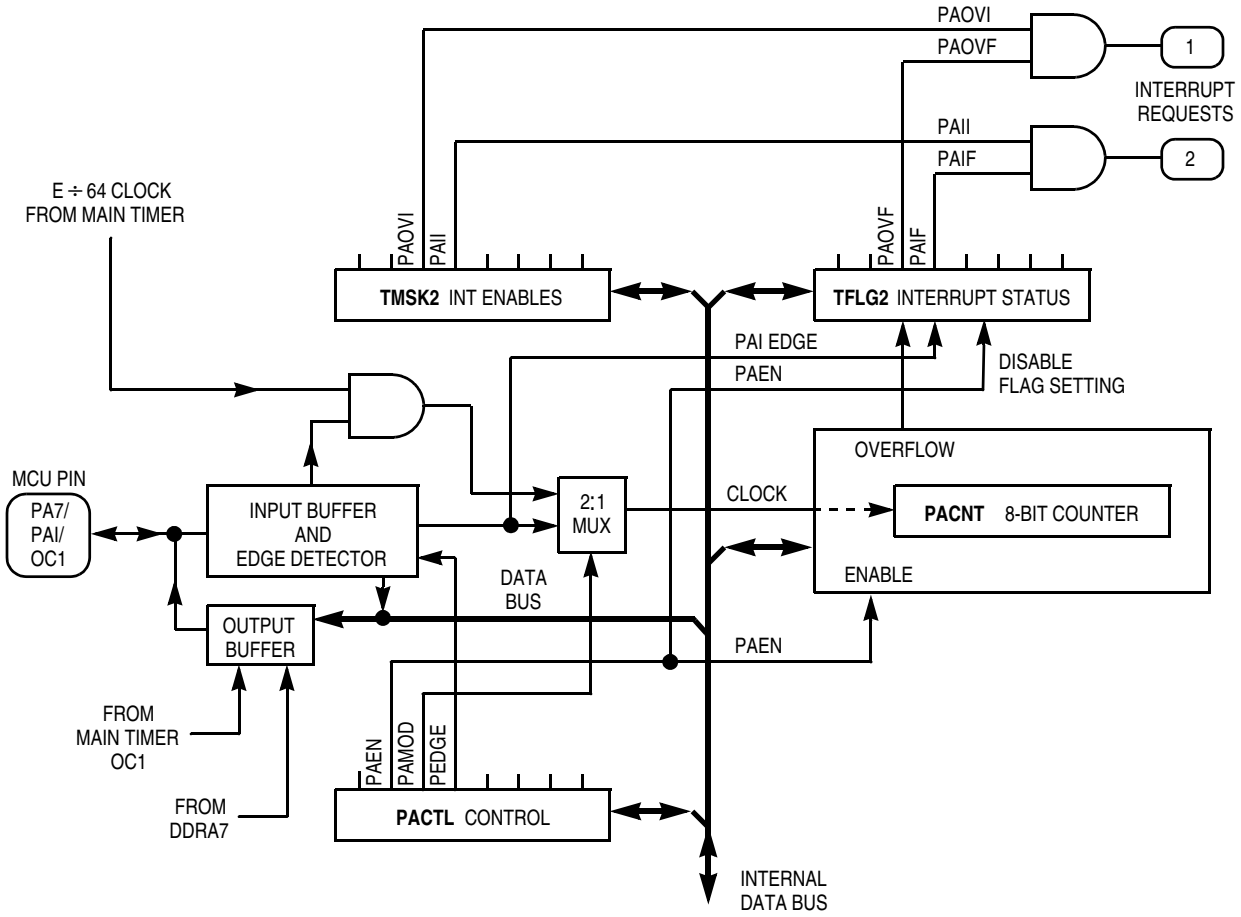


Figure 9-24. Pulse Accumulator

Figure 10-8. Port Write Timing Diagram

Figure 10-9. Simple Input Strobe Timing Diagram

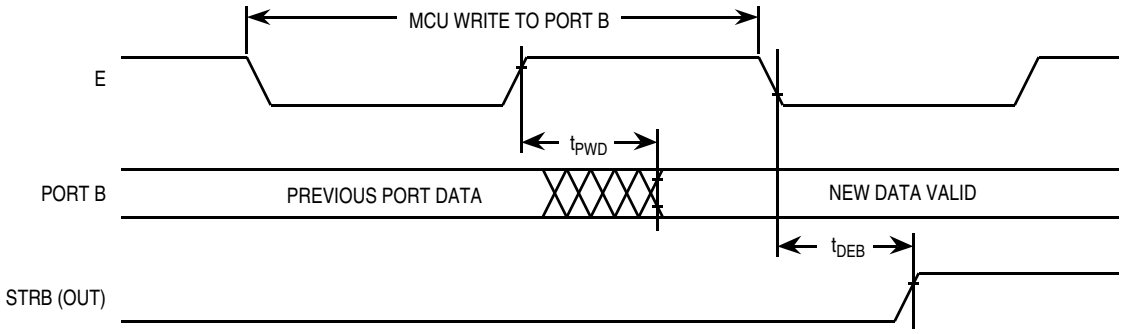
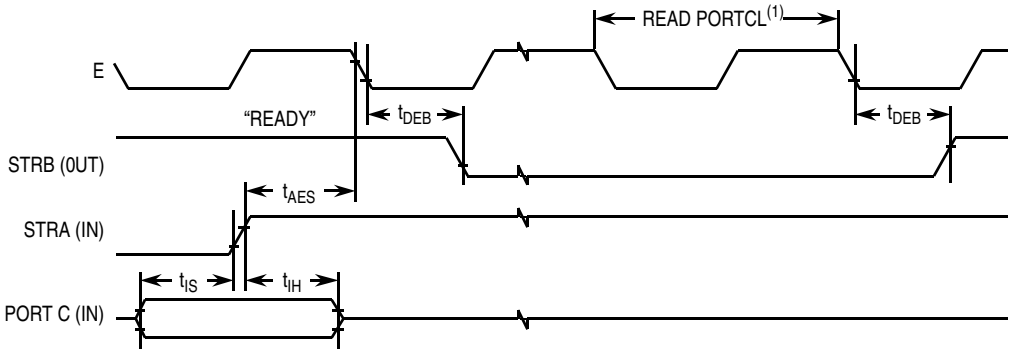


Figure 10-10. Simple Output Strobe Timing Diagram



Notes:
1. After reading PIOC with STAF set
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

Figure 10-11. Port C Input Handshake Timing Diagram

10.19 EEPROM Characteristics

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

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

10.20 MC68L11E9/E20 EEPROM Characteristics

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

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

10.21 EPROM Characteristics

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

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

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

64-pin quad flat pack (QFP)

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

64-pin quad flat pack (continued)

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

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

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

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

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

A.4 Modular Development System (MMDS11)

The M68MMDS11 modular development system (MMDS11) is an emulator system for developing embedded systems based on an M68HC11 microcontroller unit (MCU). The MMDS11 provides a bus state analyzer (BSA) and real-time memory windows. The unit's integrated development environment includes an editor, an assembler, user interface, and source-level debug. These features significantly reduce the time necessary to develop and debug an embedded MCU system. The unit's compact size requires a minimum of desk space.

The MMDS11 is one component of Freescale's modular approach to MCU-based product development. This modular approach allows easy configuration of the MMDS11 to fit a wide range of requirements. It also reduces development system cost by allowing the user to purchase only the modular components necessary to support the particular MCU derivative.

MMDS11 features include:

- Real-time, non-intrusive, in-circuit emulation at the MCU’s operating frequency
- Real-time bus state analyzer
  - 8 K x 64 real-time trace buffer
  - Display of real-time trace data as raw data, disassembled instructions, raw data and disassembled instructions, or assembly-language source code
  - Four hardware triggers for commencing trace and to provide breakpoints
  - Nine triggering modes
  - As many as 8190 pre- or post-trigger points for trace data
  - 16 general-purpose logic clips, four of which can be used to trigger the bus state analyzer sequencer
  - 16-bit time tag or an optional 24-bit time tag that reduces the logic clips traced from 16 to eight
- Four data breakpoints (hardware breakpoints)
- Hardware instruction breakpoints over either the 64-Kbyte M68HC11 memory map or over a 1-Mbyte bank switched memory map
- 32 real-time variables, nine of which can be displayed in the variables window. These variables may be read or written while the MCU is running
- 32 bytes of real-time memory can be displayed in the memory window. This memory may be read or written while the MCU is running
- 64 Kbytes of fast emulation memory (SRAM)
- Current-limited target input/output connections
- Six software-selectable oscillator clock sources: five internally generated frequencies and an external frequency via a bus analyzer logic clip
- Command and response logging to MS-DOS® disk files to save session history
- SCRIPT command for automatic execution of a sequence of MMDS11 commands
- Assembly or C-language source-level debugging with global variable viewing
- Host/emulator communications speeds as high as 57,600 baud for quick program loading

® MS-DOS is a registered trademark of Microsoft Corporation.

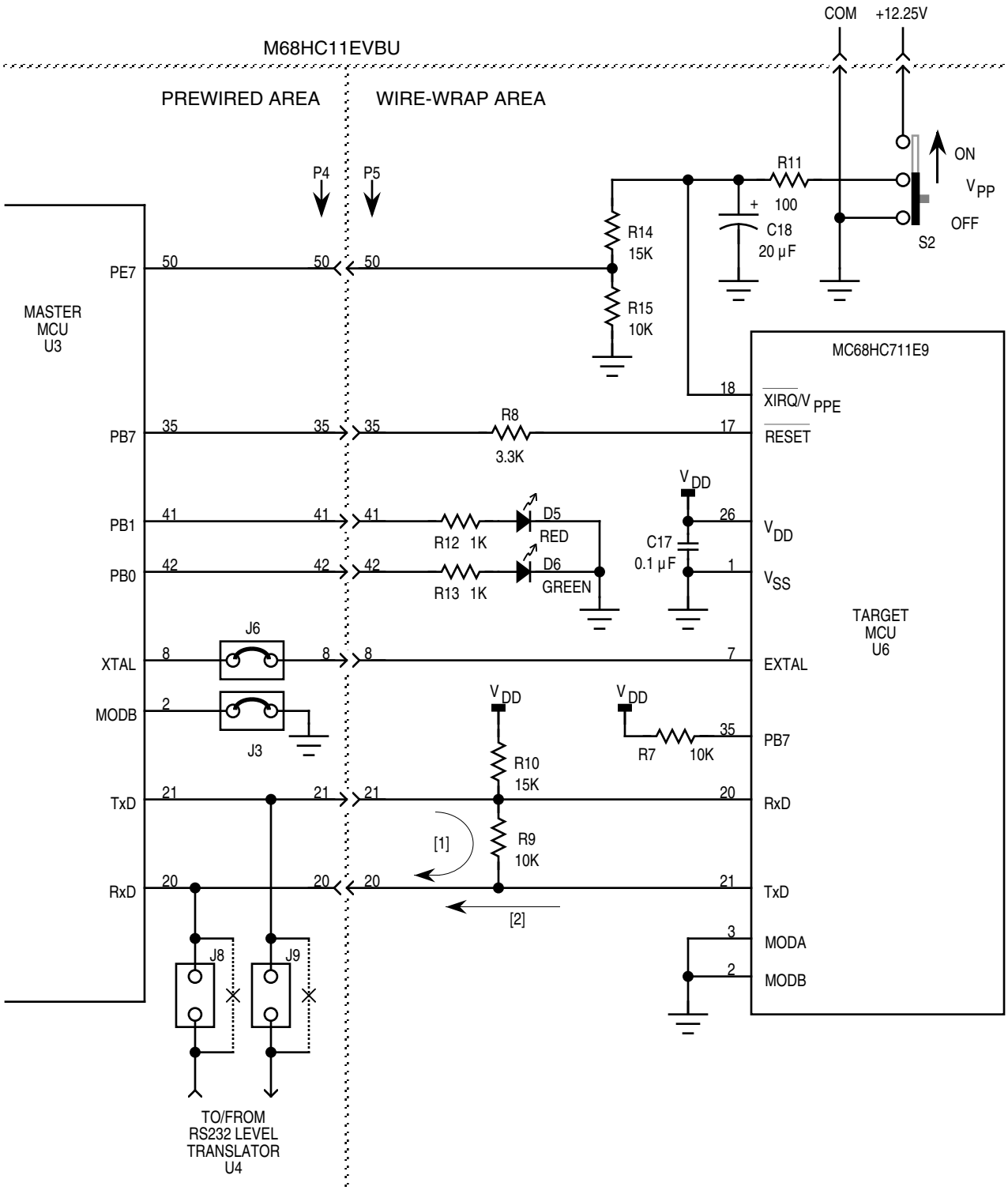


Figure 6. MCU-to-MCU EPROM Duplicator Schematic

Listing 3. MC68HC711E9 Bootloader ROM

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