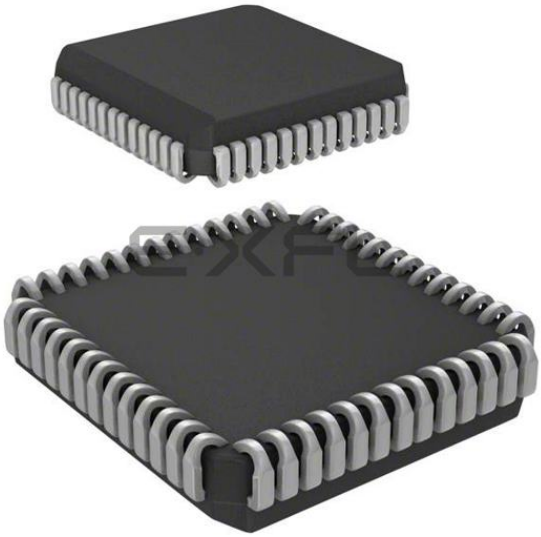


Welcome to E-XFL.COM

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	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 ~ 105°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/mc68711e20vfne2

Chapter 1
General Description

1.1 Introduction

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

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

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

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

1.2 Features

Features of the E-series devices include:

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

1.4.1 V_{DD} and V_{SS}

Power is supplied to the MCU through V_{DD} and V_{SS}. V_{DD} is the power supply, V_{SS} is ground. The MCU operates from a single 5-volt (nominal) power supply. Low-voltage devices in the E series operate at 3.0–5.5 volts.

Very fast signal transitions occur on the MCU pins. The short rise and fall times place high, short duration current demands on the power supply. To prevent noise problems, provide good power supply bypassing at the MCU. Also, use bypass capacitors that have good

high-frequency characteristics and situate them as close to the MCU as possible. Bypass requirements vary, depending on how heavily the MCU pins are loaded.

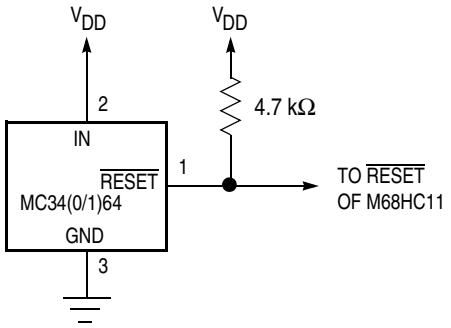


Figure 1-7. External Reset Circuit

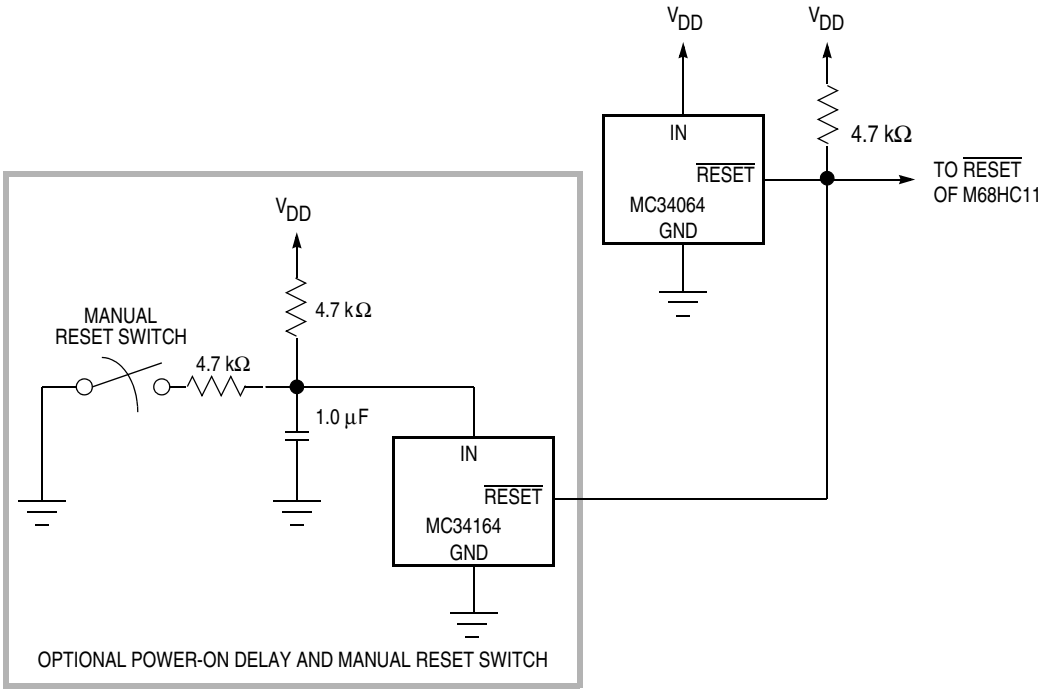


Figure 1-8. External Reset Circuit with Delay

Chapter 2
Operating Modes and On-Chip Memory

2.1 Introduction

This section contains information about the operating modes and the on-chip memory for M68HC11 E-series MCUs. Except for a few minor differences, operation is identical for all devices in the E series. Differences are noted where necessary.

2.2 Operating Modes

The values of the mode select inputs MODB and MODA during reset determine the operating mode. Single-chip and expanded multiplexed are the normal modes.

- In single-chip mode only on-chip memory is available.
- Expanded mode, however, allows access to external memory.

Each of the two normal modes is paired with a special mode:

- Bootstrap, a variation of the single-chip mode, is a special mode that executes a bootloader program in an internal bootstrap ROM.
- Test is a special mode that allows privileged access to internal resources.

2.2.1 Single-Chip Mode

In single-chip mode, ports B and C and strobe pins A (STRA) and B (STRB) are available for general-purpose parallel input/output (I/O). In this mode, all software needed to control the MCU is contained in internal resources. If present, read-only memory (ROM) and/or erasable, programmable read-only memory (EPROM) will always be enabled out of reset, ensuring that the reset and interrupt vectors will be available at locations $FFC0–$FFFF.

NOTE

For the MC68HC811E2, the vector locations are the same; however, they are contained in the 2048-byte EEPROM array.

2.2.2 Expanded Mode

In expanded operating mode, the MCU can access the full 64-Kbyte address space. The space includes:

- The same on-chip memory addresses used for single-chip mode
- Addresses for external peripherals and memory devices

The expansion bus is made up of ports B and C, and control signals AS (address strobe) and R/$\overline{W}$ (read/write). R/$\overline{W}$ and AS allow the low-order address and the 8-bit data bus to be multiplexed on the same pins. During the first half of each bus cycle address information is present. During the second half of each bus cycle the pins become the bidirectional data bus. AS is an active-high latch enable signal for an external address latch. Address information is allowed through the transparent latch while AS is high and is latched when AS drives low.

The address, R/$\overline{W}$, and AS signals are active and valid for all bus cycles, including accesses to internal memory locations. The E clock is used to enable external devices to drive data onto the internal data bus during the second half of a read bus cycle (E clock high). R/$\overline{W}$ controls the direction of data transfers. R/$\overline{W}$ drives low when data is being written to the internal data bus. R/$\overline{W}$ will remain low during consecutive data bus write cycles, such as when a double-byte store occurs.

Refer to Figure 2-1.

NOTE

The write enable signal for an external memory is the NAND of the E clock and the inverted R/$\overline{W}$ signal.

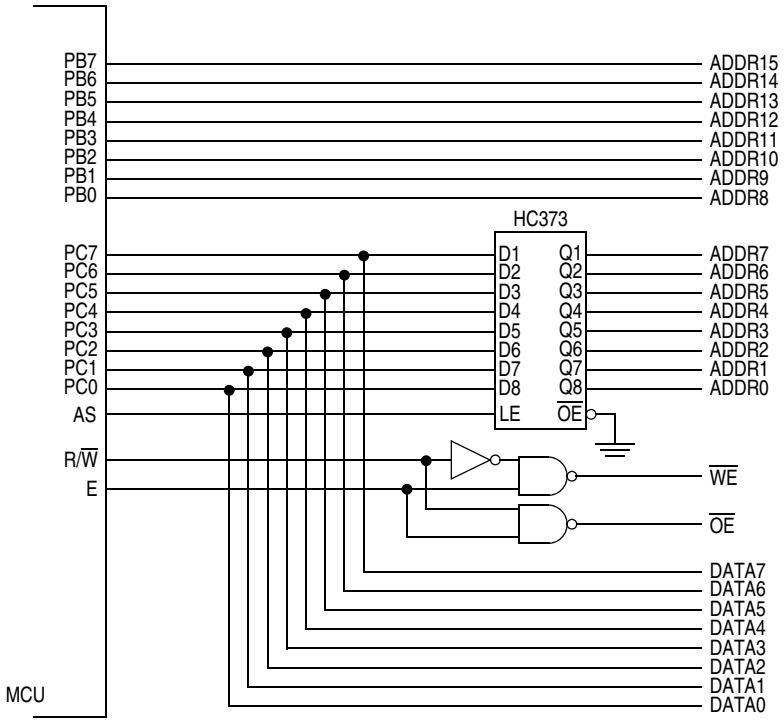


Figure 2-1. Address/Data Demultiplexing

2.2.3 Test Mode

Test mode, a variation of the expanded mode, is primarily used during Freescale's internal production testing; however, it is accessible for programming the configuration (CONFIG) register, programming calibration data into electrically erasable, programmable read-only memory (EEPROM), and supporting emulation and debugging during development.

2.2.4 Bootstrap Mode

When the MCU is reset in special bootstrap mode, a small on-chip read-only memory (ROM) is enabled at address $BF00–$BFFF. The ROM contains a bootloader program and a special set of interrupt and reset vectors. The MCU fetches the reset vector, then executes the bootloader.

Bootstrap mode is a special variation of the single-chip mode. Bootstrap mode allows special-purpose programs to be entered into internal random-access memory (RAM). When bootstrap mode is selected at reset, a small bootstrap ROM becomes present in the memory map. Reset and interrupt vectors are

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$1025	Timer Interrupt Flag 2 (TFLG2) See page 142.	Read:	TOF	RTIF	PAOVF	PAIF				
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$1026	Pulse Accumulator Control Register (PACTL) See page 142.	Read:	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$1027	Pulse Accumulator Count Register (PACNT) See page 146.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$1028	Serial Peripheral Control Register (SPCR) See page 123.	Read:	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0
		Write:								
		Reset:	0	0	0	0	0	1	U	U
$1029	Serial Peripheral Status Register (SPSR) See page 124.	Read:	SPIF	WCOL		MODF				
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$102A	Serial Peripheral Data I/O Register (SPDR) See page 125.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$102B	Baud Rate Register (BAUD) See page 113.	Read:	TCLR	SCP2[1]	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0
		Write:								
		Reset:	0	0	0	0	0	U	U	U
$102C	Serial Communications Control Register 1 (SCCR1) See page 110.	Read:	R8	T8		M	WAKE			
		Write:								
		Reset:	I	I	0	0	0	0	0	0
$102D	Serial Communications Control Register 2 (SCCR2) See page 111.	Read:	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$102E	Serial Communications Status Register (SCSR) See page 112.	Read:	TDRE	TC	RDRF	IDLE	OR	NF	FE	
		Write:								
		Reset:	1	1	0	0	0	0	0	0

1. SCP2 adds ÷39 to SCI prescaler and is present only in MC68HC(7)11E20.

$102F	Serial Communications Data Register (SCDR) See page 110.	Read:	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0
		Write:								
		Reset:	Indeterminate after reset							
$1030	Analog-to-Digital Control Status Register (ADCTL) See page 62.	Read:	CCF		SCAN	MULT	CD	CC	CB	CA
		Write:								
		Reset:	0	0	Indeterminate after reset					

= Unimplemented R = Reserved U = Unaffected

I = Indeterminate after reset

Figure 2-7. Register and Control Bit Assignments (Sheet 4 of 6)

Table 2-4. RAM Mapping

RAM[3:0]	Address
0000	$0000–$0xFF
0001	$1000–$1xFF
0010	$2000–$2xFF
0011	$3000–$3xFF
0100	$4000–$4xFF
0101	$5000–$5xFF
0110	$6000–$6xFF
0111	$7000–$7xFF
1000	$8000–$8xFF
1001	$9000–$9xFF
1010	$A000–$AxFF
1011	$B000–$BxFF
1100	$C000–$CxFF
1101	$D000–$DxFF
1110	$E000–$ExFF
1111	$F000–$FxFF

Table 2-5. Register Mapping

REG[3:0]	Address
0000	$0000–$003F
0001	$1000–$103F
0010	$2000–$203F
0011	$3000–$303F
0100	$4000–$403F
0101	$5000–$503F
0110	$6000–$603F
0111	$7000–$703F
1000	$8000–$803F
1001	$9000–$903F
1010	$A000–$A03F
1011	$B000–$B03F
1100	$C000–$C03F
1101	$D000–$D03F
1110	$E000–$E03F
1111	$F000–$F03F

2.3.3.3 System Configuration Options Register

The 8-bit, special-purpose system configuration options register (OPTION) sets internal system configuration options during initialization. The time protected control bits, IRQE, DLY, and CR[1:0], can be written only once after a reset and then they become read-only. This minimizes the possibility of any accidental changes to the system configuration.

Address: $1039

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

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

= Unimplemented

Figure 2-13. 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

Selects alternate clock source for on-chip EEPROM charge pump. Refer to 2.5.1 EEPROM and CONFIG Programming and Erasure for more information on EEPROM use.

CSEL also selects the clock source for the A/D converter, a function discussed in Chapter 3 Analog-to-Digital (A/D) Converter.

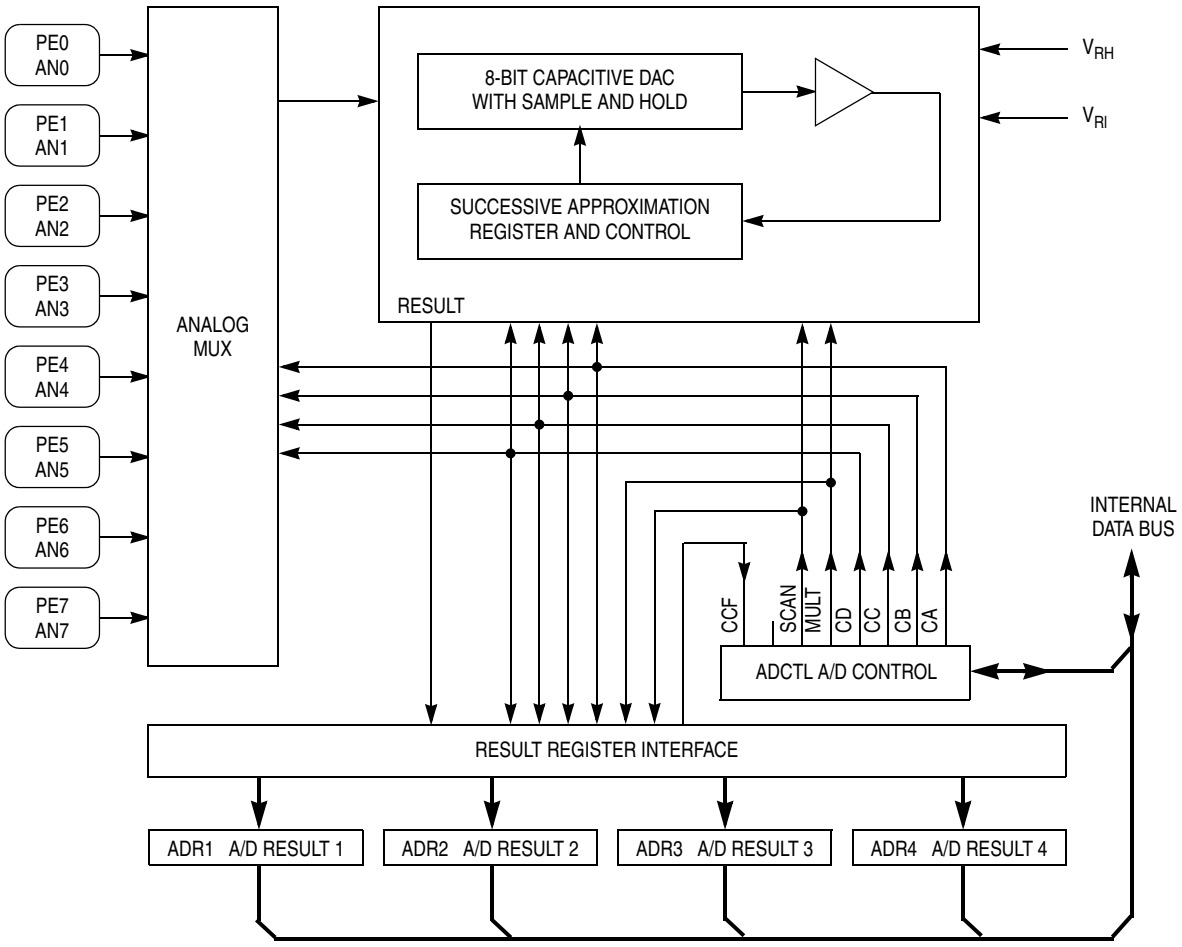
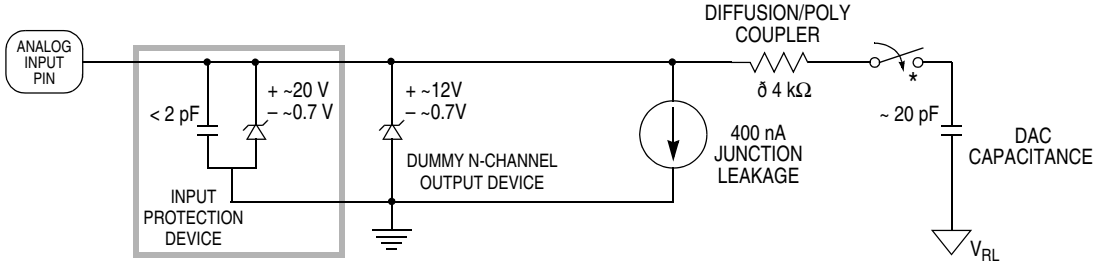


Figure 3-1. A/D Converter Block Diagram



* 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)

3.3 A/D Converter Power-Up and Clock Select

Bit 7 of the OPTION register controls A/D converter power-up. Clearing ADPU removes power from and disables the A/D converter system. Setting ADPU enables the A/D converter system. Stabilization of the analog bias voltages requires a delay of as much as 100 µs after turning on the A/D converter. When the A/D converter system is operating with the MCU E clock, all switching and comparator operations are inherently synchronized to the main MCU clocks. This allows the comparator output to be sampled at relatively quiet times during MCU clock cycles. Since the internal RC oscillator is asynchronous to the MCU clock, there is more error attributable to internal system clock noise. A/D converter accuracy is reduced slightly while the internal RC oscillator is being used (CSEL = 1).

Address: $1039

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

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

= Unimplemented

Figure 3-4. System Configuration Options Register (OPTION)

ADPU — A/D Power-Up Bit

0 = A/D powered down
1 = A/D powered up

CSEL — Clock Select Bit

0 = A/D and EEPROM use system E clock.
1 = A/D and EEPROM use internal RC clock.

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

Refer to Chapter 5 Resets and Interrupts.

DLY — Enable Oscillator Startup Delay Bit

0 = The oscillator startup delay coming out of stop is bypassed and the MCU resumes processing within about four bus cycles.
1 = A delay of approximately 4000 E-clock cycles is imposed as the MCU is started up from the stop power-saving mode. This delay allows the crystal oscillator to stabilize.

CME — Clock Monitor Enable Bit

Refer to Chapter 5 Resets and Interrupts.

Bit 2 — Not implemented

Always reads 0

CR[1:0] — COP Timer Rate Select Bits

Refer to Chapter 5 Resets and Interrupts and Chapter 9 Timing Systems.

At the end of the interrupt service routine, an return-from interrupt (RTI) instruction is executed. The RTI instruction causes the saved registers to be pulled off the stack in reverse order. Program execution resumes at the return address.

Certain instructions push and pull the A and B accumulators and the X and Y index registers and are often used to preserve program context. For example, pushing accumulator A onto the stack when entering a subroutine that uses accumulator A and then pulling accumulator A off the stack just before leaving the subroutine ensures that the contents of a register will be the same after returning from the subroutine as it was before starting the subroutine.

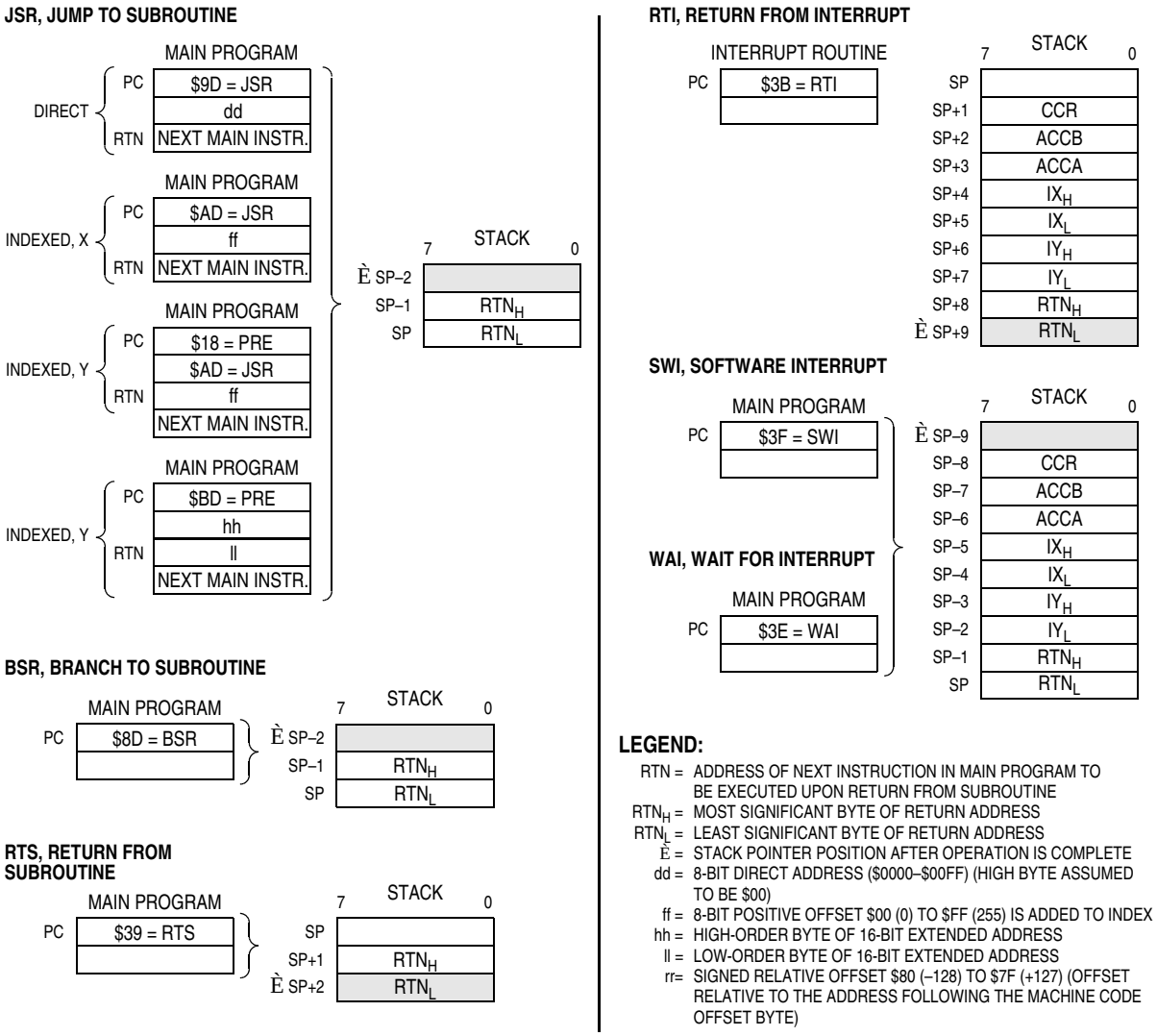
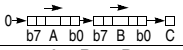
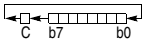
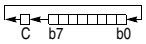
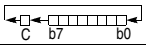
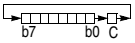
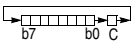
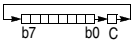


Figure 4-2. Stacking Operations

Table 4-2. Instruction Set (Sheet 5 of 7)

Mnemonic	Operation	Description	Addressing Mode	Instruction			Condition Codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
LSRD	Logical Shift Right Double		INH	04	—	3	—	—	—	—	0	Δ	Δ	Δ
MUL	Multiply 8 by 8	A * B ⇒ D	INH	3D	—	10	—	—	—	—	—	—	—	Δ
NEG (opr)	Two's Complement Memory Byte	0 – M ⇒ M	EXT IND,X IND,Y	70 60 18 60	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
NEGA	Two's Complement A	0 – A ⇒ A	A INH	40	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NEGB	Two's Complement B	0 – B ⇒ B	B INH	50	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NOP	No operation	No Operation	INH	01	—	2	—	—	—	—	—	—	—	—
ORAA (opr)	OR Accumulator A (Inclusive)	A + M ⇒ A	A IMM A DIR A EXT A IND,X A IND,Y	8A 9A BA AA 18 AA	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
ORAB (opr)	OR Accumulator B (Inclusive)	B + M ⇒ B	B IMM B DIR B EXT B IND,X B IND,Y	CA DA FA EA 18 EA	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
PSHA	Push A onto Stack	A ⇒ Stk,SP = SP – 1	A INH	36	—	3	—	—	—	—	—	—	—	—
PSHB	Push B onto Stack	B ⇒ Stk,SP = SP – 1	B INH	37	—	3	—	—	—	—	—	—	—	—
PSHX	Push X onto Stack (Lo First)	IX ⇒ Stk,SP = SP – 2	INH	3C	—	4	—	—	—	—	—	—	—	—
PSHY	Push Y onto Stack (Lo First)	IY ⇒ Stk,SP = SP – 2	INH	18 3C	—	5	—	—	—	—	—	—	—	—
PULA	Pull A from Stack	SP = SP + 1, A ⇐ Stk	A INH	32	—	4	—	—	—	—	—	—	—	—
PULB	Pull B from Stack	SP = SP + 1, B ⇐ Stk	B INH	33	—	4	—	—	—	—	—	—	—	—
PULX	Pull X From Stack (Hi First)	SP = SP + 2, IX ⇐ Stk	INH	38	—	5	—	—	—	—	—	—	—	—
PULY	Pull Y from Stack (Hi First)	SP = SP + 2, IY ⇐ Stk	INH	18 38	—	6	—	—	—	—	—	—	—	—
ROL (opr)	Rotate Left		EXT IND,X IND,Y	79 69 18 69	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
ROLA	Rotate Left A		A INH	49	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROLB	Rotate Left B		B INH	59	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ROR (opr)	Rotate Right		EXT IND,X IND,Y	76 66 18 66	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
RORA	Rotate Right A		A INH	46	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RORB	Rotate Right B		B INH	56	—	2	—	—	—	—	Δ	Δ	Δ	Δ
RTI	Return from Interrupt	See Figure 3–2	INH	3B	—	12	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ
RTS	Return from Subroutine	See Figure 3–2	INH	39	—	5	—	—	—	—	—	—	—	—
SBA	Subtract B from A	A – B ⇒ A	INH	10	—	2	—	—	—	—	Δ	Δ	Δ	Δ

Chapter 7
Serial Communications Interface (SCI)

7.1 Introduction

The serial communications interface (SCI) is a universal asynchronous receiver transmitter (UART), one of two independent serial input/output (I/O) subsystems in the M68HC11 E series of microcontrollers. It has a standard non-return-to-zero (NRZ) format (one start bit , eight or nine data bits, and one stop bit). Several baud rates are available. The SCI transmitter and receiver are independent, but use the same data format and bit rate.

All members of the E series contain the same SCI, with one exception. The SCI system in the MC68HC11E20 and MC68HC711E20 MCUs have an enhanced SCI baud rate generator. A divide-by-39 stage has been added that is enabled by an extra bit in the BAUD register. This increases the available SCI baud rate selections. Refer to Figure 7-8 and 7.7.5 Baud Rate Register.

7.2 Data Format

The serial data format requires these conditions:

1. An idle line in the high state before transmission or reception of a message
2. A start bit, logic 0, transmitted or received, that indicates the start of each character
3. Data that is transmitted and received least significant bit (LSB) first
4. A stop bit, logic 1, used to indicate the end of a frame. A frame consists of a start bit, a character of eight or nine data bits, and a stop bit.
5. A break, defined as the transmission or reception of a logic 0 for some multiple number of frames

Selection of the word length is controlled by the M bit of SCI control register (SCCR1).

7.3 Transmit Operation

The SCI transmitter includes a parallel transmit data register (SCDR) and a serial shift register. The contents of the serial shift register can be written only through the SCDR. This double buffered operation allows a character to be shifted out serially while another character is waiting in the SCDR to be transferred into the

serial shift register. The output of the serial shift register is applied to TxD as long as transmission is in progress or the transmit enable (TE) bit of serial communication control register 2 (SCCR2) is set. The block diagram, Figure 7-1, shows the transmit serial shift register and the buffer logic at the top of the figure.

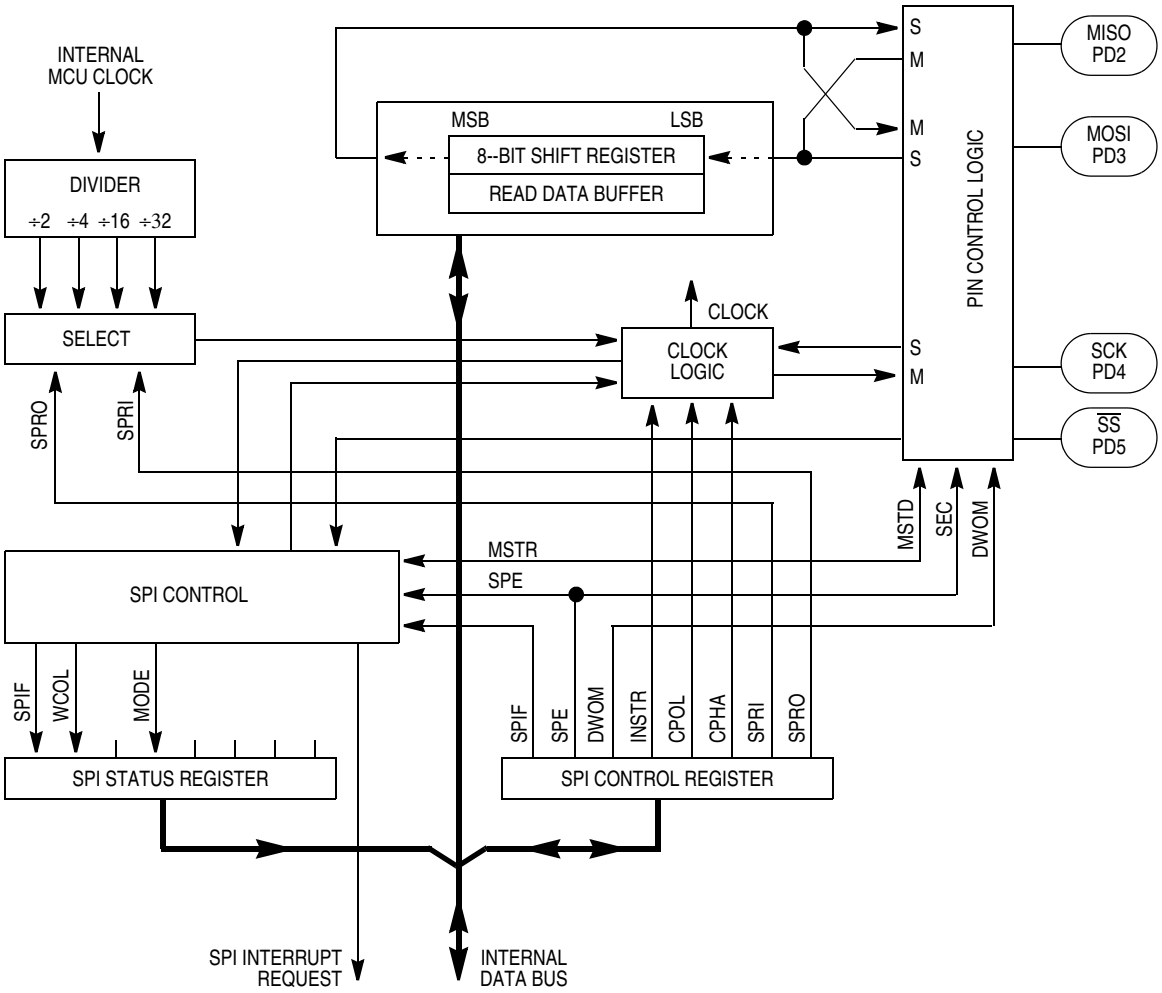


Figure 8-1. SPI Block Diagram

8.4 Clock Phase and Polarity Controls

Software can select one of four combinations of serial clock phase and polarity using two bits in the SPI control register (SPCR). The clock polarity is specified by the CPOL control bit, which selects an active high or active low clock, and has no significant effect on the transfer format. The clock phase (CPHA) control bit selects one of two different transfer formats. The clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transfers to allow a master device to communicate with peripheral slaves having different requirements.

When CPHA equals 0, the $\overline{SS}$ line must be negated and reasserted between each successive serial byte. Also, if the slave writes data to the SPI data register (SPDR) while $\overline{SS}$ is low, a write collision error results.

When CPHA equals 1, the $\overline{SS}$ line can remain low between successive transfers.

Bit 5 — Unimplemented

Always reads 0

MODF — Mode Fault Bit

To clear the MODF bit, read the SPSR (with MODF set), then write to the SPCR. Refer to 8.5.4 Slave Select and 8.6 SPI System Errors.

0 = No mode fault
1 = Mode fault

Bits [3:0] — Unimplemented

Always read 0

8.7.3 Serial Peripheral Data I/O Register

The SPDR is used when transmitting or receiving data on the serial bus. Only a write to this register initiates transmission or reception of a byte, and this only occurs in the master device. At the completion of transferring a byte of data, the SPIF status bit is set in both the master and slave devices.

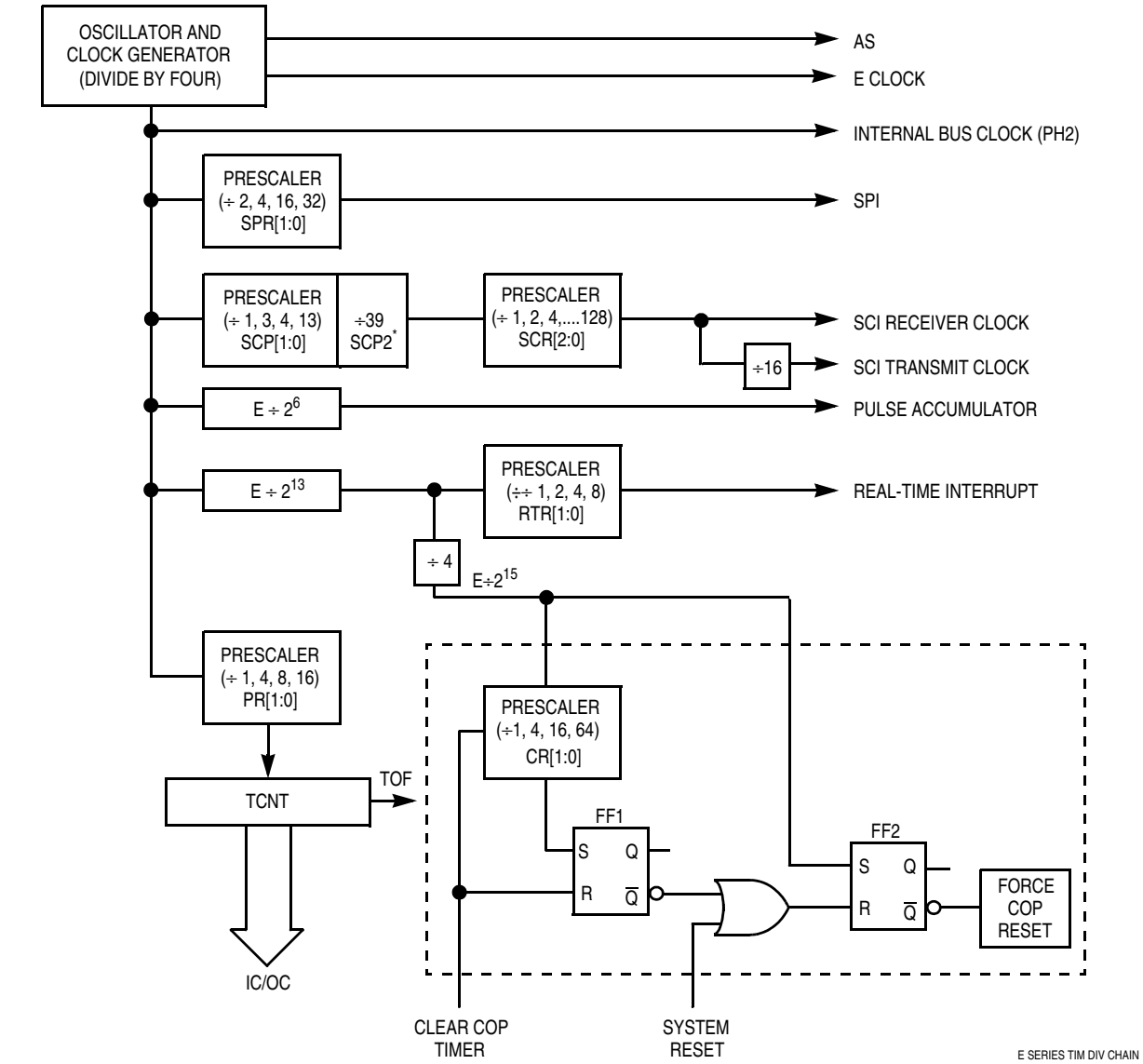
A read of the SPDR is actually a read of a buffer. To prevent an overrun and the loss of the byte that caused the overrun, the first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated.

Address: $102A

	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 8-5. Serial Peripheral Data I/O Register (SPDR)

SPI is double buffered in and single buffered out.



* SCP2 present on MC68HC(7)11E20 only

Figure 9-1. Timer Clock Divider Chains

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

FOC[1:5] — Force Output Comparison Bit

When the FOC bit associated with an output compare circuit is set, the output compare circuit immediately performs the action it is programmed to do when an output match occurs.

0 = Not affected
1 = Output x action occurs

Bits [2:0] — Unimplemented

Always read 0

9.4.3 Output Compare Mask Register

Use OC1M with OC1 to specify the bits of port A that are affected by a successful OC1 compare. The bits of the OC1M register correspond to PA[7:3].

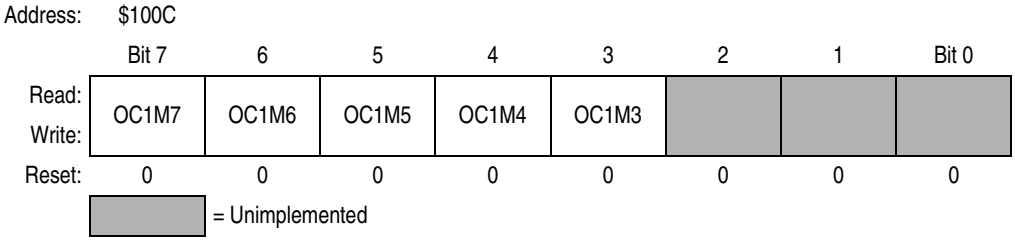


Figure 9-13. Output Compare 1 Mask Register (OC1M)

OC1M[7:3] — Output Compare Masks

0 = OC1 disabled
1 = OC1 enabled to control the corresponding pin of port A

Bits [2:0] — Unimplemented

Always read 0

9.4.4 Output Compare Data Register

Use this register with OC1 to specify the data that is to be stored on the affected pin of port A after a successful OC1 compare. When a successful OC1 compare occurs, a data bit in OC1D is stored in the corresponding bit of port A for each bit that is set in OC1M.

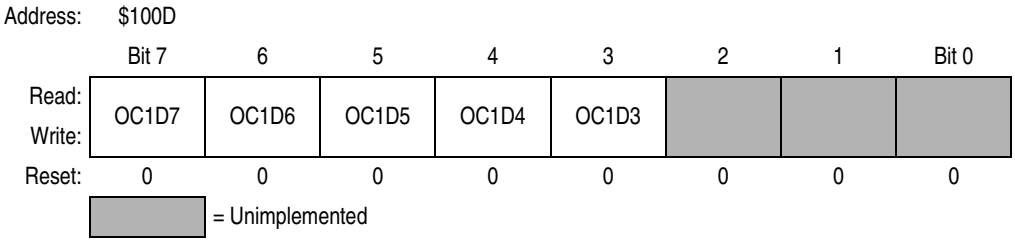


Figure 9-14. Output Compare 1 Data Register (OC1D)

If OC1Mx is set, data in OC1Dx is output to port A bit x on successful OC1 compares.

Bits [2:0] — Unimplemented

Always read 0

9.5.2 Timer Interrupt Flag Register 2

Bits of this register indicate the occurrence of timer system events. Coupled with the four high-order bits of TMSK2, the bits of TFLG2 allow the timer subsystem to operate in either a polled or interrupt driven system. Each bit of TFLG2 corresponds to a bit in TMSK2 in the same position.

Address: $1025

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	TOF	RTIF	PAOVF	PAIF				
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 9-22. Timer Interrupt Flag 2 Register (TFLG2)

Clear flags by writing a 1 to the corresponding bit position(s).

TOF — Timer Overflow Interrupt Flag
Set when TCNT changes from $FFFF to $0000

RTIF — Real-Time Interrupt Flag
The RTIF status bit is automatically set to 1 at the end of every RTI period. To clear RTIF, write a byte to TFLG2 with bit 6 set.

PAOVF — Pulse Accumulator Overflow Interrupt Flag
Refer to 9.7 Pulse Accumulator.

PAIF — Pulse Accumulator Input Edge Interrupt Flag
Refer to 9.7 Pulse Accumulator.

Bits [3:0] — Unimplemented
Always read 0

9.5.3 Pulse Accumulator Control Register

Bits RTR[1:0] of this register select the rate for the RTI system. The remaining bits control the pulse accumulator and IC4/OC5 functions.

Address: $1026

	Bit 7	6	5	4	3	2	1	Bit 0
Read: / Write:	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0
Reset:	0	0	0	0	0	0	0	0

Figure 9-23. Pulse Accumulator Control Register (PACTL)

DDRA7 — Data Direction for Port A Bit 7
Refer to Chapter 6 Parallel Input/Output (I/O) Ports.

PAEN — Pulse Accumulator System Enable Bit
Refer to 9.7 Pulse Accumulator.

PAMOD — Pulse Accumulator Mode Bit
Refer to 9.7 Pulse Accumulator.

- Extensive on-line MCU information via the CHIPINFO command. View memory map, vectors, register, and pinout information pertaining to the device being emulated
- Host software supports:
  – An editor
  – An assembler and user interface
  – Source-level debug
  – Bus state analysis
  – IBM® mouse

A.5 SPGMR11 — Serial Programmer for M68HC11 MCUs

The SPGMR11 is a modular EPROM/EEPROM programming tool for all M68HC11 devices. The programmer features interchangeable adapters that allow programming of various M68HC11 package types.

Programmer features include:

- Programs M68HC11 Family devices that contain an EPROM or EEPROM array.
- Can be operated as a stand-alone programmer connected to a host computer or connected between a host computer and the M68HC11 modular development system (MMDS11) station module
- Uses plug-in programming adapters to accommodate a variety of MCU devices and packages
- On-board programming voltage circuit eliminates the need for an external 12-volt supply.
- Includes programming software and a user's manual
- Includes a +5-volt power cable and a DB9 to DB25 connector adapter

® IBM is a registered trademark145 of International Business Machines Corporation.

```
8491 '*         DECIMAL TO HEX CONVERSION
8492 '*                 INPUT:  K - INTEGER TO BE CONVERTED
8493 '*                 OUTPUT: HX$ - TWO CHARACTER STRING WITH HEX CONVERSION
8494 '*************************************************************************
8500 IF K > 255 THEN HX$="Too big":GOTO 8530
8510 HX$=MID$(H$,K\16+1,1)            'UPPER NIBBLE
8520 HX$=HX$+MID$(H$,(K MOD 16)+1,1)  'LOWER NIBBLE
8530 RETURN
9499 '******************** BOOT CODE ***************************************
9500 DATA 86, 23                 'LDAA   #$23
9510 DATA B7, 10, 02             'STAA   OPT2    make port C wire or
9520 DATA 86, FE                 'LDAA   #$FE
9530 DATA B7, 10, 03             'STAA   PORTC   light 1 LED on port C bit 0
9540 DATA C6, FF                 'LDAB   #$FF
9550 DATA F7, 10, 07             'STAB   DDRC    make port C outputs
9560 DATA CE, 0F, A0             'LDX    #4000   2msec at 2MHz
9570 DATA 18, CE, E0, 00         'LDY    #$E000  Start of BUFFALO 3.4
9580 DATA 7E, BF, 00             'JMP    $BF00   EPROM routine start address
9590 '*************************************************************************
```

Common Bootstrap Mode Problems

It is not unusual for a user to encounter problems with bootstrap mode because it is new to many users. By knowing some of the common difficulties, the user can avoid them or at least recognize and quickly correct them.

Reset Conditions vs. Conditions as Bootloaded Program Starts

It is common to confuse the reset state of systems and control bits with the state of these systems and control bits when a bootloaded program in RAM starts.

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.

Step 5

The CONFIG register defaults to hexadecimal 103F on the MC68HC711E9. PCBUG11 needs adressing parameters to allow programming of a specific block of memory so the following parameter must be given.

At the PCbug11 command prompt, type: EEPROM 0.

Then type: EEPROM 103F 103F.

Step 6

Erase the CONFIG to allow byte programming.

At the PCbug11 command prompt, type: EEPROM ERASE BULK 103F.

Step 7

You are now ready to download the program into the EEPROM and EPROM.

At the PCbug11command prompt, type: LOADSC:\MYPROG\MYPROG.S19.

For more details on programming the EPROM, read the engineering bulletin *Programming MC68HC711E9 Devices with PCbug11 and the M68HC11EVB*, Freescale document number EB187.

Step 8

You are now ready to enable the security feature on the MCHC711E9.

At the PCbug11 command prompt type: MS 103F 05.

Step 9

After the programming operation is complete, verifyng the CONFIG on the MCHC711E9 is not possible because in bootstrap mode the default value is always forced.

Step 10

The part is now in secure mode and whatever code you loaded into EEPROM will be erased if you tried to bring the microcontroller up in either expanded mode or bootstrap mode.

NOTE

It is important to note that the microcontroller will work properly in secure mode only in single chip mode.

NOTE

If the part is placed in bootstrap or expanded, the code in EEPROM and RAM will be erased and the microcontroller cannot be reused. The security software will constantly read the NOSEC bit and lock the part.