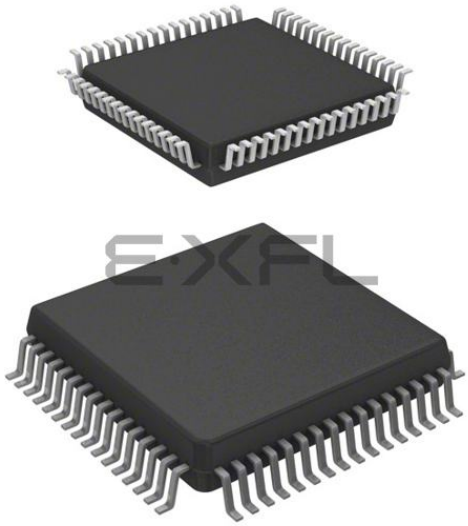


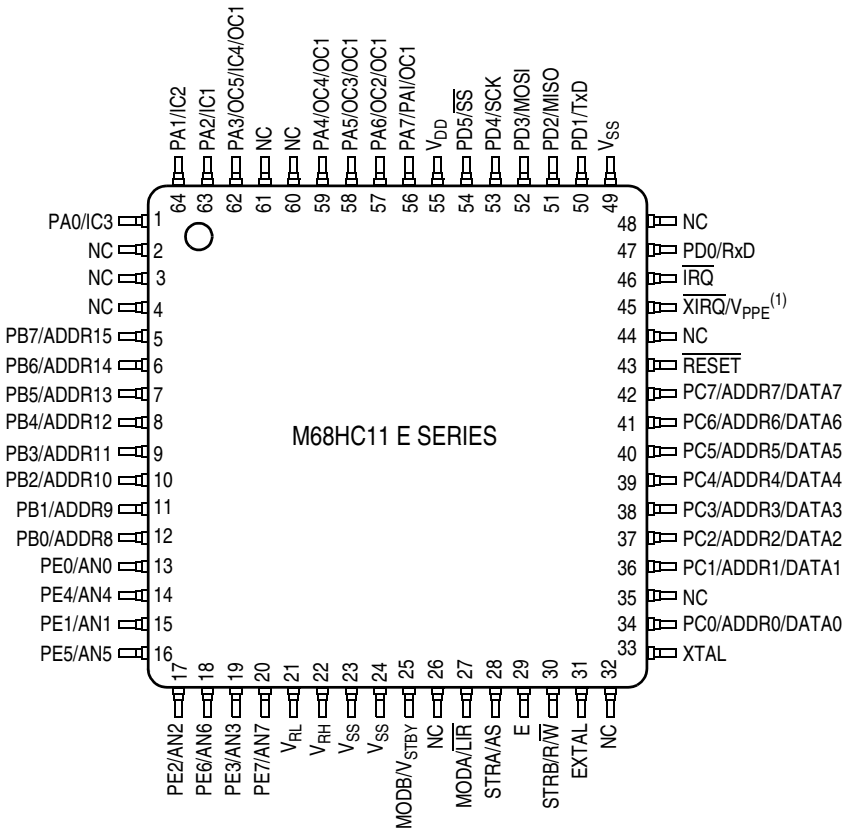
Welcome to E-XFL.COM

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	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	64-QFP
Supplier Device Package	64-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68711e20cfue2



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

Figure 1-3. Pin Assignments for 64-Pin QFP

NOTE

$\overline{IRQ}$ must be configured for level-sensitive operation if there is more than one source of $\overline{IRQ}$ interrupt.

There should be a single pullup resistor near the MCU interrupt input pin (typically 4.7 kΩ). There must also be an interlock mechanism at each interrupt source so that the source holds the interrupt line low until the MCU recognizes and acknowledges the interrupt request. If one or more interrupt sources are still pending after the MCU services a request, the interrupt line will still be held low and the MCU will be interrupted again as soon as the interrupt mask bit in the MCU is cleared (normally upon return from an interrupt). Refer to Chapter 5 Resets and Interrupts.

V_{PPE} is the input for the 12-volt nominal programming voltage required for EPROM/OTPROM programming. On devices without EPROM/OTPROM, this pin is only an $\overline{XIRQ}$ input.

CAUTION

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.

1.4.7 MODA and MODB (MODA/$\overline{LIR}$ and MODB/V_{STBY})

During reset, MODA and MODB select one of the four operating modes:

- Single-chip mode
- Expanded mode
- Test mode
- Bootstrap mode

Refer to Chapter 2 Operating Modes and On-Chip Memory.

After the operating mode has been selected, the load instruction register ($\overline{LIR}$) pin provides an open-drain output to indicate that execution of an instruction has begun. A series of E-clock cycles occurs during execution of each instruction. The $\overline{LIR}$ signal goes low during the first E-clock cycle of each instruction (opcode fetch). This output is provided for assistance in program debugging.

The V_{STBY} pin is used to input random-access memory (RAM) standby power. When the voltage on this pin is more than one MOS threshold (about 0.7 volts) above the V_{DD} voltage, the internal RAM and part of the reset logic are powered from this signal rather than the V_{DD} input. This allows RAM contents to be retained without V_{DD} power applied to the MCU. Reset must be driven low before V_{DD} is removed and must remain low until V_{DD} has been restored to a valid level.

1.4.8 V_{RL} and V_{RH}

These two inputs provide the reference voltages for the analog-to-digital (A/D) converter circuitry:

- V_{RL} is the low reference, typically 0 Vdc.
- V_{RH} is the high reference.

For proper A/D converter operation:

- V_{RH} should be at least 3 Vdc greater than V_{RL}.
- V_{RL} and V_{RH} should be between V_{SS} and V_{DD}.

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)

Address: $103F

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	EE3	EE2	EE1	EE0	NOSEC	NOCOP		EEON
Resets:								
Single chip:	1	1	1	1	U	U	1	1
Bootstrap:	1	1	1	1	U	U(L)	1	1
Expanded:	U	U	U	U	1	U	1	U
Test:	U	U	U	U	1	U(L)	1	0

= Unimplemented

U indicates a previously programmed bit. U(L) indicates that the bit resets to the logic level held in the latch prior to reset, but the function of COP is controlled by the DISR bit in TEST1 register.

Figure 2-11. MC68HC811E2 System Configuration Register (CONFIG)

EE[3:0] — EEPROM Mapping Bits

EE[3:0] apply only to MC68HC811E2 and allow the 2048 bytes of EEPROM to be remapped to any 4-Kbyte boundary. See Table 2-3.

Table 2-3. EEPROM Mapping

EE[3:0]	EEPROM Location
0 0 0 0	$0800–$0FFF
0 0 0 1	$1800–$1FFF
0 0 1 0	$2800–$2FFF
0 0 1 1	$3800–$3FFF
0 1 0 0	$4800–$4FFF
0 1 0 1	$5800–$5FFF
0 1 1 0	$6800–$6FFF
0 1 1 1	$7800–$7FFF
1 0 0 0	$8800–$8FFF
1 0 0 1	$9800–$9FFF
1 0 1 0	$A800–$AFFF
1 0 1 1	$B800–$BFFF
1 1 0 0	$C800–$CFFF
1 1 0 1	$D800–$DFFF
1 1 1 0	$E800–$EFFF
1 1 1 1	$F800–$FFFF

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.

5.2.5 System Configuration Options Register

Address: $1039

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

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

= Unimplemented

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

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

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

CSEL — Clock Select Bit

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

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

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

DLY — Enable Oscillator Startup Delay Bit

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

CME — Clock Monitor Enable Bit

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

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

Bit 2 — Unimplemented

Always reads 0

CR[1:0] — COP Timer Rate Select Bit

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

Table 5-4. Interrupt and Reset Vector Assignments

Vector Address	Interrupt Source	CCR Mask Bit	Local Mask
FFC0, C1 – FFD4, D5	Reserved	—	—
FFD6, D7	SCI serial system • SCI receive data register full • SCI receiver overrun • SCI transmit data register empty • SCI transmit complete • SCI idle line detect	I	RIE RIE TIE TCIE ILIE
FFD8, D9	SPI serial transfer complete	I	SPIE
FFDA, DB	Pulse accumulator input edge	I	PAII
FFDC, DD	Pulse accumulator overflow	I	PAOVI
FFDE, DF	Timer overflow	I	TOI
FFE0, E1	Timer input capture 4/output compare 5	I	I4/O5I
FFE2, E3	Timer output compare 4	I	OC4I
FFE4, E5	Timer output compare 3	I	OC3I
FFE6, E7	Timer output compare 2	I	OC2I
FFE8, E9	Timer output compare 1	I	OC1I
FFEA, EB	Timer input capture 3	I	IC3I
FFEC, ED	Timer input capture 2	I	IC2I
FFEE, EF	Timer input capture 1	I	IC1I
FFF0, F1	Real-time interrupt	I	RTII
FFF2, F3	$\overline{\text{IRQ}}$ (external pin)	I	None
FFF4, F5	$\overline{\text{XIRQ}}$ pin	X	None
FFF6, F7	Software interrupt	None	None
FFF8, F9	Illegal opcode trap	None	None
FFFA, FB	COP failure	None	NOCOP
FFFC, FD	Clock monitor fail	None	CME
FFFE, FF	$\overline{\text{RESET}}$	None	None

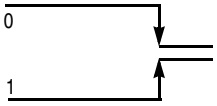
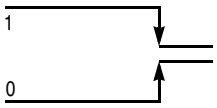
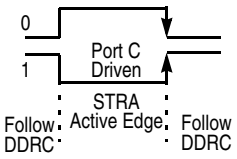
5.5.1 Interrupt Recognition and Register Stacking

An interrupt can be recognized at any time after it is enabled by its local mask, if any, and by the global mask bit in the CCR. Once an interrupt source is recognized, the CPU responds at the completion of the instruction being executed. Interrupt latency varies according to the number of cycles required to complete the current instruction. When the CPU begins to service an interrupt, the contents of the CPU registers are pushed onto the stack in the order shown in Table 5-5. After the CCR value is stacked, the I bit and the X bit, if $\overline{\text{XIRQ}}$ is pending, are set to inhibit further interrupts. The interrupt vector for the highest priority pending source is fetched and execution continues at the address specified by the vector. At the

6.8 Parallel I/O Control Register

The parallel handshake functions are available only in the single-chip operating mode. PIOC is a read/write register except for bit 7, which is read only. Table 6-2 shows a summary of handshake operations.

Table 6-2. Parallel I/O Control

	STAF Clearing Sequence	HNDS	OIN	PLS	EGA	Port B	Port C
Simple strobed mode	Read PIOC with STAF = 1 then read PORTCL	0	X	X		Inputs latched into PORTCL on any active edge on STRA	STRB pulses on writes to PORTB
Full-input hand-shake mode	Read PIOC with STAF = 1 then read PORTCL	1	0	0 = STRB active level 1 = STRB active pulse		Inputs latched into PORTCL on any active edge on STRA	Normal output port, unaffected in handshake modes
Full-output hand-shake mode	Read PIOC with STAF = 1 then write PORTCL	1	1	0 = STRB active level 1 = STRB active pulse		Driven as outputs if STRA at active level; follows DDRC if STRA not at active level	Normal output port, unaffected in handshake modes

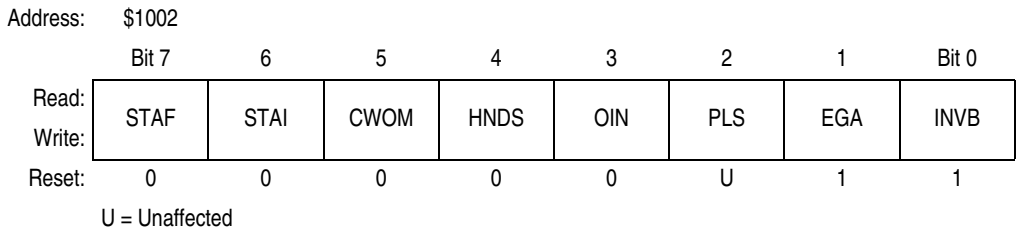


Figure 6-10. Parallel I/O Control Register (PIOC)

STAF — Strobe A Interrupt Status Flag

STAF is set when the selected edge occurs on strobe A. This bit can be cleared by a read of PIOC with STAF set followed by a read of PORTCL (simple strobed or full input handshake mode) or a write to PORTCL (output handshake mode).

0 = No edge on strobe A
1 = Selected edge on strobe A

STAI — Strobe A Interrupt Enable Mask Bit

0 = STAF does not request interrupt
1 = STAF requests interrupt

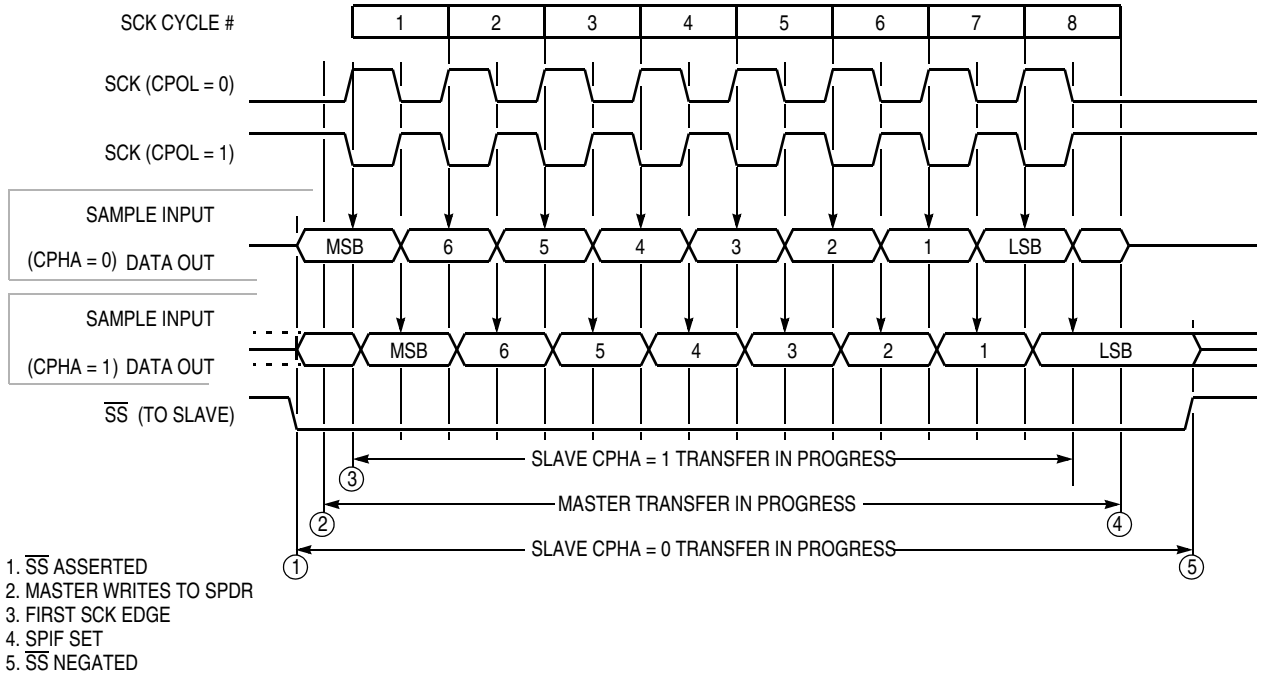


Figure 8-2. SPI Transfer Format

8.5 SPI Signals

This subsection contains descriptions of the four SPI signals:

- Master in/slave out (MISO)
- Master out/slave in (MOSI)
- Serial clock (SCK)
- Slave select ($\overline{SS}$)

Any SPI output line must have its corresponding data direction bit in DDRD register set. If the DDR bit is clear, that line is disconnected from the SPI logic and becomes a general-purpose input. All SPI input lines are forced to act as inputs regardless of the state of the corresponding DDR bits in DDRD register.

8.5.1 Master In/Slave Out

MISO is one of two unidirectional serial data signals. It is an input to a master device and an output from a slave device. The MISO line of a slave device is placed in the high-impedance state if the slave device is not selected.

8.5.2 Master Out/Slave In

The MOSI line is the second of the two unidirectional serial data signals. It is an output from a master device and an input to a slave device. The master device places data on the MOSI line a half-cycle before the clock edge that the slave device uses to latch the data.

MSTR — Master Mode Select Bit

It is customary to have an external pullup resistor on lines that are driven by open-drain devices.

0 = Slave mode
1 = Master mode

CPOL — Clock Polarity Bit

When the clock polarity bit is cleared and data is not being transferred, the SCK pin of the master device has a steady state low value. When CPOL is set, SCK idles high. Refer to Figure 8-2 and 8.4 Clock Phase and Polarity Controls.

CPHA — Clock Phase Bit

The clock phase bit, in conjunction with the CPOL bit, controls the clock-data relationship between master and slave. The CPHA bit selects one of two different clocking protocols. Refer to Figure 8-2 and 8.4 Clock Phase and Polarity Controls.

SPR[1:0] — SPI Clock Rate Select Bits

These two bits select the SPI clock (SCK) rate when the device is configured as master. When the device is configured as slave, these bits have no effect. Refer to Table 8-1.

Table 8-1. SPI Clock Rates

SPR[1:0]	Divide E Clock By	Frequency at E = 1 MHz (Baud)	Frequency at E = 2 MHz (Baud)	Frequency at E = 3 MHz (Baud)	Frequency at E = 4 MHz (Baud)
0 0	2	500 kHz	1.0 MHz	1.5 MHz	2 MHz
0 1	4	250 kHz	500 kHz	750 kHz	1 MHz
1 0	16	62.5 kHz	125 kHz	187.5 kHz	250 kHz
1 1	32	31.3 kHz	62.5 kHz	93.8 kHz	125 kHz

8.7.2 Serial Peripheral Status Register

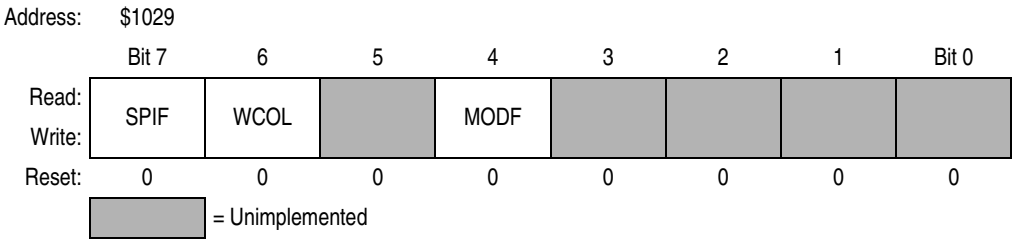


Figure 8-4. Serial Peripheral Status Register (SPSR)

SPIF — SPI Interrupt Complete Flag

SPIF is set upon completion of data transfer between the processor and the external device. If SPIF goes high, and if SPIE is set, a serial peripheral interrupt is generated. To clear the SPIF bit, read the SPSR with SPIF set, then access the SPDR. Unless SPSR is read (with SPIF set) first, attempts to write SPDR are inhibited.

WCOL — Write Collision Bit

Clearing the WCOL bit is accomplished by reading the SPSR (with WCOL set) followed by an access of SPDR. Refer to 8.5.4 Slave Select and 8.6 SPI System Errors.

0 = No write collision
1 = Write collision

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)

9.7.2 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.

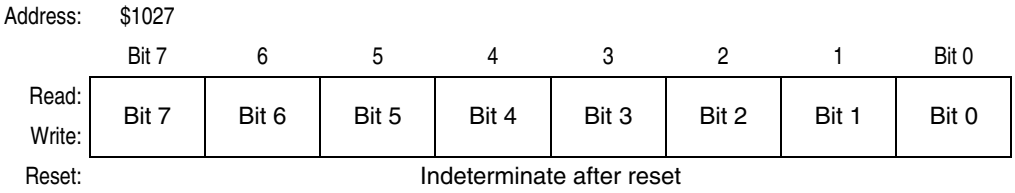


Figure 9-26. Pulse Accumulator Count Register (PACNT)

9.7.3 Pulse Accumulator Status and Interrupt Bits

The pulse accumulator control bits, PAOVI and PAII, PAOVF and PAIF, are located within timer registers TMSK2 and TFLG2.

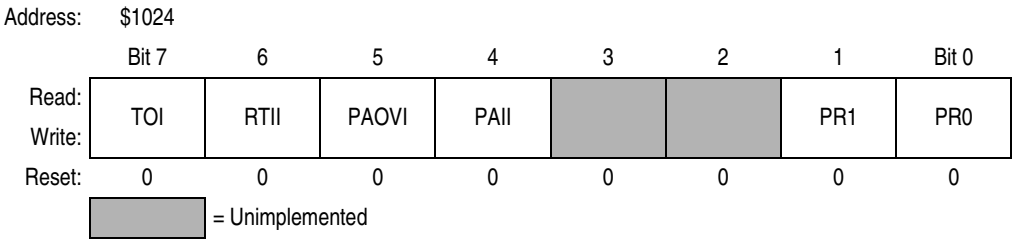


Figure 9-27. Timer Interrupt Mask 2 Register (TMSK2)

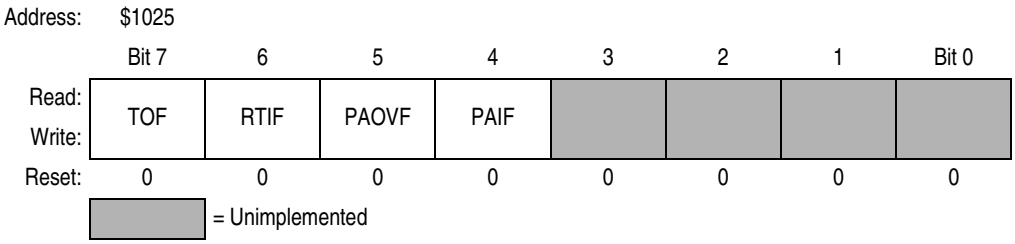


Figure 9-28. Timer Interrupt Flag 2 Register (TFLG2)

PAOVI and PAOVF — Pulse Accumulator Interrupt Enable and Overflow Flag

The PAOVF status bit is set each time the pulse accumulator count rolls over from $FF to $00. To clear this status bit, write a 1 in the corresponding data bit position (bit 5) of the TFLG2 register. The PAOVI control bit allows configuring the pulse accumulator overflow for polled or interrupt-driven operation and does not affect the state of PAOVF. When PAOVI is 0, pulse accumulator overflow interrupts are inhibited, and the system operates in a polled mode, which requires that PAOVF be polled by user software to determine when an overflow has occurred. When the PAOVI control bit is set, a hardware interrupt request is generated each time PAOVF is set. Before leaving the interrupt service routine, software must clear PAOVF by writing to the TFLG2 register.

Chapter 10
Electrical Characteristics

10.1 Introduction

This section contains electrical specifications for the M68HC11 E-series devices.

10.2 Maximum Ratings for Standard and Extended Voltage Devices

Maximum ratings are the extreme limits to which the microcontroller unit (MCU) can be exposed without permanently damaging it.

NOTE

This device is not guaranteed to operate properly at the maximum ratings. Refer to 10.5 DC Electrical Characteristics, 10.6 Supply Currents and Power Dissipation, 10.7 MC68L11E9/E20 DC Electrical Characteristics, and 10.8 MC68L11E9/E20 Supply Currents and Power Dissipation for guaranteed operating conditions.

Rating	Symbol	Value	Unit
Supply voltage	V_{DD}	–0.3 to +7.0	V
Input voltage	V_{In}	–0.3 to +7.0	V
Current drain per pin(1) excluding V_{DD}, V_{SS}, AV_{DD}, V_{RH}, V_{RL}, and $\overline{XIRQ}/V_{PPE}$	I_D	25	mA
Storage temperature	T_{STG}	–55 to +150	°C

1. One pin at a time, observing maximum power dissipation limits

NOTE

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{In} and V_{Out} be constrained to the range $V_{SS} \le (V_{In}$ or $V_{Out}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (for example, either V_{SS} or V_{DD}).

10.11 Peripheral Port Timing

Characteristic[1] [2]	Symbol	1.0 MHz		2.0 MHz		3.0 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of operation E-clock frequency	f_o	dc	1.0	dc	2.0	dc	3.0	MHz
E-clock period	t_{CYC}	1000	—	500	—	333	—	ns
Peripheral data setup time MCU read of ports A, C, D, and E	t_{PDSU}	100	—	100	—	100	—	ns
Peripheral data hold time MCU read of ports A, C, D, and E	t_{PDH}	50	—	50	—	50	—	ns
Delay time, peripheral data write t_{PWD} = 1/4 t_{CYC}+ 100 ns MCU writes to port A MCU writes to ports B, C, and D	t_{PWD}	 — —	 200 350	 — —	 200 225	 — —	 200 183	ns
Port C input data setup time	t_{IS}	60	—	60	—	60	—	ns
Port C input data hold time	t_{IH}	100	—	100	—	100	—	ns
Delay time, E fall to STRB t_{DEB} = 1/4 t_{CYC}+ 100 ns	t_{DEB}	—	350	—	225	—	183	ns
Setup time, STRA asserted to E fall[3]	t_{AES}	0	—	0	—	0	—	ns
Delay time, STRA asserted to port C data output valid	t_{PCD}	—	100	—	100	—	100	ns
Hold time, STRA negated to port C data	t_{PCH}	10	—	10	—	10	—	ns
3-state hold time	t_{PCZ}	—	150	—	150	—	150	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. Ports C and D timing is valid for active drive. (CWOM and DWOM bits are not set in PIOC and SPCR registers, respectively.)
3. If this setup time is met, STRB acknowledges in the next cycle. If it is not met, the response may be delayed one more cycle.

10.15 Expansion Bus Timing Characteristics

Num	Characteristic[1]	Symbol	1.0 MHz		2.0 MHz		3.0 MHz		Unit
			Min	Max	Min	Max	Min	Max	
	Frequency of operation (E-clock frequency)	f_o	dc	1.0	dc	2.0	dc	3.0	MHz
1	Cycle time	t_{CYC}	1000	—	500	—	333	—	ns
2	Pulse width, E low[2], PW_{EL} = 1/2 t_{CYC}–23 ns	PW_{EL}	477	—	227	—	146	—	ns
3	Pulse width, E high[2], PW_{EH} = 1/2 t_{CYC}–28 ns	PW_{EH}	472	—	222	—	141	—	ns
4a	E and AS rise time	t_r	—	20	—	20	—	20	ns
4b	E and AS fall time	t_f	—	20	—	20	—	15	ns
9	Address hold time[2] [3]a, t_{AH} = 1/8 t_{CYC}–29.5 ns	t_{AH}	95.5	—	33	—	26	—	ns
12	Non-multiplexed address valid time to E rise $t_{AV} = PW_{EL} – (t_{ASD} + 80\ ns)$[2] [3]a	t_{AV}	281.5	—	94	—	54	—	ns
17	Read data setup time	t_{DSR}	30	—	30	—	30	—	ns
18	Read data hold time, max = t_{MAD}	t_{DHR}	0	145.5	0	83	0	51	ns
19	Write data delay time, t_{DDW} = 1/8 t_{CYC}+ 65.5 ns[2] [3]a	t_{DDW}	—	190.5	—	128		71	ns
21	Write data hold time, t_{DHW} = 1/8 t_{CYC}–29.5 ns[2] [3]a	t_{DHW}	95.5	—	33	—	26	—	ns
22	Multiplexed address valid time to E rise $t_{AVM} = PW_{EL} – (t_{ASD} + 90\ ns)$[2] [3]a	t_{AVM}	271.5	—	84	—	54	—	ns
24	Multiplexed address valid time to AS fall $t_{ASL} = PW_{ASH} – 70\ ns$[2]	t_{ASL}	151	—	26	—	13	—	ns
25	Multiplexed address hold time t_{AHL} = 1/8 t_{CYC}–29.5 ns[2] [3]b	t_{AHL}	95.5	—	33	—	31	—	ns
26	Delay time, E to AS rise, t_{ASD} = 1/8 t_{CYC}–9.5 ns[2] [3]a	t_{ASD}	115.5	—	53	—	31	—	ns
27	Pulse width, AS high, PW_{ASH} = 1/4 t_{CYC}–29 ns[2]	PW_{ASH}	221	—	96	—	63	—	ns
28	Delay time, AS to E rise, t_{ASED} = 1/8 t_{CYC}–9.5 ns[2] [3]b	t_{ASED}	115.5	—	53	—	31	—	ns
29	MPU address access time[3]a $t_{ACCA} = t_{CYC} – (PW_{EL} – t_{AVM}) – t_{DSR} – t_f$	t_{ACCA}	744.5	—	307	—	196	—	ns
35	MPU access time, $t_{ACCE} = PW_{EH} – t_{DSR}$	t_{ACCE}	—	442	—	192		111	ns
36	Multiplexed address delay (Previous cycle MPU read) $t_{MAD} = t_{ASD} + 30\ ns$[2] [3]a	t_{MAD}	145.5	—	83	—	51	—	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. Formula only for dc to 2 MHz
3. Input clocks with duty cycles other than 50% affect bus performance. Timing parameters affected by input clock duty cycle are identified by (a) and (b). To recalculate the approximate bus timing values, substitute the following expressions in place of 1/8 t_{CYC}in the above formulas, where applicable:
   (a) (1–dc) × 1/4 t_{CYC}
   (b) dc × 1/4 t_{CYC}

Where:
dc is the decimal value of duty cycle percentage (high time)

Listing 1. MCU-to-MCU Duplicator Program

```
 82 B666
 83 B666 150482   DUNPRG     BCLR   PORTB (RESET+RED) Red OFF, apply reset
 84 B669 20FE                BRA    *                 Done so just hang
 85 B66B
 86               ************************************************
 87               * Subroutine to get & send an SCI char.  Also
 88               *  advances pointer (X).
 89               ************************************************
 90 B66B A600     SEND1      LDAA   0,X               Get a character
 91 B66D 132E80FC TRDYLP     BRCLR  SCSR TDRE TRDYLP Wait for TDRE
 92 B671 972F                STAA   SCDR              Send character
 93 B673 08                  INX                     Advance pointer
 94 B674 39                  RTS                     ** Return **
 95
 96               ************************************************
 97               * Program to be bootloaded to target '711E9
 98               ************************************************
 99 B675 8604     BLPROG     LDAA   #$04              Pattern for DWOM off, no SPI
100 B677 B71028              STAA   $1028             Turns off DWOM in target MCU
101               * NOTE: Can't use direct addressing in target MCU because
102               *       regs are located at $1000.
103 B67A 7EBF00              JMP    PROGRAM           Jumps to EPROM prog routine
104 B67D          ENDBPR     EQU    *

Symbol Table:
Symbol Name     Value  Def.#  Line Number Cross Reference

BEGIN            B600 *00029
BLLOOP           B616 *00038  00040
BLPROG           B675 *00099  00037
DATALP           B648 *00068  00079
DLYLP            B620 *00046  00047
DLYLP2           B637 *00059  00063
DUNPRG           B666 *00083  00076
ENDBPR           B67D *00104  00039
EPSTRT           D000 *00023  00055 00066
GREEN            0001 *00015  00075 00081
INIT             103D *00009  00029
PORTB            0004 *00011  00033 00058 00061 00075 00081 00083
PORTE            000A *00016  00059
PROGRAM          BF00 *00022  00103
RDRF             0020 *00020  00034 00053 00071
RED              0002 *00014  00058 00061 00075 00083
RESET            0080 *00013  00032 00083
SCDR             002F *00021  00036 00049 00054 00072 00092
SCSR             002E *00017  00034 00048 00053 00071 00091
SEND1            B66B *00090  00038 00067 00070
SPCR             0028 *00010  00031
TDRE             0080 *00019  00091
TRDYLP           B66D *00091  00091
VERF             B64F *00071  00069 00071
VERFOK           B65F *00078  00074
WT4BRK           B60B *00034  00034
WT4FF            B627 *00053  00053
WT4VPP           B630 *00057  00060
```

```
1640 GOSUB 8000                 'GET BYTE FOR VERIFICATION
1650 RCV = I - 1
1660 LOCATE 10,1:PRINT "Verifying byte #"; I; "    "
1664 IF CHR$(CODE%(RCV)) = B$ THEN 1670
1665 K=CODE%(RCV):GOSUB 8500
1666 LOCATE 1,1:PRINT "Byte #"; I; "    ", " - Sent "; HX$;
1668 K=ASC(B$):GOSUB 8500
1669 PRINT "  Received "; HX$;
1670 NEXT I
1680 GOSUB 8000                 'GET BYTE FOR VERIFICATION
1690 RCV = CODESIZE% - 1
1700 LOCATE 10,1:PRINT "Verifying byte #"; CODESIZE%; "    "
1710 IF CHR$(CODE%(RCV)) = B$ THEN 1720
1713 K=CODE(RCV):GOSUB 8500
1714 LOCATE 1,1:PRINT "Byte #"; CODESIZE%; "    ", " - Sent "; HX$;
1715 K=ASC(B$):GOSUB 8500
1716 PRINT "  Received "; HX$;
1720 LOCATE 8, 1: PRINT : PRINT "Done!!!!"
4900 CLOSE
4910 INPUT "Press [RETURN] to quit...", Q$
5000 END
5900 '************************************************************************
5910 '*         SUBROUTINE TO READ IN ONE BYTE FROM A DISK FILE
5930 '*                 RETURNS BYTE IN A$
5940 '************************************************************************
6000 FLAG = 0
6010 IF EOF(1) THEN FLAG = 1: RETURN
6020 A$ = INPUT$(1, #1)
6030 RETURN
6490 '************************************************************************
6492 '*         SUBROUTINE TO SEND THE STRING IN A$ OUT TO THE DEVICE
6494 '*                 OPENED AS FILE #2.
6496 '************************************************************************
6500 PRINT #2, A$;
6510 RETURN
6590 '************************************************************************
6594 '*         SUBROUTINE THAT CONVERTS THE HEX DIGIT IN A$ TO AN INTEGER
6596 '************************************************************************
7000 X = INSTR(H$, A$)
7010 IF X = 0 THEN FLAG = 1
7020 X = X - 1
7030 RETURN
7990 '************************************************************************
7992 '*         SUBROUTINE TO READ IN ONE BYTE THROUGH THE COMM PORT OPENED
7994 '*                 AS FILE #2.  WAITS INDEFINITELY FOR THE BYTE TO BE
7996 '*                 RECEIVED.  SUBROUTINE WILL BE ABORTED BY ANY
7998 '*                 KEYBOARD INPUT.  RETURNS BYTE IN B$.  USES Q$.
7999 '************************************************************************
8000 WHILE LOC(2) = 0           'WAIT FOR COMM PORT INPUT
8005 Q$ = INKEY$: IF Q$ <> "" THEN 4900 'IF ANY KEY PRESSED, THEN ABORT
8010 WEND
8020 B$ = INPUT$(1, #2)
8030 RETURN
8490 '************************************************************************
```

Freescale Semiconductor
Engineering Bulletin

EB188
Rev. 0.1, 07/2005

Enabling the Security Feature on M68HC811E2 Devices with PCbug11 on the M68HC711E9PGMR

By Edgar Saenz
Austin, Texas

Introduction

The PCbug11 software, needed along with the M68HC711E9PGMR to program MC68HC811E2 devices, is available from the download section of the Microcontroller Worldwide Web site

http://www.freescale.com

Retrieve the file pcbug342.exe (a self-extracting archive) from the MCU11 directory.

Some Freescale evaluation board products also are shipped with PCbug11.

NOTE

For specific information about any of the PCbug11 commands, see the appropriate sections in the PCbug11 User's Manual (part number M68PCBUG11/D2), which is available from the Freescale Literature http://www.freescale.com. The file is also on the software download system and is called pcbug11.pdf.

© Freescale Semiconductor, Inc., 2005. All rights reserved.

Step 6

After the programming operation is complete, PCbug11 will display this message

Total bytes loaded: $xxxx

Total bytes programmed: $yyyy

- You should now remove the programming voltage from P4 connector pin 18, the XIRQ* pin.
- Each ORG directive in your assembly language source will cause a pair of these lines to be generated. For this operation, $yyyy will be incremented by the size of each block of code programmed into the EPROM of the MC68HC711E9.
- PCbug11 will display the above message whether or not the programming operation was successful. As a precaution, you should have PCbug11 verify your code.
- At the PCbug11 command prompt type: VERF C:\MYPROG\ISHERE.S19

Substitute the name of your program into the command above. Use a full path name if your program is not located in the same directory as PCbug11.

If the verify operation fails, a list of addresses which did not program correctly is displayed. Should this occur, you probably need to erase your part more completely. To do so, allow the MC68HC711E9 to sit for at least 45 minutes under an ultraviolet light source. Attempt the programming operation again. If you have purchased devices in plastic packages (one-time programmable parts), you will need to try again with a new, unprogrammed device.