

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	3MHz
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/mc68711e20cfue3

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.

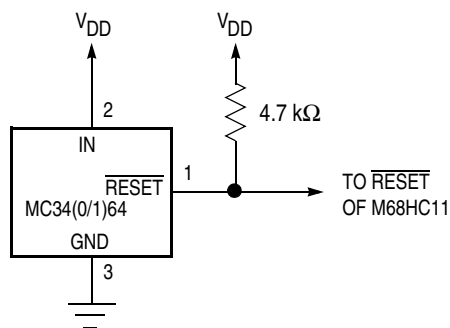


Figure 1-7. External Reset Circuit

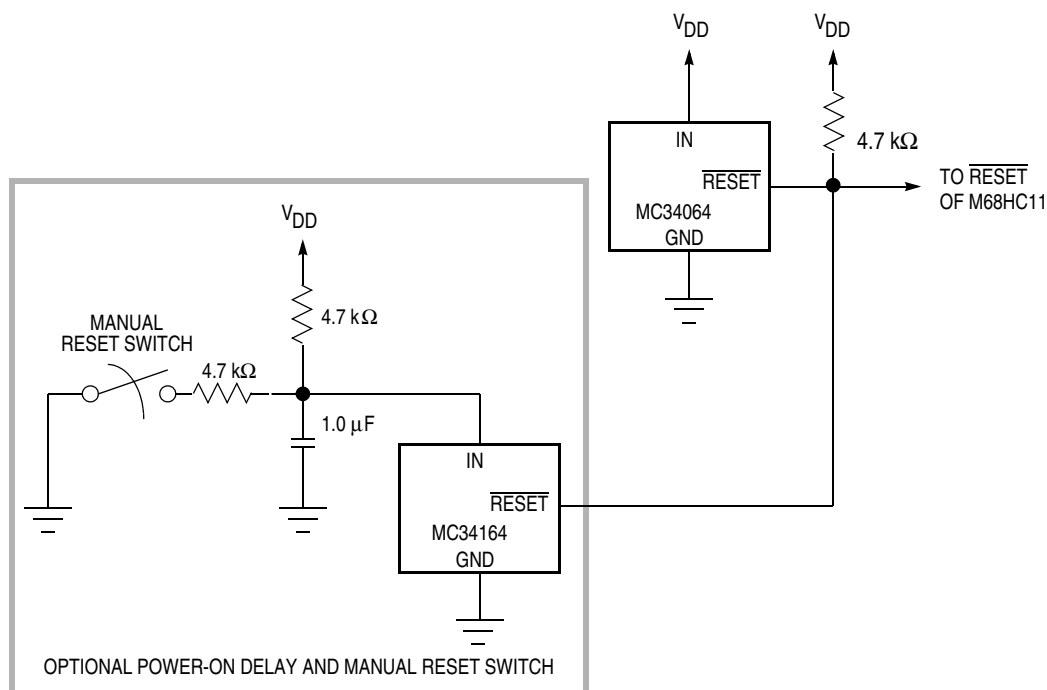


Figure 1-8. External Reset Circuit with Delay

1.4.9 STRA/AS

The strobe A (STRA) and address strobe (AS) pin performs either of two separate functions, depending on the operating mode:

- In single-chip mode, STRA performs an input handshake (strobe input) function.
- In the expanded multiplexed mode, AS provides an address strobe function.

AS can be used to demultiplex the address and data signals at port C. Refer to Chapter 2 Operating Modes and On-Chip Memory.

1.4.10 STRB/R \overline{W}

The strobe B (STRB) and read/write (R \overline{W}) pin act as either an output strobe or as a data bus direction indicator, depending on the operating mode.

In single-chip operating mode, STRB acts as a programmable strobe for handshake with other parallel devices. Refer to Chapter 6 Parallel Input/Output (I/O) Ports for further information.

In expanded multiplexed operating mode, R \overline{W} is used to indicate the direction of transfers on the external data bus. A low on the R \overline{W} pin indicates data is being written to the external data bus. A high on this pin indicates that a read cycle is in progress. R \overline{W} stays low during consecutive data bus write cycles, such as a double-byte store. It is possible for data to be driven out of port C, if internal read visibility (IRV) is enabled and an internal address is read, even though R \overline{W} is in a high-impedance state. Refer to Chapter 2 Operating Modes and On-Chip Memory for more information about IRVNE (internal read visibility not E).

1.4.11 Port Signals

Port pins have different functions in different operating modes. Pin functions for port A, port D, and port E are independent of operating modes. Port B and port C, however, are affected by operating mode. Port B provides eight general-purpose output signals in single-chip operating modes. When the microcontroller is in expanded multiplexed operating mode, port B pins are the eight high-order address lines.

Port C provides eight general-purpose input/output signals when the MCU is in the single-chip operating mode. When the microcontroller is in the expanded multiplexed operating mode, port C pins are a multiplexed address/data bus.

Refer to Table 1-1 for a functional description of the 40 port signals within different operating modes. Terminate unused inputs and input/output (I/O) pins configured as inputs high or low.

1.4.12 Port A

In all operating modes, port A can be configured for three timer input capture (IC) functions and four timer output compare (OC) functions. An additional pin can be configured as either the fourth IC or the fifth OC. Any port A pin that is not currently being used for a timer function can be used as either a general-purpose input or output line. Only port A pins PA7 and PA3 have an associated data direction control bit that allows the pin to be selectively configured as input or output. Bits DDRA7 and DDRA3 located in PACTL register control data direction for PA7 and PA3, respectively. All other port A pins are fixed as either input or output.

PA7 can function as general-purpose I/O or as timer output compare for OC1. PA7 is also the input to the pulse accumulator, even while functioning as a general-purpose I/O or an OC1 output.

General Description

1.4.15 Port D

Pins PD5–PD0 can be used for general-purpose I/O signals. These pins alternately serve as the serial communication interface (SCI) and serial peripheral interface (SPI) signals when those subsystems are enabled.

- PD0 is the receive data input (RxD) signal for the SCI.
- PD1 is the transmit data output (TxD) signal for the SCI.
- PD5–PD2 are dedicated to the SPI:
 - PD2 is the master in/slave out (MISO) signal.
 - PD3 is the master out/slave in (MOSI) signal.
 - PD4 is the serial clock (SCK) signal.
 - PD5 is the slave select (\overline{SS}) input.

1.4.16 Port E

Use port E for general-purpose or analog-to-digital (A/D) inputs.

CAUTION

If high accuracy is required for A/D conversions, avoid reading port E during sampling, as small disturbances can reduce the accuracy of that result.

Operating Modes and On-Chip Memory

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

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

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

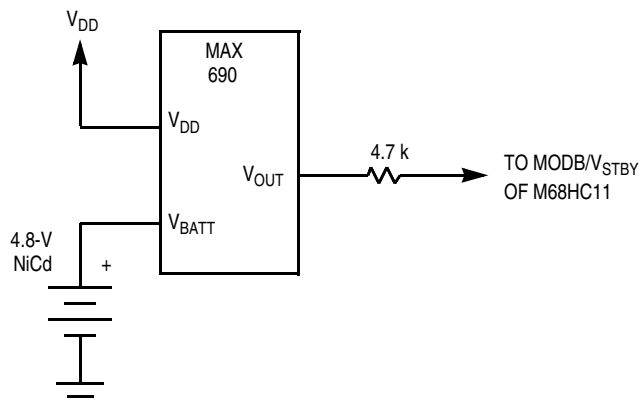


Figure 2-8. RAM Standby MODB/ V_{STBY} Connections

The bootloader program is contained in the internal bootstrap ROM. This ROM, which appears as internal memory space at locations \$BF00–\$BFFF, is enabled only if the MCU is reset in special bootstrap mode.

In expanded modes, the ROM/EPROM/OTPROM (if present) is enabled out of reset and located at the top of the memory map if the ROMON bit in the CONFIG register is set. ROM or EPROM is enabled out of reset in single-chip and bootstrap modes, regardless of the state of ROMON.

For devices with 512 bytes of EEPROM, the EEPROM is located at \$B600–\$B7FF and has the same read cycle time as the internal ROM. The 512 bytes of EEPROM cannot be remapped to other locations.

For the MC68HC811E2, EEPROM is located at \$F800–\$FFFF and can be remapped to any 4-Kbyte boundary. EEPROM mapping control bits (EE[3:0] in CONFIG) determine the location of the 2048 bytes of EEPROM and are present only on the MC68HC811E2. Refer to 2.3.3.1 System Configuration Register for a description of the MC68HC811E2 CONFIG register.

EEPROM can be programmed or erased by software and an on-chip charge pump, allowing EEPROM changes using the single V_{DD} supply.

2.3.2 Mode Selection

The four mode variations are selected by the logic states of the MODA and MODB pins during reset. The MODA and MODB logic levels determine the logic state of SMOD and the MDA control bits in the highest priority I-bit interrupt and miscellaneous (HPRIO) register.

After reset is released, the mode select pins no longer influence the MCU operating mode. In single-chip operating mode, the MODA pin is connected to a logic level 0. In expanded mode, MODA is normally connected to V_{DD} through a pullup resistor of 4.7 kΩ. The MODA pin also functions as the load instruction register \overline{LIR} pin when the MCU is not in reset. The open-drain active low \overline{LIR} output pin drives low during the first E cycle of each instruction. The MODB pin also functions as standby power input (V_{STBY}), which allows RAM contents to be maintained in absence of V_{DD} .

Refer to Table 2-1, which is a summary of mode pin operation, the mode control bits, and the four operating modes.

Table 2-1. Hardware Mode Select Summary

Input Levels at Reset		Mode	Control Bits in HPRI0 (Latched at Reset)		
MODB	MODA		RBOOT	SMOD	MDA
1	0	Single chip	0	0	0
1	1	Expanded	0	0	1
0	0	Bootstrap	1	1	0
0	1	Special test	0	1	1

A normal mode is selected when MODB is logic 1 during reset. One of three reset vectors is fetched from address \$FFFA–\$FFFF, and program execution begins from the address indicated by this vector. If MODB is logic 0 during reset, the special mode reset vector is fetched from addresses \$BFFA–\$BFFF, and software has access to special test features. Refer to Chapter 5 Resets and Interrupts.

Address: \$103C								
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	RBOOT ⁽¹⁾	SMOD ⁽¹⁾	MDA ⁽¹⁾	IRV(NE) ⁽¹⁾	PSEL3	PSEL2	PSEL1	PSEL0
Write:								
Resets:								
Single chip:	0	0	0	0	0	1	1	0
Expanded:	0	0	1	0	0	1	1	0
Bootstrap:	1	1	0	0	0	1	1	0
Test:	0	1	1	1	0	1	1	0

1. The reset values depend on the mode selected at the RESET pin rising edge.

Figure 2-9. Highest Priority I-Bit Interrupt and Miscellaneous Register (HPRI0)

RBOOT — Read Bootstrap ROM Bit

Valid only when SMOD is set (bootstrap or special test mode); can be written only in special modes
0 = Bootloader ROM disabled and not in map
1 = Bootloader ROM enabled and in map at \$BE00–\$BFFF

SMOD and MDA — Special Mode Select and Mode Select A Bits

The initial value of SMOD is the inverse of the logic level present on the MODB pin at the rising edge of reset. The initial value of MDA equals the logic level present on the MODA pin at the rising edge of reset. These two bits can be read at any time. They can be written anytime in special modes. MDA can be written only once in normal modes. SMOD cannot be set once it has been cleared.

Input		Mode	Latched at Reset	
MODB	MODA		SMOD	MDA
1	0	Single chip	0	0
1	1	Expanded	0	1

2.3.3.1 System Configuration Register

The system configuration register (CONFIG) consists of an EEPROM byte and static latches that control the startup configuration of the MCU. The contents of the EEPROM byte are transferred into static working latches during reset sequences. The operation of the MCU is controlled directly by these latches and not by CONFIG itself. In normal modes, changes to CONFIG do not affect operation of the MCU until after the next reset sequence. When programming, the CONFIG register itself is accessed. When the CONFIG register is read, the static latches are accessed. See 2.5.1 EEPROM and CONFIG Programming and Erasure for information on modifying CONFIG.

To take full advantage of the MCU's functionality, customers can program the CONFIG register in bootstrap mode. This can be accomplished by setting the mode pins to logic 0 and downloading a small program to internal RAM. For more information, Freescale application note AN1060 entitled M68HC11 Bootstrap Mode has been included at the back of this document. The downloadable talker will consist of:

- Bulk erase
- Byte programming
- Communication server

All of this functionality is provided by PCbug11 which can be found on the Freescale Web site at <http://www.freescale.com>. For more information on using PCbug11 to program an E-series device, Freescale engineering bulletin EB296 entitled Programming MC68HC711E9 Devices with PCbug11 and the M68HC11EVBU has been included at the back of this document.

NOTE

The CONFIG register on the 68HC11 is an EEPROM cell and must be programmed accordingly.

Operation of the CONFIG register in the MC68HC811E2 differs from other devices in the M68HC11 E series. See Figure 2-10 and Figure 2-11.

Address: \$103F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:					NOSEC	NOCOP	ROMON	EEON
Write:								

Resets:

Single chip:	0	0	0	0	U	U	1	U
Bootstrap:	0	0	0	0	U	U(L)	U	U
Expanded:	0	0	0	0	1	U	U	U
Test:	0	0	0	0	1	U(L)	U	U

= 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 2-10. System Configuration Register (CONFIG)

4.4 Opcodes and Operands

The M68HC11 Family of microcontrollers uses 8-bit opcodes. Each opcode identifies a particular instruction and associated addressing mode to the CPU. Several opcodes are required to provide each instruction with a range of addressing capabilities. Only 256 opcodes would be available if the range of values were restricted to the number able to be expressed in 8-bit binary numbers.

A 4-page opcode map has been implemented to expand the number of instructions. An additional byte, called a prebyte, directs the processor from page 0 of the opcode map to one of the other three pages. As its name implies, the additional byte precedes the opcode.

A complete instruction consists of a prebyte, if any, an opcode, and zero, one, two, or three operands. The operands contain information the CPU needs for executing the instruction. Complete instructions can be from one to five bytes long.

4.5 Addressing Modes

Six addressing modes can be used to access memory:

- Immediate
- Direct
- Extended
- Indexed
- Inherent
- Relative

These modes are detailed in the following paragraphs. All modes except inherent mode use an effective address. The effective address is the memory address from which the argument is fetched or stored or the address from which execution is to proceed. The effective address can be specified within an instruction, or it can be calculated.

4.5.1 Immediate

In the immediate addressing mode, an argument is contained in the byte(s) immediately following the opcode. The number of bytes following the opcode matches the size of the register or memory location being operated on. There are 2-, 3-, and 4- (if prebyte is required) byte immediate instructions. The effective address is the address of the byte following the instruction.

4.5.2 Direct

In the direct addressing mode, the low-order byte of the operand address is contained in a single byte following the opcode, and the high-order byte of the address is assumed to be \$00. Addresses \$00–\$FF are thus accessed directly, using 2-byte instructions. Execution time is reduced by eliminating the additional memory access required for the high-order address byte. In most applications, this 256-byte area is reserved for frequently referenced data. In M68HC11 MCUs, the memory map can be configured for combinations of internal registers, RAM, or external memory to occupy these addresses.

5.3.2 Memory Map

After reset, the INIT register is initialized to \$01, mapping the RAM at \$00 and the control registers at \$1000.

For the MC68HC811E2, the CONFIG register resets to \$FF. EEPROM mapping bits (EE[3:0]) place the EEPROM at \$F800. Refer to the memory map diagram for MC68HC811E2 in Chapter 2 Operating Modes and On-Chip Memory.

5.3.3 Timer

During reset, the timer system is initialized to a count of \$0000. The prescaler bits are cleared, and all output compare registers are initialized to \$FFFF. All input capture registers are indeterminate after reset. The output compare 1 mask (OC1M) register is cleared so that successful OC1 compares do not affect any I/O pins. The other four output compares are configured so that they do not affect any I/O pins on successful compares. All input capture edge-detector circuits are configured for capture disabled operation. The timer overflow interrupt flag and all eight timer function interrupt flags are cleared. All nine timer interrupts are disabled because their mask bits have been cleared.

The I4/O5 bit in the PACTL register is cleared to configure the I4/O5 function as OC5; however, the OM5:OL5 control bits in the TCTL1 register are clear so OC5 does not control the PA3 pin.

5.3.4 Real-Time Interrupt (RTI)

The real-time interrupt flag (RTIF) is cleared and automatic hardware interrupts are masked. The rate control bits are cleared after reset and can be initialized by software before the real-time interrupt (RTI) system is used.

5.3.5 Pulse Accumulator

The pulse accumulator system is disabled at reset so that the pulse accumulator input (PAI) pin defaults to being a general-purpose input pin.

5.3.6 Computer Operating Properly (COP)

The COP watchdog system is enabled if the NOCOP control bit in the CONFIG register is cleared and disabled if NOCOP is set. The COP rate is set for the shortest duration timeout.

5.3.7 Serial Communications Interface (SCI)

The reset condition of the SCI system is independent of the operating mode. At reset, the SCI baud rate control register (BAUD) is initialized to \$04. All transmit and receive interrupts are masked and both the transmitter and receiver are disabled so the port pins default to being general-purpose I/O lines. The SCI frame format is initialized to an 8-bit character size. The send break and receiver wakeup functions are disabled. The TDRE and TC status bits in the SCI status register (SCSR) are both 1s, indicating that there is no transmit data in either the transmit data register or the transmit serial shift register. The RDRF, IDLE, OR, NF, FE, PF, and RAF receive-related status bits in the SCI control register 2 (SCCR2) are cleared.

5.3.8 Serial Peripheral Interface (SPI)

The SPI system is disabled by reset. The port pins associated with this function default to being general-purpose I/O lines.

5.5.4 Software Interrupt (SWI)

SWI is an instruction, and thus cannot be interrupted until complete. SWI is not inhibited by the global mask bits in the CCR. Because execution of SWI sets the I mask bit, once an SWI interrupt begins, other interrupts are inhibited until SWI is complete, or until user software clears the I bit in the CCR.

5.5.5 Maskable Interrupts

The maskable interrupt structure of the MCU can be extended to include additional external interrupt sources through the $\overline{\text{IRQ}}$ pin. The default configuration of this pin is a low-level sensitive wired-OR network. When an event triggers an interrupt, a software accessible interrupt flag is set. When enabled, this flag causes a constant request for interrupt service. After the flag is cleared, the service request is released.

5.5.6 Reset and Interrupt Processing

Figure 5-5 and Figure 5-6 illustrate the reset and interrupt process. Figure 5-5 illustrates how the CPU begins from a reset and how interrupt detection relates to normal opcode fetches. Figure 5-6 is an expansion of a block in Figure 5-5 and illustrates interrupt priorities. Figure 5-7 shows the resolution of interrupt sources within the SCI subsystem.

5.6 Low-Power Operation

Both stop mode and wait mode suspend CPU operation until a reset or interrupt occurs. Wait mode suspends processing and reduces power consumption to an intermediate level. Stop mode turns off all on-chip clocks and reduces power consumption to an absolute minimum while retaining the contents of the entire RAM array.

5.6.1 Wait Mode

The WAI opcode places the MCU in wait mode, during which the CPU registers are stacked and CPU processing is suspended until a qualified interrupt is detected. The interrupt can be an external $\overline{\text{IRQ}}$, an XIRQ, or any of the internally generated interrupts, such as the timer or serial interrupts. The on-chip crystal oscillator remains active throughout the wait standby period.

The reduction of power in the wait condition depends on how many internal clock signals driving on-chip peripheral functions can be shut down. The CPU is always shut down during wait. While in the wait state, the address/data bus repeatedly runs read cycles to the address where the CCR contents were stacked. The MCU leaves the wait state when it senses any interrupt that has not been masked.

The free-running timer system is shut down only if the I bit is set to 1 and the COP system is disabled by NOCOP being set to 1. Several other systems also can be in a reduced power-consumption state depending on the state of software-controlled configuration control bits. Power consumption by the analog-to-digital (A/D) converter is not affected significantly by the wait condition. However, the A/D converter current can be eliminated by writing the ADPU bit to 0. The SPI system is enabled or disabled by the SPE control bit. The SCI transmitter is enabled or disabled by the TE bit, and the SCI receiver is enabled or disabled by the RE bit. Therefore, the power consumption in wait is dependent on the particular application.

Chapter 8

Serial Peripheral Interface (SPI)

8.1 Introduction

The serial peripheral interface (SPI), an independent serial communications subsystem, allows the MCU to communicate synchronously with peripheral devices, such as:

- Frequency synthesizers
- Liquid crystal display (LCD) drivers
- Analog-to-digital (A/D) converter subsystems
- Other microprocessors

The SPI is also capable of inter-processor communication in a multiple master system. The SPI system can be configured as either a master or a slave device. When configured as a master, data transfer rates can be as high as one-half the E-clock rate (1.5 Mbits per second for a 3-MHz bus frequency). When configured as a slave, data transfers can be as fast as the E-clock rate (3 Mbits per second for a 3-MHz bus frequency).

8.2 Functional Description

The central element in the SPI system is the block containing the shift register and the read data buffer. The system is single buffered in the transmit direction and double buffered in the receive direction. This means that new data for transmission cannot be written to the shifter until the previous transfer is complete; however, received data is transferred into a parallel read data buffer so the shifter is free to accept a second serial character. As long as the first character is read out of the read data buffer before the next serial character is ready to be transferred, no overrun condition occurs. A single MCU register address is used for reading data from the read data buffer and for writing data to the shifter.

The SPI status block represents the SPI status functions (transfer complete, write collision, and mode fault) performed by the serial peripheral status register (SPSR). The SPI control block represents those functions that control the SPI system through the serial peripheral control register (SPCR).

Refer to Figure 8-1, which shows the SPI block diagram.

8.3 SPI Transfer Formats

During an SPI transfer, data is simultaneously transmitted and received. A serial clock line synchronizes shifting and sampling of the information on the two serial data lines. A slave select line allows individual selection of a slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. On a master SPI device, the select line can optionally be used to indicate a multiple master bus contention. Refer to Figure 8-2.

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

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.

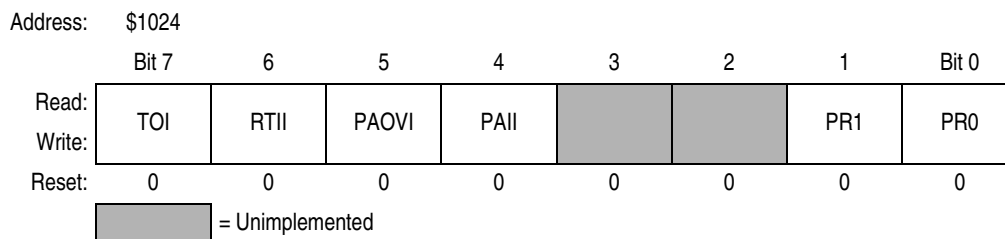


Figure 9-19. Timer Interrupt Mask 2 Register (TMSK2)

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

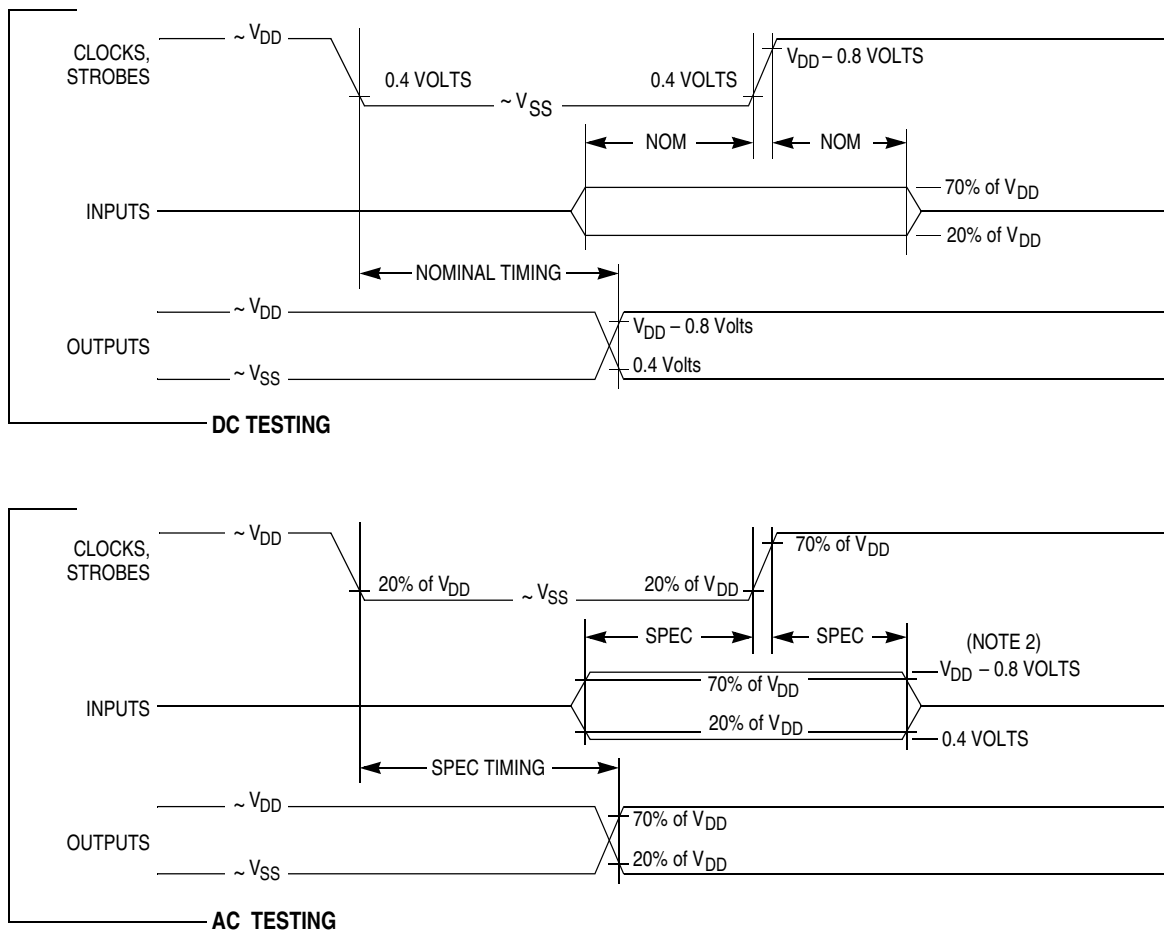
These bits are used to select the prescaler divide-by ratio. In normal modes, PR[1:0] can be written only once, and the write must be within 64 cycles after reset. Refer to Table 9-1 and Table 9-4 for specific timing values.

Table 9-4. Timer Prescale

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

NOTE

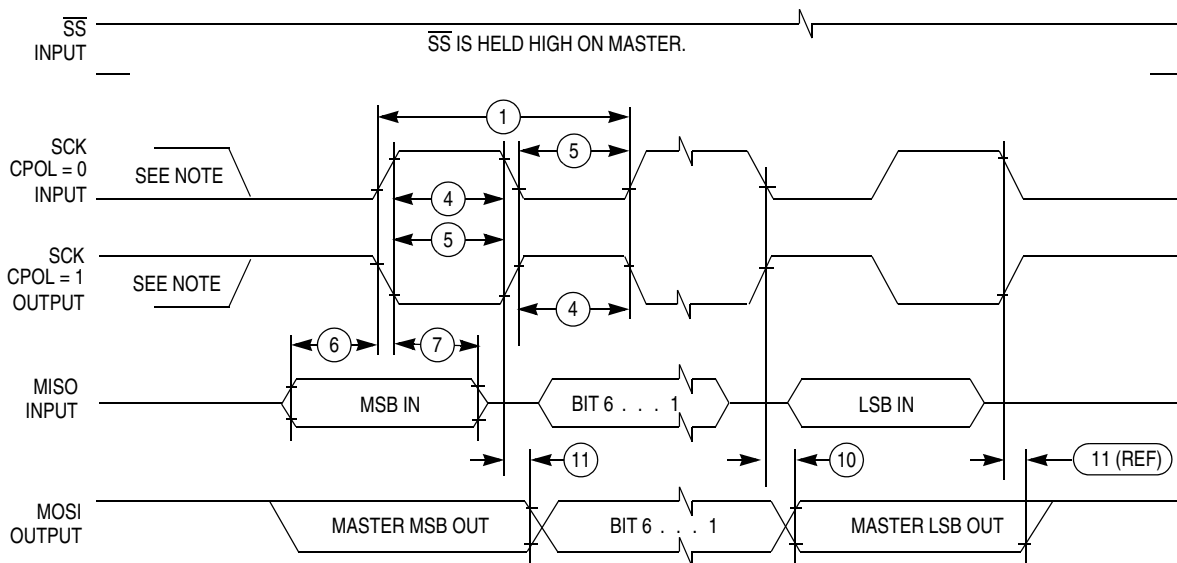
Bits in TMSK2 correspond bit for bit with flag bits in TFLG2. Bits in TMSK2 enable the corresponding interrupt sources.



Notes:

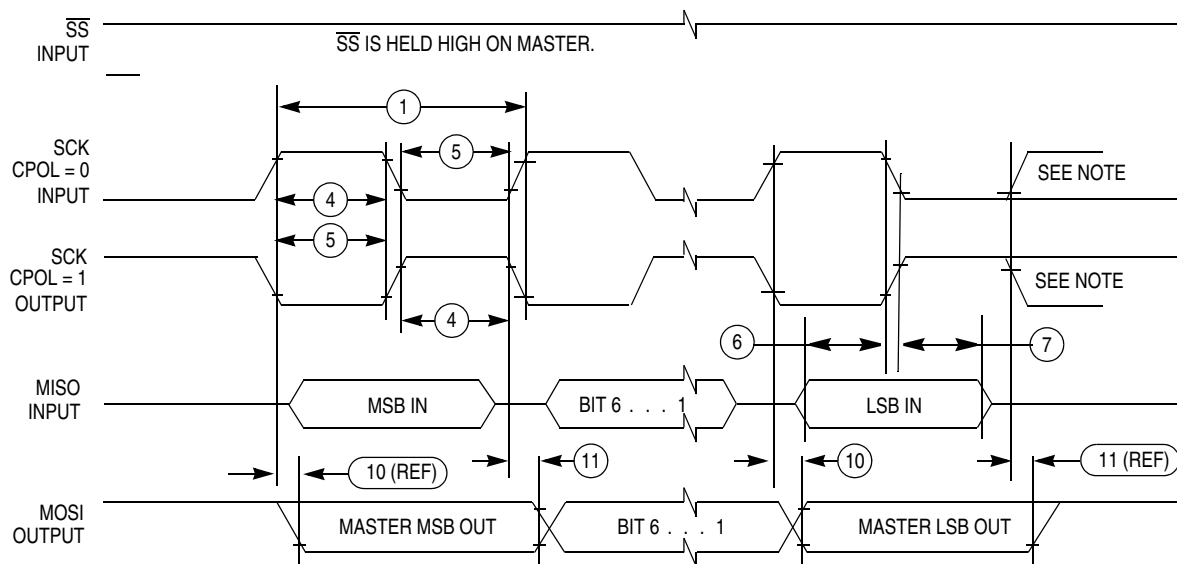
1. Full test loads are applied during all dc electrical tests and ac timing measurements.
2. During ac timing measurements, inputs are driven to 0.4 volts and $V_{DD} - 0.8 \text{ volts}$ while timing measurements are taken at 20% and 70% of V_{DD} points.

Figure 10-1. Test Methods



Note: This first clock edge is generated internally but is not seen at the SCK pin.

A) SPI Master Timing (CPHA = 0)

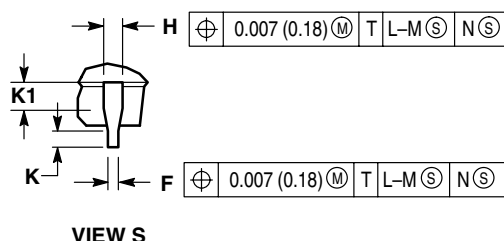
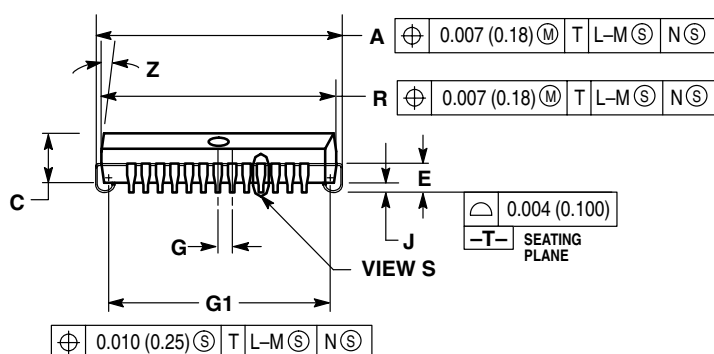
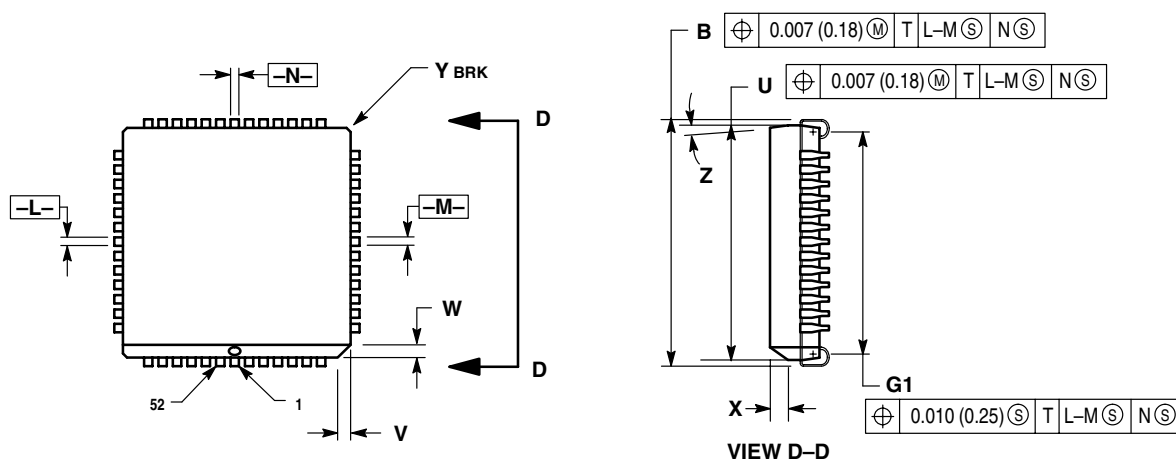


Note: This first clock edge is generated internally but is not seen at the SCK pin.

B) SPI Master Timing (CPHA = 1)

Figure 10-15. SPI Timing Diagram (Sheet 1 of 2)

11.5 52-Pin Plastic-Leaded Chip Carrier (Case 778)

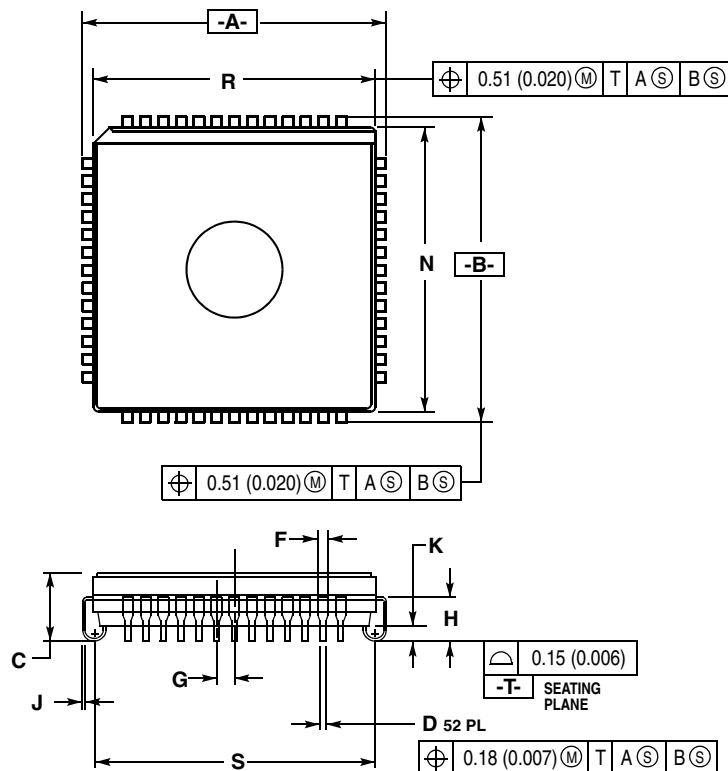


NOTES:

1. DATUMS -L-, -M-, AND -N- DETERMINED WHERE TOP OF LEAD SHOULDER EXITS PLASTIC BODY AT MOLD PARTING LINE.
2. DIMENSION G1, TRUE POSITION TO BE MEASURED AT DATUM -T-, SEATING PLANE.
3. DIMENSIONS R AND U DO NOT INCLUDE MOLD FLASH. ALLOWABLE MOLD FLASH IS 0.010 (0.250) PER SIDE.
4. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
5. CONTROLLING DIMENSION: INCH.
6. THE PACKAGE TOP MAY BE SMALLER THAN THE PACKAGE BOTTOM BY UP TO 0.012 (0.300). DIMENSIONS R AND U ARE DETERMINED AT THE OUTERMOST EXTREMES OF THE PLASTIC BODY EXCLUSIVE OF MOLD FLASH, TIE BAR BURRS, GATE BURRS AND INTERLEAD FLASH, BUT INCLUDING ANY MISMATCH BETWEEN THE TOP AND BOTTOM OF THE PLASTIC BODY.
7. DIMENSION H DOES NOT INCLUDE DAMBAR PROTRUSION OR INTRUSION. THE DAMBAR PROTRUSION(S) SHALL NOT CAUSE THE H DIMENSION TO BE GREATER THAN 0.037 (0.940). THE DAMBAR INTRUSION(S) SHALL NOT CAUSE THE H DIMENSION TO BE SMALLER THAN 0.025 (0.635).

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.785	0.795	19.94	20.19
B	0.785	0.795	19.94	20.19
C	0.165	0.180	4.20	4.57
E	0.090	0.110	2.29	2.79
F	0.013	0.019	0.33	0.48
G	0.050 BSC		1.27 BSC	
H	0.026	0.032	0.66	0.81
J	0.020	—	0.51	—
K	0.025	—	0.64	—
R	0.750	0.756	19.05	19.20
U	0.750	0.756	19.05	19.20
V	0.042	0.048	1.07	1.21
W	0.042	0.048	1.07	1.21
X	0.042	0.056	1.07	1.42
Y	—	0.020	—	0.50
Z	2°	10°	2°	10°
G1	0.710	0.730	18.04	18.54
K1	0.040	—	1.02	—

11.6 52-Pin Windowed Ceramic-Leaded Chip Carrier (Case 778B)



NOTES:

1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: INCH.
3. DIMENSION R AND N DO NOT INCLUDE GLASS PROTRUSION. GLASS PROTRUSION TO BE 0.25 (0.010) MAXIMUM.
4. ALL DIMENSIONS AND TOLERANCES INCLUDE LEAD TRIM OFFSET AND LEAD

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.785	0.795	19.94	20.19
B	0.785	0.795	19.94	20.19
C	0.165	0.200	4.20	5.08
D	0.017	0.021	0.44	0.53
F	0.026	0.032	0.67	0.81
G	0.050 BSC		1.27 BSC	
H	0.090	0.130	2.29	3.30
J	0.006	0.010	0.16	0.25
K	0.035	0.045	0.89	1.14
N	0.735	0.756	18.67	19.20
R	0.735	0.756	18.67	19.20
S	0.690	0.730	17.53	18.54

Appendix B

EVBU Schematic

Refer to Figure B-1 for a schematic diagram of the M68HC11EVBU Universal Evaluation Board. This diagram is included for reference only.

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

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

\$FF Character Is Required before Loading into RAM

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

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

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

Variable-Length Download

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

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

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

EPROM/OTP Versions of M68HC11 Have an EPROM Emulation Mode

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

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

How to Reach Us:

Home Page:

www.freescale.com

E-mail:

support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

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