MC68HC11PH8/D

HC11

MC68HC11PH8
MC68HC711PH8

TECHNICAL
DATA

MOTOROLA

MC68HC11PH8

TECHNICAL DATA

MOTOROLA

MC68HC11PH8
MC68HC711PH8

High-density Complementary Metal Oxide Semiconductor (HCMOS) Microcomputer Unit

All Trade Marks recognized. This document contains information on new products. Specifications and information herein are subject to change without notice.

All products are sold on Motorola's Terms & Conditions of Supply. In ordering a product covered by this document the Customer agrees to be bound by those Terms & Conditions and nothing contained in this document constitutes or forms part of a contract (with the exception of the contents of this Notice). A copy of Motorola's Terms & Conditions of Supply is available on request.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola 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 can and do vary in different applications. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola 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 Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola 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 Motorola was negligent regarding the design or manufacture of the part. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

The Customer should ensure that it has the most up to date version of the document by contacting its local Motorola office. This document supersedes any earlier documentation relating to the products referred to herein. The information contained in this document is current at the date of publication. It may subsequently be updated, revised or withdrawn.

Conventions

Abbreviations

Definitions of any abbreviations used can be found in the Glossary.

Text in italics

This document contains information referring to the MC68HC11PH8 and the MC68HC711PH8. All references to the MC68HC11PH8 apply equally to the MC68HC711PH8, unless otherwise noted. *References specific to the MC68HC711PH8 are italicised in the text.*

Register tables

Because the bits in any one register are not necessarily linked by a common function, the description of a register may appear in several sections referring to different aspects of device operation. A full description of a bit is given only in a section in which it has relevance. Elsewhere, it appears shaded in the register diagram and is only briefly described.

State on reset

x	state of bit on reset depends on factors such as operating mode
u	state of bit on reset is undefined

– Cut along this line to remove –

CUSTOMER FEEDBACK QUESTIONNAIRE (MC68HC11PH8/D Rev 1)

Motorola wishes to continue to improve the quality of its documentation. We would welcome your feedback on the publication you have just received. Having used the document, please complete this card (or a photocopy of it, if you prefer).

1. How would you rate the quality of the document? Check one box in each category.

	Excellent			Poor		Excellent			Poor
Organization	□	□	□	□	Tables	□	□	□	□
Readability	□	□	□	□	Table of contents	□	□	□	□
Understandability	□	□	□	□	Index	□	□	□	□
Accuracy	□	□	□	□	Page size/binding	□	□	□	□
Illustrations	□	□	□	□	Overall impression	□	□	□	□

Comments:

2. What is your intended use for this document? If more than one option applies, please rank them (1, 2, 3).

Selection of device for new application □ Other □ Please specify:

System design □

Training purposes □

3. How well does this manual enable you to perform the task(s) outlined in question 2?

Completely			Not at all
□	□	□	□

Comments:

4. How easy is it to find the information you are looking for?

Easy			Difficult
□	□	□	□

Comments:

5. Is the level of technical detail in the following sections sufficient to allow you to understand how the device functions?

		Too little detail				Too much detail
SECTION 1	INTRODUCTION	□	□	□	□	□
SECTION 2	PIN DESCRIPTIONS	□	□	□	□	□
SECTION 3	OPERATING MODES AND ON-CHIP MEMORY	□	□	□	□	□
SECTION 4	PARALLEL INPUT/OUTPUT	□	□	□	□	□
SECTION 5	SERIAL COMMUNICATIONS INTERFACE	□	□	□	□	□
SECTION 6	MOTOROLA INTERCONNECT BUS (MI BUS)	□	□	□	□	□
SECTION 7	SERIAL PERIPHERAL INTERFACE	□	□	□	□	□
SECTION 8	TIMING SYSTEM	□	□	□	□	□
SECTION 9	ANALOG-TO-DIGITAL CONVERTER	□	□	□	□	□
SECTION 10	RESETS AND INTERRUPTS	□	□	□	□	□
SECTION 11	CPU CORE AND INSTRUCTION SET	□	□	□	□	□
APPENDIX A	ELECTRICAL SPECIFICATIONS (STANDARD)	□	□	□	□	□
APPENDIX B	MECHANICAL DATA AND ORDERING INFORMATION	□	□	□	□	□
APPENDIX C	DEVELOPMENT SUPPORT	□	□	□	□	□
		□	□	□	□	□

Comments:

6. Have you found any errors? If so, please comment:

7. From your point of view, is anything missing from the document? If so, please say what:

8. How could we improve this document? ________________________________

____

____

9. How would you rate Motorola’s documentation?

	Excellent			Poor
– In general	□	□	□	□
– Against other semiconductor suppliers	□	□	□	□

10. Which semiconductor manufacturer provides the best technical documentation? ________________

11. Which company (in any field) provides the best technical documentation? ________________

12. How many years have you worked with microprocessors?

Less than 1 year □ 1–3 years □ 3–5 years □ More than 5 years □

– Second fold back along this line –

By air mail
Par avion

IBRS NUMBER PHQ-B/207/G
CCRI NUMERO PHQ-B/207/G

NE PAS AFFRANCHIR

NO STAMP REQUIRED

REPONSE PAYEE
GRANDE-BRETAGNE

Motorola Ltd.,
Colvilles Road,
Kelvin Industrial Estate,
EAST KILBRIDE,
G75 8BR.
GREAT BRITAIN.

F.A.O. Technical Publications Manager
(re: MC68HC11PH8/D Rev 1)

MOTOROLA LTD.
Semiconductor Products Sector

– First fold back along this line –

– Cut along this line to remove –

– Third fold back along this line –

13. Currently there is some discussion in the semiconductor industry regarding a move towards providing data sheets in electronic form. If you have any opinion on this subject, please comment. ________________

____

14. We would be grateful if you would supply the following information (at your discretion), or attach your card.

Name: ________________ Phone No: ________________

Position: ________________ FAX No: ________________

Department: ________________________________

Company: ________________________________

Address: ________________________________

____

Thank you for helping us improve our documentation,
Graham Livey, Technical Publications Manager, Motorola Ltd., Scotland.

– Last tuck this edge into opposite flap –

TABLE OF CONTENTS

Paragraph Number | **Title** | Page Number

1 INTRODUCTION

2 PIN DESCRIPTIONS

3
OPERATING MODES AND ON-CHIP MEMORY

4
PARALLEL INPUT/OUTPUT

5
SERIAL COMMUNICATIONS INTERFACE

6
MOTOROLA INTERCONNECT BUS (MI BUS)

7
SERIAL PERIPHERAL INTERFACE

Paragraph
Number | **Title** | Page
Number

8
TIMING SYSTEM

9
ANALOG-TO-DIGITAL CONVERTER

10
RESETS AND INTERRUPTS

11
CPU CORE AND INSTRUCTION SET

A
ELECTRICAL SPECIFICATIONS (STANDARD)

B
MECHANICAL DATA AND ORDERING INFORMATION

C
DEVELOPMENT SUPPORT

GLOSSARY

INDEX

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

Figure Number	Title	Page Number

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

1
INTRODUCTION

The MC68HC11PH8 8-bit microcontroller is a member of the M68HC11 family of HCMOS microcontrollers. In addition to 48K bytes of ROM, the MC68HC11PH8 contains 2K bytes of RAM and 768 bytes of EEPROM. Making use of an 84-pin PLCC, or 112-pin TQFP package, a non-multiplexed expanded bus is a feature of this device. The timer system has been expanded to include three input captures, four output compares and a software selectable input capture or output compare. There are three 8-bit modulus timers, one of which may be used as a prescaler for the other two. The inclusion of a PLL circuit allows power consumption and performance to be adjusted to suit operational requirements. Other major features of this device are: 8-channel, 8-bit A/D converter, four PWM timer channels, wired-OR capability for keyboard interrupt, four LCD segment drivers and two SPI and two enhanced SCI subsystems. The MC68HC11PH8 is especially suitable for mobile communications and automotive applications.

The MC68HC711PH8 is an EPROM version of the MC68HC11PH8, with the user ROM replaced by a similar amount of EPROM. All references to the MC68HC11PH8 apply equally to the MC68HC711PH8, unless otherwise noted. *References specific to the MC68HC711PH8 are italicised in the text.*

1.1 Features

- Low power, high performance M68HC11 CPU core
- 48K bytes of user ROM (MC68HC11PH8); *48K bytes of user EPROM (MC68HC711PH8)*
- 2K bytes of RAM
- 768 bytes of byte-erasable user EEPROM, with on-chip charge pump
- Up to 54 general purpose I/O lines, plus up to 8 input-only lines
- Non-multiplexed address and data buses, permitting direct access to the full 64K address map
- 16-bit timer with 3/4 input captures and 4/5 output compares; pulse accumulator and COP watchdog timer
- Three 8-bit modulus timers, for generating periodic interrupts

- Power saving PLL circuit
- Wired-OR interrupt capability for keyboard support, allowing wake-up from STOP and WAIT modes
- Two 8- or 9-bit SCI subsystems, one with MI BUS[†] capability; both NRZ type for RS232 compatibility
- Two SPI subsystems, with software selectable MSB/LSB first option
- 8-channel, 8-bit analog-to-digital (A/D) converter
- Four 8-bit PWM timer channels (may be concatenated to form one or two 16-bit channels)
- 4-segment LCD driver
- Available in 84-pin PLCC or 112-pin TQFP packages (MC68HC11PH8); *also 84-pin CERQUAD package (MC68HC711PH8)*

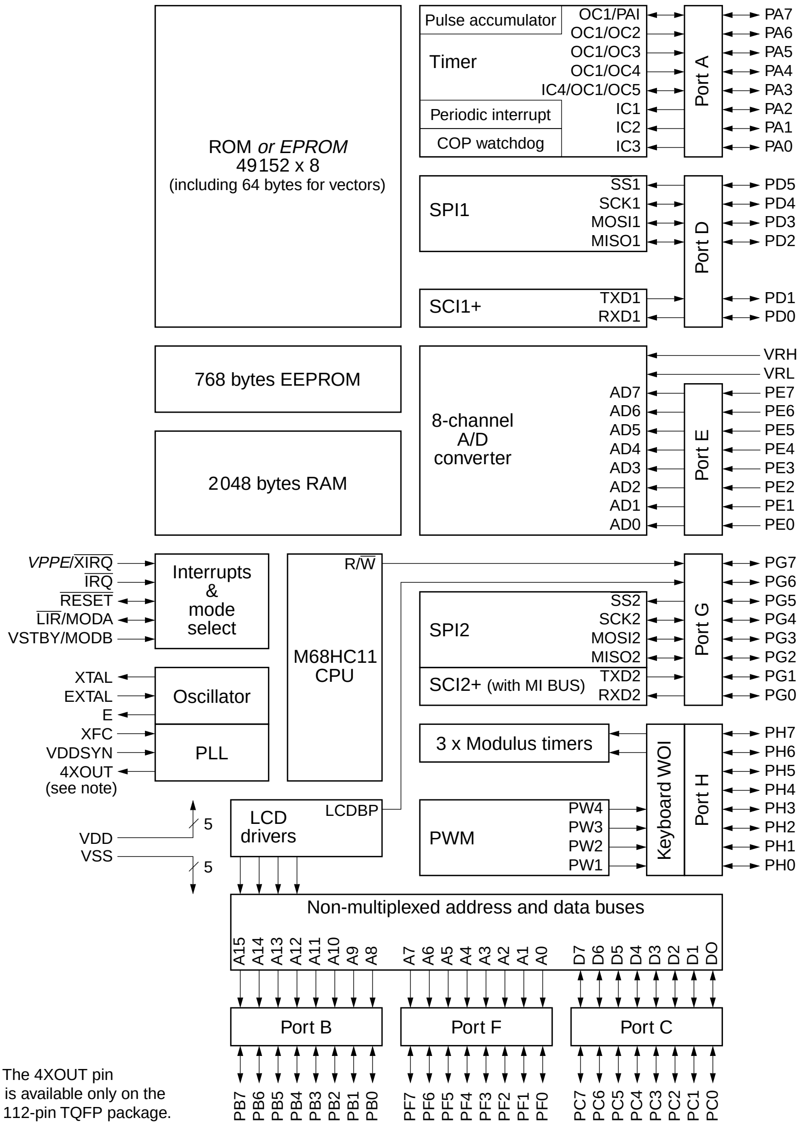
1.2 Mask options

There are five mask options available on the MC68HC11PH8. These options are programmed during manufacture and must be specified on the order form.

- Security option (available/unavailable); see Section 3.4.4
- PLL oscillator frequency (32 kHz/614 kHz); see Section 2.5
- Oscillator buffer type (inverter/Schmitt trigger); see Section 2.3
- POR/exit from STOP start-up time (4064/128 bus cycles); see Section 3.3.2.4
- ROMON bit software switchable in user expanded mode (enable/disable); see Section 3.3.2.1

Note: *These options are not available on the MC68HC711PH8; on this device, the security option is always available, the PLL oscillator is optimized for operation at 32 kHz, the oscillator buffer is an inverter, the POR/exit from STOP start-up time is 4064 bus cycles and the ROMON bit is software switchable in user expanded mode.*

[†] The Motorola Interconnect Bus (MI BUS) is a serial communications protocol which supports distributed real-time control efficiently and with a high degree of noise immunity. It allows data to be transferred between the MCU and the slave device using only one wire, making this type of communication suitable for medium speed networks requiring very low cost multiplex wiring.



Note: The 4XOUT pin
is available only on the
112-pin TQFP package.

Figure 1-1 MC68HC11PH8/*MC68HC711PH8* block diagram

THIS PAGE INTENTIONALLY LEFT BLANK

2
PIN DESCRIPTIONS

The MC68HC11PH8 is available in an 84-pin plastic-leaded chip carrier (PLCC) and in an 112-pin thin quad flat pack (TQFP); *in addition to those two packages, the MC68HC711PH8 is also available in an 84-pin windowed cerquad package, to allow the EPROM to be erased.* Most pins on this MCU serve two or more functions, as described in the following paragraphs. Refer to Figure 2-1 and to Figure 2-2 which show the pin assignments for the 84 and 112-pin packages respectively.

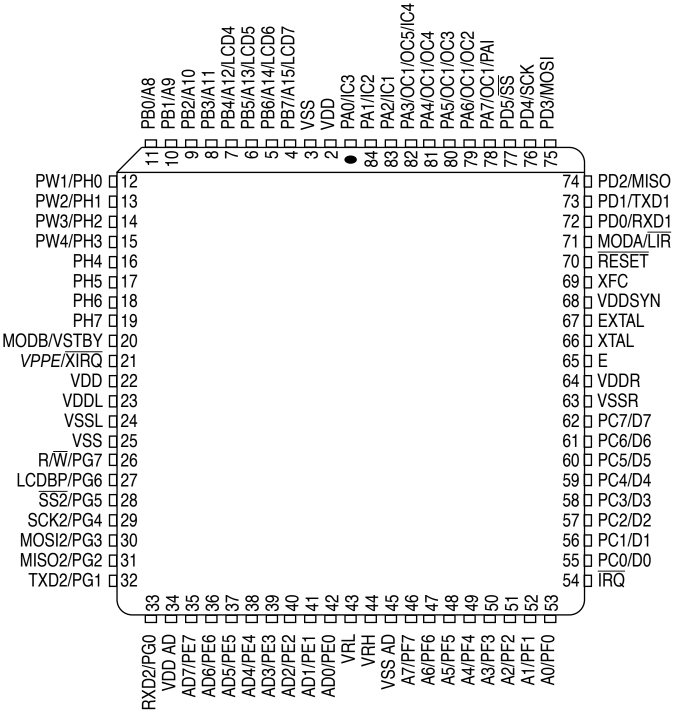


Figure 2-1 84-pin PLCC/*CERQUAD* pinout

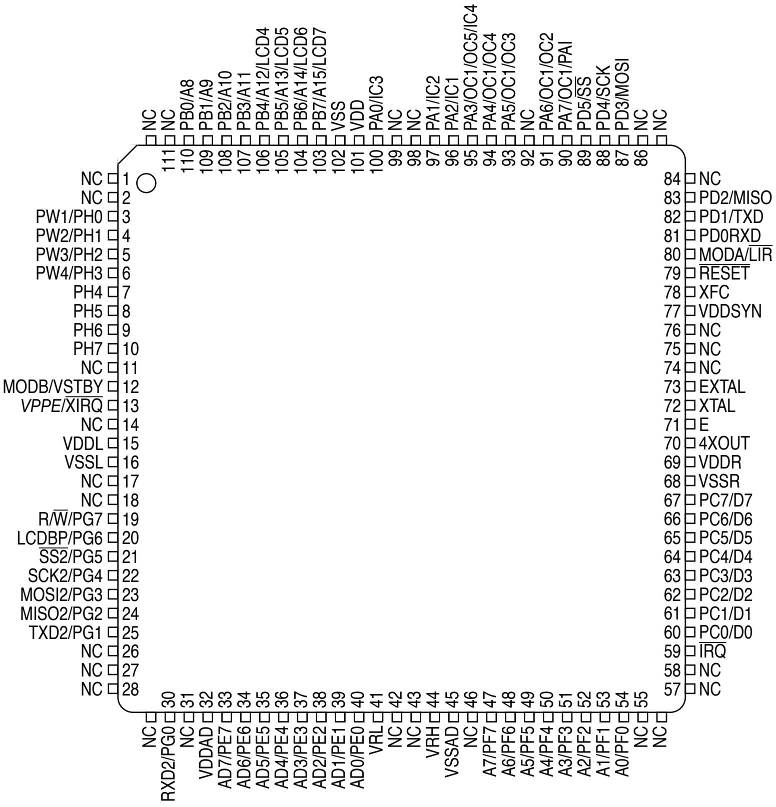


Figure 2-2 112-pin TQFP pinout

2.1 VDD and VSS

Power is supplied to the microcontroller via these pins. VDD is the positive supply and VSS is ground. The MCU operates from a single 5V (nominal) power supply.

It is in the nature of CMOS designs that very fast signal transitions occur on the MCU pins. These short rise and fall times place very high short-duration current demands on the power supply. To prevent noise problems, special care must be taken to provide good power supply bypassing at the MCU. Bypass capacitors should have good high-frequency characteristics and be as close to the MCU as possible. Bypassing requirements vary, depending on how heavily the MCU pins are loaded.

The MC68HC11PH8 MCU has five VDD pins and five VSS pins. One pair of these pins is reserved for supplying power to the analog-to-digital converter (VDD AD, VSS AD); two pairs are used for the internal logic (VDD, VSS); the remaining two pairs supply power for the port logic on either half of the chip (VDDL, VSSL and VDDR, VSSR). This arrangement minimizes the injection of noise into the digital circuitry on the chip.

2.2 $\overline{\text{RESET}}$

An active low bidirectional control signal, $\overline{\text{RESET}}$, acts as an input to initialize the MCU to a known start-up state. It also acts as an open-drain output to indicate that an internal failure has been detected in either the clock monitor or the COP watchdog circuit. The CPU distinguishes between internal and external reset conditions by sensing whether the reset pin rises to a logic one in less than four E clock cycles after an internal reset has been released. It is therefore not advisable to connect an external resistor-capacitor (RC) power-up delay circuit to the reset pin of M68HC11 devices because the circuit charge time constant can cause the device to misinterpret the type of reset that occurred. Refer to Section 10 for further information.

Figure 2-3 illustrates a typical reset circuit that includes an external switch together with a low voltage inhibit circuit, to prevent power transitions, or RAM or EEPROM corruption.

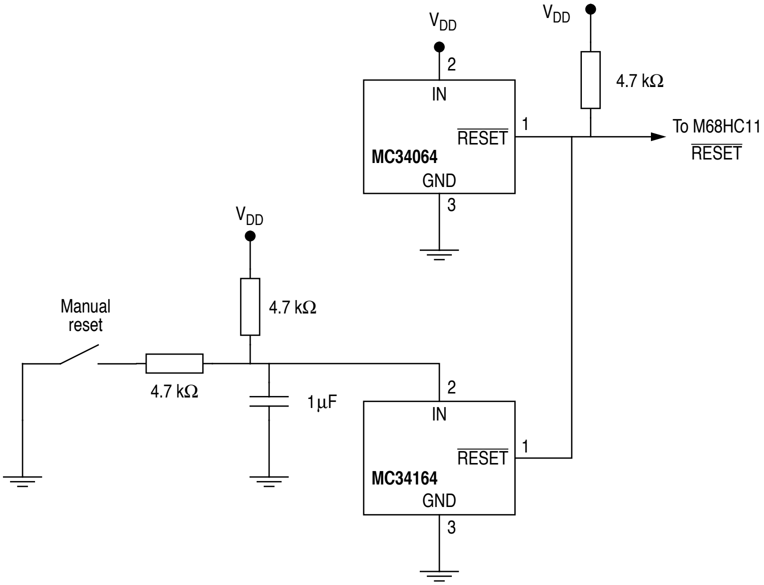


Figure 2-3 External reset circuitry

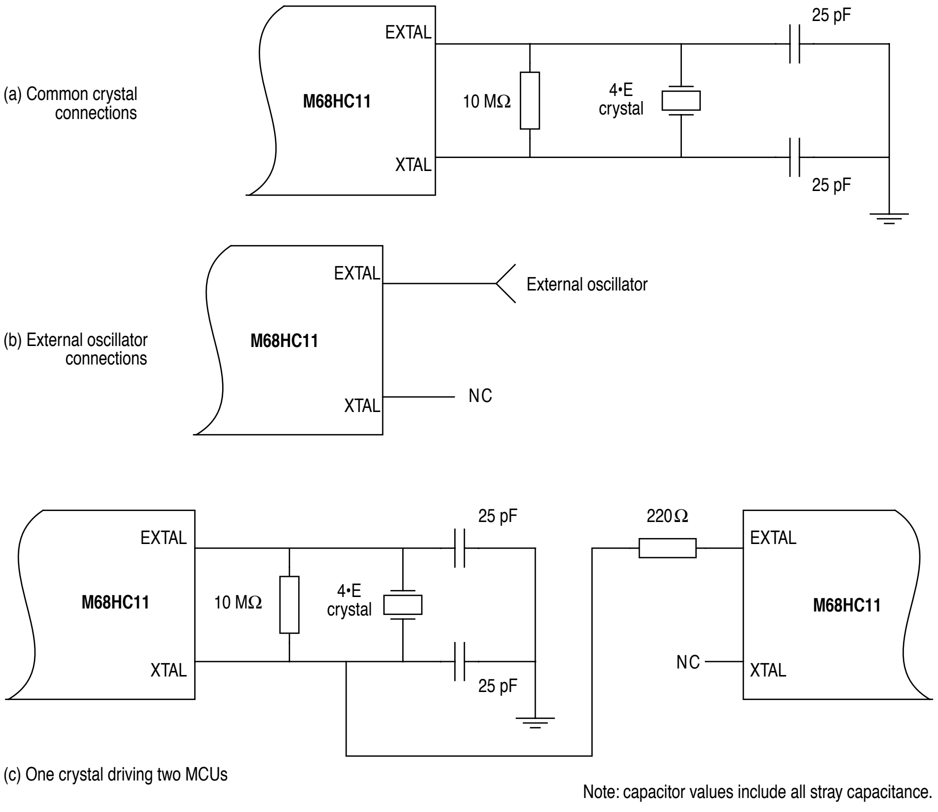
2.3 Crystal driver and external clock input (XTAL, EXTAL)

These two pins provide the interface for either a crystal or a CMOS compatible clock to control the internal clock generator circuitry. If the PLL circuit is not being used to provide the E clock, the frequency applied to these pins must be four times higher than the desired E clock rate. Figure 2-4 shows oscillator connections that should be used when the PLL is disabled, and Figure 2-5 shows the connections that should be used when the PLL is enabled.

The XTAL pin is normally left unconnected when an external CMOS compatible clock input is connected to the EXTAL pin. However, a 10 kΩ to 100 kΩ load resistor connected from XTAL to ground can be used to reduce RFI noise emission. The XTAL output is normally intended to drive only a crystal. The XTAL output can be buffered with a high-impedance buffer, or it can be used to drive the EXTAL input of another M68HC11 family device (unless the PLL circuit is in use, in which case the 4XOUT output must be used to clock a second device; see Section 2.5).

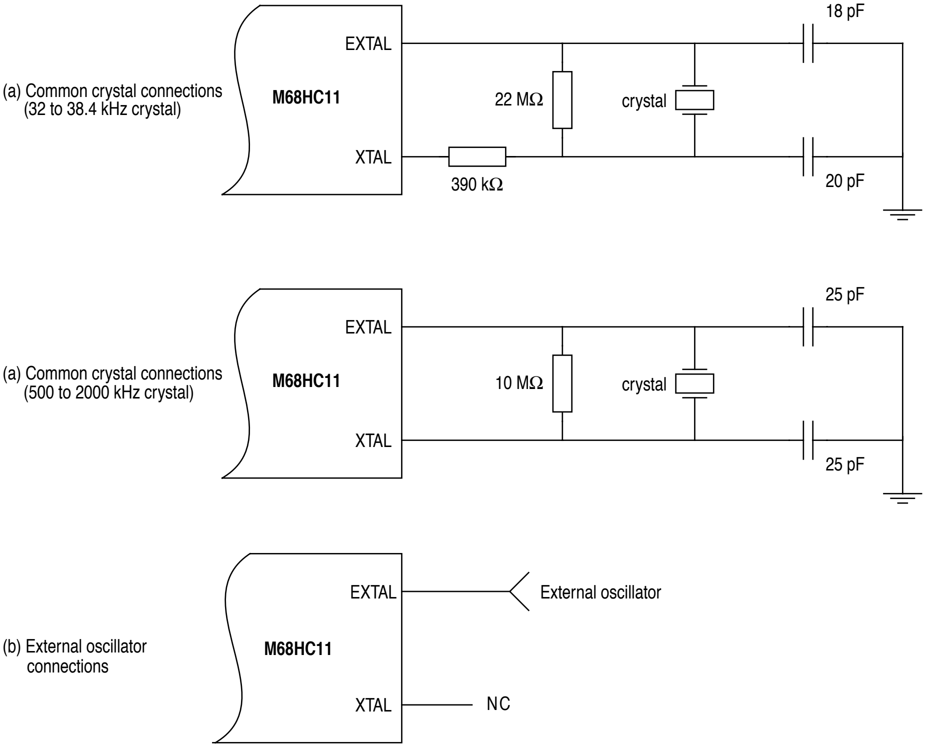
On the MC68HC11PH8, the type of internal crystal oscillator buffer is determined by a mask option; it can be either an inverter or a Schmitt trigger. Use of the Schmitt trigger type reduces problems caused by noise, in particular with slow clocks. At crystal power-up, the Schmitt trigger will only generate internal clocks when the crystal amplitude is sufficient. However, this type of buffer requires a larger XTAL amplitude and is not recommended for use with high frequency crystals, especially if a second MCU is to be driven. *This option is not available on the MC68HC711PH8, on which the crystal oscillator buffer is an inverter.*

In all cases, use caution when designing circuitry associated with the oscillator pins.



(a) Common crystal connections
M68HC11
EXTAL
XTAL
10 MΩ
4•E crystal
25 pF
25 pF
(b) External oscillator connections
M68HC11
EXTAL
XTAL
External oscillator
NC
(c) One crystal driving two MCUs
M68HC11
EXTAL
XTAL
10 MΩ
4•E crystal
25 pF
25 pF
220Ω
EXTAL
M68HC11
NC
XTAL
Note: capacitor values include all stray capacitance.

Figure 2-4 Oscillator connections (VDDSYN = 0, PLL disabled)



Note: capacitor values include all stray capacitance.
Note: all values of capacitance and resistance shown are approximate; exact values must be calculated knowing the crystal parameters and the expected voltage and temperature ranges.

Figure 2-5 Oscillator connections (VDDSYN = 1, PLL enabled)

2.4 E clock output (E)

E is the output connection for the internally generated E clock. The signal from E is used as a timing reference. The frequency of the E clock output is one quarter that of the input frequency at the XTAL and EXTAL pins (except when the PLL is used as the clock source). When E clock output is low, an internal process is taking place; when it is high, data is being accessed. All clocks, including the E clock, are halted when the MCU is in STOP mode. The E clock output can be turned off to reduce the effects of RFI (see Section 3.3.2.5).

2.5 Phase-locked loop (XFC, VDDSYN, 4XOUT)

The XFC and VDDSYN pins are the inputs for the on-chip PLL (phase-locked loop) circuitry. On reset, all system clocks are derived from the internal EXTAL signal (EXTALi). If enabled (VDDSYN high), the PLL uses the EXTALi frequency as a reference to generate a clock frequency that can be varied under software control. The user may choose to use the PLL output instead of EXTALi as the source clock for the system.

The PLL consists of a variable bandwidth loop filter, a voltage controlled oscillator (VCO), a feedback frequency divider and a digital phase detector. PLL functions are controlled by the PLLCR and SYNR registers. A block diagram of the PLL circuit is shown in Figure 2-6; refer also to Figure 8-1.

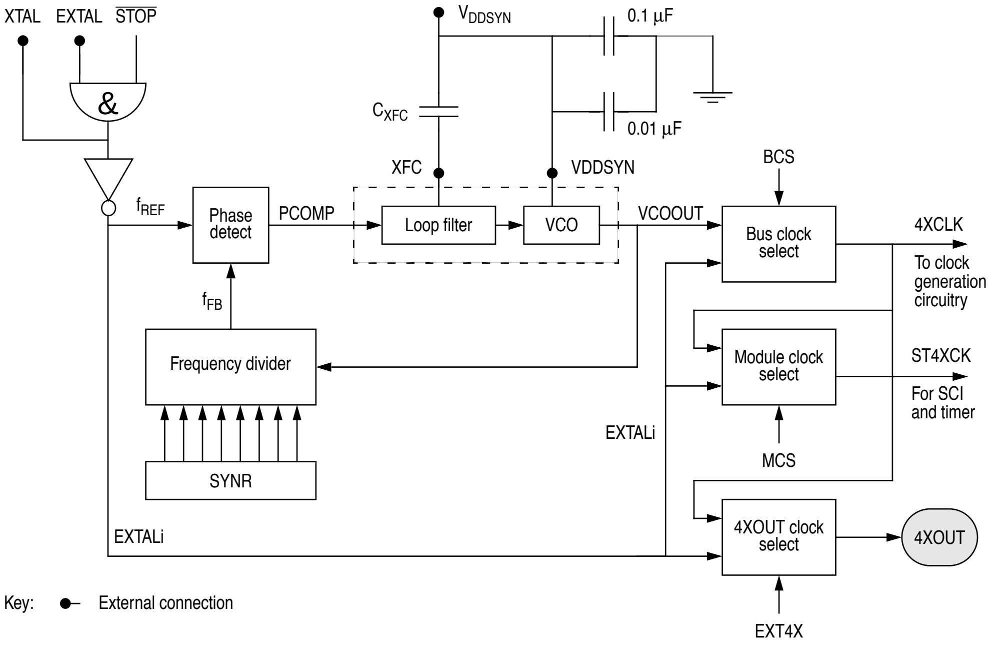


Figure 2-6 PLL circuit

If enabled by the CLK4X bit in the CONFIG register, either the 4XCLK signal or the EXTALi signal can be output on the 4XOUT pin, depending on the state of the EXT4X bit in the OPT2 register. Refer to Figure 2-6, and to Section 3 for a description of the CLK4X and EXT4X bits. The signal output on the 4XOUT pin could be used to clock another MCU.

Note: The 4XOUT pin is not available on 84-pin packaged devices.

2.5.1 PLL operation

The voltage controlled oscillator (VCO) generates the PLL output frequency VCOOUT. This signal is fed back through a frequency divider, which divides the signal frequency by a factor determined by the contents of the SYNR register, to produce the feedback signal f_{FB}. This signal is input to the phase detector along with the reference signal, f_{REF}. The phase detector generates a control signal (PCOMP) which is a function of the phase difference between f_{FB} and f_{REF}. PCOMP is then integrated, and the resultant dc voltage (visible on XFC) is applied to the VCO, modifying the output signal VCOOUT to lock it in phase with f_{REF}.

Note: Because the operation of the PLL depends on repeated adjustments to the voltage input to the VCO, a time t_{PLLS} is required for the stabilization of the output frequency.

The state of two bits in the PLLCR register, MCS and BCS, determine whether VCOOUT or EXTALi is used for the system clocks.

A mask option on the MC68HC11PH8 allows the PLL circuit to be optimized for operation in either of two frequency ranges, as shown in Table 2-1 *(this option is not available on the MC68HC711PH8; on this device the PLL is optimized for operation at 32kHz)*. Input frequencies other than those included in Table 2-1 can be used, but, for operation above the maximum frequency specified, VDDSYN should be grounded to disable the PLL and enable the high frequency oscillator circuit; in this state the oscillator is designed for 16MHz operation and XFC may be left unconnected. Refer also to Figure 2-5.

Table 2-1 PLL mask options

Characteristic	Mask option 1	Mask option 2
Typical input frequency	32 kHz	614 kHz
Maximum input frequency	50 kHz	2 MHz

VDDSYN is the power supply pin for the PLL and should be suitably bypassed. Connecting it high enables the internal low frequency oscillator circuitry designed for the PLL. The external capacitor on XFC (C_{XFC}) should be located as close to the chip as possible to minimize noise. In general, a larger capacitor will improve the PLL's frequency stability, at the expense of increasing the time required for it to settle (t_{PLLS}) at the desired frequency. A capacitor value of 47nF is usually adequate for either 32kHz or 614kHz applications. Refer to Section A.5.2 for PLL control timing information.

The PLL filter has two bandwidths that can be manually selected under control of the BWC bit in PLLCR. Whenever the PLL is first enabled, the wide bandwidth mode should be used, to enable the PLL frequency to ramp up quickly. After a time t_{PLLS} has elapsed, the filter can be switched to the narrow bandwidth mode, to make the final frequency more stable.

Warning: Bit 5 of the PLLCR (AUTO) must be cleared before an attempt is made to use BWC; manual bandwidth control should always be used.

2.5.2 Synchronization of PLL with subsystems

If the MCS bit in PLLCR is set, then the SCI and timer clocks run off the PLL output (4XCLK) as does the CPU. If MCS is cleared, then the timer and SCI subsystems operate off the EXTALi frequency, but are accessed by the CPU relative to the internal PH2 signal. In this case, although EXTALi is used as the reference for the PLL, the PH2 clock and the module clocks for the timer and the SCI are not synchronized. In order to ensure synchronized data, special circuitry has been incorporated into both subsystems.

2.5.3 Changing the PLL frequency

The PLL output frequency can be changed by altering the contents of the SYNR register (see Section 2.5.4.2). To prevent possible bursts of high frequency operation during the reconfiguration of the PLL, the following sequence should be performed:

1) Switch to the low frequency bus rate (BCS = 0).
2) Disable the PLL (PLLON = 0).
3) Change the value in SYNR.
4) Enable the PLL (PLLON = 1).
5) Wait a time t_{PLLS} for the PLL frequency to stabilize.
6) Switch to the high frequency bus rate (BCS = 1).

2.5.4 PLL registers

Two registers are used to control the operation of the MC68HC11PH8 phase locked loop circuitry. These are the PLL control register and the synthesizer program register, each of which is described below.

2.5.4.1 PLLCR — PLL control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
PLL control (PLLCR)	$002E	PLLON	BCS	AUTO	BWC	VCOT	MCS	T16EN	WEN	x010 1010

This read/write register contains two bits that are used to enable and disable the synthesizer and to switch from slow (EXTALi) to one of the fast speeds. Two further bits are used to control the filter bandwidth. The SCI, timer, timer clock source and the slow clock for WAIT mode are also controlled by this register.

PLLON — PLL on

1 (set) – Switch PLL on.

0 (clear) – Switch PLL off.

This bit activates the synthesizer circuit without connecting it to the control circuit. This allows the circuit to stabilize before it drives the CPU clocks.

On reset, PLLON is forced low if the VDDSYN supply is low. If VDDSYN is at VDD, PLLON is set by reset to allow the control loop to stabilize during power-up. PLLON cannot be cleared whilst using VCOOUT to drive the internal processor clock, i.e. when BCS is set.

BCS — Bus clock select

1 (set) – VCOOUT output drives the clock circuit (4XCLK).

0 (clear) – EXTALi drives the clock circuit (4XCLK).

This bit determines which signal drives the clock circuit generating the bus clocks. Once BCS has been altered it can take up to [1.5 EXTALi + 1.5 VCOOUT] cycles for the change in the clock to occur. Reset clears this bit.

Note: PLLON and BCS have built-in safeguards so that VCOOUT cannot be selected as the clock source (BCS = 1) if the PLL is off (PLLON = 0). Similarly, the PLL cannot be turned off (PLLON = 0) if it is on and in use (BCS = 1). Turning the PLL on and selecting VCOOUT as the clock source therefore requires two independent writes to PLLCR.

AUTO — Automatic bandwidth control (Test mode only)

1 (set) – Automatic bandwidth control selected.

0 (clear) – Manual bandwidth control selected.

Reset sets this bit.

Warning: This bit must be cleared before an attempt is made to use BWC; manual bandwidth control should always be used.

BWC — Bandwidth control

1 (set) – Wide (high and low) bandwidth control selected.

0 (clear) – Narrow (low) bandwidth control selected.

Bandwidth selection can only be controlled by BWC when AUTO is cleared. After the PLL is first enabled, or after a change in frequency, a delay of t_{PLLS} is required before clearing BWC. The low bandwidth driver is always enabled, so this bit determines whether the high bandwidth driver is on or off. Reset clears this bit.

VCOT — VCO test (Test mode only)

1 (set) – Loop filter operates as specified by AUTO and BWC.

0 (clear) – Low bandwidth mode of the PLL filter is disabled.

This bit is used to isolate the loop filter from the VCO for testing purposes. VCOT is always set in user modes. This bit is writable only in bootstrap and test modes. Reset sets this bit.

MCS — Module clock select

1 (set) – 4XCLK is the source for the SCI and timer divider chain.

0 (clear) – EXTALi is the source for the SCI and timer divider chain.

Reset clears this bit.

T16EN — 16-bit timer clock enable (refer to Section 8)

1 (set) – 16-bit timer clock enabled.

0 (clear) – 16-bit timer clock disabled.

WEN — WAIT enable

1 (set) – Low-power WAIT mode selected (PLL set to ‘idle’ in WAIT mode).

0 (clear) – Do not alter the 4XCLK during WAIT mode.

This bit determines whether the 4XCLK is disconnected from VCOOUT during WAIT and connected to EXTALi. Reset clears this bit.

When WEN is set, the CPU will respond to a WAIT instruction by first stacking the relevant registers, then by clearing BCS and setting the PLL to ‘idle’, with modulus = 1. BWC is set so that the wide bandwidth control is selected.

Any interrupt, any reset, or the assertion of RAF (receiver active flag) in either of the SCIs will allow the PLL to resume operating at the frequency specified in the SYNR. The user must set BCS after the PLL has had time to adjust (t_{PLLS}). If, for a specific SCI, the RE bit (receiver enable bit) is clear, then RAF cannot become set, hence the PLL will not resume normal operation. For a description of RAF and RE, see Section 5.

2.5.4.2 SYNR — Synthesizer program register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Synthesizer program (SYNR)	$002F	SYNX1	SYNX0	SYNY5	SYNY4	SYNY3	SYNY2	SYNY1	SYNY0	0000 1011

The PLL frequency synthesizer multiplies the frequency of the input oscillator. The multiplication factor is software programmable via a loop divider, which consists of a six-bit modulo N counter, with a further two bit scaling factor.

The multiplication factor is given by $2(Y + 1)2^X$, where $0 \le X \le 3$ and $0 \le Y \le 63$.

Bits in SYNR can be read at any time but can only be written if PLLON = 0.

Note: Exceeding recommended operating frequencies can result in indeterminate MCU operation.

SYNX[1:0]

These bits program the binary taps (divide by 1, 2, 4 and 8). Reset clears these bits.

SYNY[5:0]

These bits program the six-bit modulo N (1 to 64) counter. Reset sets these bits to %001011.

Note: The resolution of the multiplication factors decreases by a factor of two, as X increases:

X	Y	Possible multipliers
0	0 – 63	2, 4, 6, 8, …, 128
1	0 – 63	4, 8, 12, 16, …, 256
2	0 – 63	8, 16, 24, 32, …, 512
3	0 – 63	16, 32, 48, 64, …, 1024

2.6 Interrupt request ($\overline{\text{IRQ}}$)

The $\overline{\text{IRQ}}$ input provides a means of applying asynchronous interrupt requests to the MCU. Either falling edge sensitive triggering or level sensitive triggering is program selectable (OPTION register). $\overline{\text{IRQ}}$ is always configured to level sensitive triggering at reset.

Note: Connect an external pull-up resistor, typically 4.7 kΩ, to V_{DD} when $\overline{\text{IRQ}}$ is used in a level sensitive wired-OR configuration. See also Section 2.7.

2.7 Nonmaskable interrupt ($\overline{\text{XIRQ}}$/*VPPE*)

The $\overline{\text{XIRQ}}$ input provides a means of requesting a nonmaskable interrupt after reset initialization. During reset, the X bit in the condition code register (CCR) is set and any interrupt is masked until MCU software enables it. Because the $\overline{\text{XIRQ}}$ input is level-sensitive, it can be connected to a multiple-source wired-OR network with an external pull-up resistor to V_{DD}. $\overline{\text{XIRQ}}$ is often used as a power loss detect interrupt.

Whenever $\overline{\text{XIRQ}}$ or $\overline{\text{IRQ}}$ is used with multiple interrupt sources ($\overline{\text{IRQ}}$ must be configured for level sensitive operation if there is more than one source of $\overline{\text{IRQ}}$ interrupt), each source must drive the interrupt input with an open-drain type of driver to avoid contention between outputs. There should be a single pull-up 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 source is 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 Section 10.

On the MC68HC711PH8, the VPPE pin is used to input the external EPROM programming voltage, which must be present during EPROM programming.

2.8 MODA and MODB (MODA/$\overline{\mathrm{LIR}}$ and MODB/VSTBY)

During reset, MODA and MODB select one of the four operating modes. Refer to Section 3.

After the operating mode has been selected, the $\overline{\mathrm{LIR}}$ pin provides an open-drain output (driven low) to indicate that execution of an instruction has begun. In order to detect consecutive instructions in a high-speed application, this signal drives high for a short time to prevent false triggering. A series of E clock cycles occurs during execution of each instruction. The $\overline{\mathrm{LIR}}$ signal goes low during the first E clock cycle of each instruction (opcode fetch). This output is provided for assistance in program debugging and its operation is controlled by the LIRDV bit in the OPT2 register.

The VSTBY pin is used to input RAM stand-by power. The MCU is powered from the VDD pin unless the difference between the level of VSTBY and VDD is greater than one MOS threshold (about 0.7 volts). When these voltages differ by more than 0.7 volts, the internal 1024-byte RAM and part of the reset logic are powered from VSTBY rather than VDD. This allows RAM contents to be retained without VDD 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.

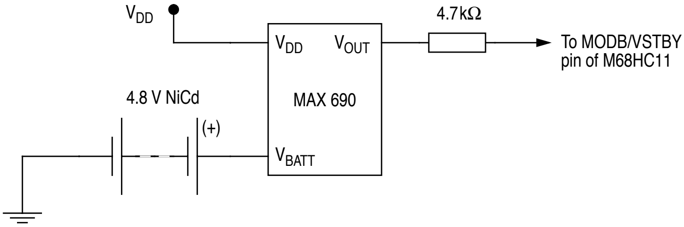


Figure 2-7 RAM stand-by connections

2.9 VRH and VRL

These pins provide the reference voltages for the analog-to-digital converter.

2.10 PG7/R/$\overline{\mathrm{W}}$

This pin provides two separate functions, depending on the operating mode. In single chip and bootstrap modes, PG7/R/$\overline{\mathrm{W}}$ acts as input/output port G bit 7. Refer to Section 4 for further information.

In expanded and test modes, PG7/R/$\overline{\mathrm{W}}$ performs the read/write function. PG7/R/$\overline{\mathrm{W}}$ signals the direction of transfers on the external data bus. A high on this pin indicates that a read cycle is in progress.

2.11 Port signals

62 pins on the device are arranged into seven 8-bit ports: A, B, C, E, F, G, and H, and one six-bit port (D). The lines of ports A, B, C, D, F, G, and H are fully bidirectional; E is input only. Each of the bidirectional ports serves a purpose other than I/O, depending on the operating mode or peripheral function selected. Note that ports B, C, F, and one bit of port G are available for I/O functions only in single chip and bootstrap modes. Refer to Table 2-2 for details of the port signals' functions in different operating modes.

Note: When using the information about port functions, do not confuse pin function with the electrical state of the pin at reset. All general purpose I/O pins configured as inputs at reset are in a high-impedance state. Port data registers reflect the functional state of the port at reset. The pin function is mode dependent.

2.11.1 Port A

Port A is an 8-bit general purpose I/O port with a data register (PORTA) and a data direction register (DDRA). Port A pins share functions with the 16-bit timer system (see Section 8 for further information). PORTA can be read at any time and always returns the pin level. If written, PORTA stores the data in internal latches. The pins are driven only if they are configured as outputs. Writes to PORTA do not change the pin state when the pins are configured for timer output compares.

Out of reset, port A pins [7:0] are general purpose high-impedance inputs. When the functions associated with these pins are disabled, the bits in DDRA govern the I/O state of the associated pin. For further information, refer to Section 4.

2.11.2 Port B

Port B is an 8-bit general purpose I/O port with a data register (PORTB) and a data direction register (DDRB). In single chip mode, port B pins are general purpose I/O pins (PB[7:0]). In expanded mode, port B pins act as the high-order address lines (A[15:8]) of the address bus. In either of these modes, the four high-order port B pins (B[7:4]) may be configured to drive four LCD segments (see Section 2.12)

PORTB can be read at any time and always returns the pin level. If PORTB is written, the data is stored in internal latches. The pins are driven only if they are configured as outputs in single chip or bootstrap mode. For further information, refer to Section 4.

Port B pins include on-chip pull-up devices which can be enabled or disabled via the port pull-up assignment register (PPAR).

Table 2-2 Port signal functions

Port/bit	Single chip and bootstrap mode	Expanded multiplexed and special test mode
PA0	PA0/IC3	
PA1	PA1/IC2	
PA2	PA2/IC1	
PA3	PA3/OC5/IC4 and/or OC1	
PA4	PA4/OC4 and/or OC1	
PA5	PA5/OC3 and/or OC1	
PA6	PA6/OC2 and/or OC1	
PA7	PA7/PAI and/or OC1	
PB[3:0]	PB[3:0]	A[11:8]
PB4	PB4/LCD4	A12/LCD4
PB5	PB5/LCD5	A13/LCD5
PB6	PB6/LCD6	A14/LCD6
PB7	PB7/LCD7	A15/LCD7
PC[7:0]	PC[7:0]	D[7:0]
PD0	PD0/RXD1	
PD1	PD1/TXD1	
PD2	PD2/MISO1	
PD3	PD3/MOSI1	
PD4	PD4/SCK1	
PD5	PD5/$\overline{\text{SS}}$1	
PE[7:0]	Input only or analog inputs	
PF[7:0]	PF[7:0]	A[7:0]
PG0	PG0/RXD2	
PG1	PG1/TXD2	
PG2	PG2/MISO2	
PG3	PG3/MOSI2	
PG4	PG4/SCK2	
PG5	PG5/$\overline{\text{SS}}$2	
PG6	PG6/LCDBP	
PG7	PG7	R/$\overline{\text{W}}$
PH0	PH0/PW1	
PH1	PH1/PW2	
PH2	PH2/PW3	
PH3	PH3/PW4	
PH4	PH4	
PH5	PH5	
PH6	PH6/Modulus timer C clock input	
PH7	PH7/Modulus timer B clock input	

2.11.3 Port C

Port C is an 8-bit general purpose I/O port with a data register (PORTC) and a data direction register (DDRC). In single chip mode, port C pins are general purpose I/O pins (PC[7:0]). In the expanded mode, port C pins are configured as data bus pins (D[7:0]).

PORTC can be read at any time and always returns the pin level. If PORTC is written, the data is stored in internal latches. The pins are driven only if they are configured as outputs in single chip or bootstrap mode. Port C pins are general purpose inputs out of reset in single chip and bootstrap modes. In expanded and test modes, these pins are data bus lines out of reset.

The CWOM control bit in the OPT2 register disables port C's p-channel output drivers. Because the n-channel driver is not affected by CWOM, setting CWOM causes port C to become an open-drain-type output port suitable for wired-OR operation. In wired-OR mode (PORTC bits at logic level zero), the pins are actively driven low by the n-channel driver. When a port C bit is at logic level one, the associated pin is in a high impedance state as neither the n-channel nor the p-channel devices are active. It is customary to have an external pull-up resistor on lines that are driven by open-drain devices. Port C can only be configured for wired-OR operation when the MCU is in single chip mode. For further information, refer to Section 4.

2.11.4 Port D

Port D, a 6-bit general purpose I/O port, has a data register (PORTD) and a data direction register (DDRD). The six port D lines (D[5:0]) can be used for general purpose I/O, for one of the serial communications interfaces (SCI1, pins [1,0]) and for one of the serial peripheral interfaces (SPI1, pins [5:2]).

PORTD can be read at any time; inputs return the pin level and outputs return the pin driver input level. If PORTD is written, the data is stored in internal latches. The pins are driven only if port D is configured for general purpose output.

The DWOM bit in SPCR disables the p-channel output drivers of pins D[5:2], and the WOMS bit in SCCR1 disables those of pins D[1,0]. Because the n-channel driver is not affected by DWOM or WOMS, setting either bit causes the corresponding port D pins to become open-drain-type outputs suitable for wired-OR operation. In wired-OR mode (PORTD bits at logic level zero), the pins are actively driven low by the n-channel driver. When a port D bit is at logic level one, the associated pin is in a high impedance state as neither the n-channel nor the p-channel devices are active. It is customary to have an external pull-up resistor on lines that are driven by open-drain devices. Port D can be configured for wired-OR operation when the MCU is in single chip mode or expanded mode.

For further information, refer to Section 4, Section 5 (SCI) and Section 7 (SPI).

2.11.5 Port E

Port E, PE/AD[7:0], is an input-only port that can also be used as the analog inputs for the analog-to-digital converter.

For further information, refer to Section 4 and Section 9 (A/D).

2.11.6 Port F

Port F is an 8-bit general purpose I/O port with a data register (PORTF) and a data direction register (DDRF). In single chip mode, port F pins are general purpose I/O pins (PF[7:0]). In expanded mode, port F pins act as the low-order address lines (A[7:0]) of the address bus.

PORTF can be read at any time and always returns the pin level. If PORTF is written, the data is stored in internal latches. The pins are driven only if they are configured as outputs in single chip or bootstrap mode.

Port F pins include on-chip pull-up devices that can be enabled or disabled via the port pull-up assignment register (PPAR). For further information, refer to Section 4.

2.11.7 Port G

In normal modes, Port G is an 8-bit general purpose I/O port with a data register (PORTG) and a data direction register (DDRG). Port G pin 7 is the R/$\overline{\text{W}}$ line in expanded mode; pin 6 can be used for the LCD backplane signal (LCDBP) in any mode; the remaining pins can be used for general purpose I/O, for one of the SCI subsystems (SCI2 with MI-bus, pins [1,0]), or for one of the serial peripheral interface subsystems (SPI2, pins [5:2]).

PORTG can be read at any time; inputs return the pin level and outputs return the pin driver input level. If PORTG is written, the data is stored in internal latches. The pins are driven only if they are configured as outputs (and only in single chip or bootstrap mode for pins G[7,6]).

The GWOM bit in SP2CR disables the p-channel output drivers of pins G[5:2], and the WOMS2 bit in S2CR1 disables those of pins G[1,0]. Because the n-channel driver is not affected by GWOM or WOMS2, setting either bit causes the corresponding port G pins to become open-drain-type outputs suitable for wired-OR operation. In wired-OR mode (appropriate PORTG bits at logic level zero), the pins are actively driven low by the n-channel driver. When a port G bit is at logic level one, the associated pin is in a high impedance state as neither the n-channel nor the p-channel devices are active. It is customary to have an external pull-up resistor on lines that are driven by open-drain devices. Port G pins [5:0] can be configured for wired-OR operation when the MCU is in single chip mode or expanded mode.

Port G pins include on-chip pull-up devices that can be enabled or disabled via the port pull-up assignment register (PPAR). For further information, refer to Section 4, Section 5 (SCI), Section 6 (MI BUS) and Section 7 (SPI).

2.11.8 Port H

Port H is an 8-bit general purpose I/O port with a data register (PORTH) and a data direction register (DDRH). Port H pins support either input/output, pulse-width modulation channels (pins [3:0]) or act as clock inputs for two of the 8-bit modulus timers (pins [7,6]).

PORTH can be read at any time and always returns the pin level.

Port H pins include on-chip pull-up devices that can be enabled or disabled via the port pull-up assignment register (PPAR). Port H pins can be configured for wired-OR interrupt to wake-up from WAIT or STOP mode under control of the wired-OR interrupt register (WOIEH).

For further information, refer to Section 4 and Section 8 (timer system).

2.12 LCD module

The MC68HC11PH8 incorporates an LCD module that allows four LCD segments to be driven under control of the LCD control and data register. The four frontplane signals are output on port B pins [7:4], with the backplane signal output on PG6. A segment is ON when the corresponding frontplane and backplane are equal in frequency and opposite in phase. The LCD function can be enabled in any mode.

2.12.1 LCDR — LCD control and data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
LCD control and data (LCDR)	$002D	LCD7	LCD6	LCD5	LCD4	0	0	LCDCK	LCDE	0000 0000

LCD[7:4] — LCD segment data

1 (set) – Segment ON (corresponding LCD output port B is opposite in phase to LCDBP).

0 (clear) – Segment OFF (corresponding LCD output port B is in phase with LCDBP).

When LCD[7:4] are all cleared, the LCD backplane is forced low[†] and all the LCD segments are off, thus reducing power consumption and RFI emissions.

[†] *This is not the case on early versions of the MC68HC711PH8; contact your local Motorola Sales Representative for more information.*

LCDCK — LCD frequency clock select

1 (set) – The clock source of the real time interrupt (RTI) toggles LCDBP.

0 (clear) – 8-bit modulus timer A underflow (CLK64) toggles LCDBP.

When the PLL clock generation circuit is not used (VDDSYN = 0), setting LCDCK selects the ST4XCK clock divided by 2^{18} as the LCD clock source. Conversely, when the PLL clock generation circuit is used (VDDSYN = 1), setting LCDCK selects the output of the 8-bit modulus timer A divided by 2^3 (CLK64/2^3) as the LCD clock source. Refer to Section 8.

LCDE — LCD function enable

1 (set) – LCD function enabled.

0 (clear) – LCD function disabled.

The LCDE bit can be written only once (the first write to this register after reset will prevent later updates of this bit). When enabled, this function will force PG6 into output mode. This output will be the backplane signal (LCDBP) for the four LCD segments. The four port B pins (PB[7:4]) used to drive the LCD segments will also be forced into output mode. If enabled in expanded modes, PB[7:4] will operate as LCD outputs, while PB3 to PB0 output the address lines to access external resources. To avoid conflicts caused by the LCDE bit being set accidentally by program error, it is recommended that the LCDE bit be written to zero if the LCD function is not required.

THIS PAGE INTENTIONALLY LEFT BLANK

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

3
OPERATING MODES AND ON-CHIP MEMORY

This section contains information about the modes that define MC68HC11PH8 operating conditions, and about the on-chip memory that allows the MCU to be configured for various applications.

3.1 Operating modes

The values of the mode select inputs MODB and MODA during reset determine the operating mode (See Table 3-4). Single chip and expanded modes are the normal modes. In single chip mode only on-board memory is available. Expanded mode, however, allows access to external memory. Each of these 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.

3.1.1 Single chip operating mode

In single chip operating mode, the MC68HC11PH8 microcontroller has no external address or data bus. Ports B, C, F, and the R/$\overline{\text{W}}$ pin are available for general purpose parallel I/O.

3.1.2 Expanded operating mode

In expanded operating mode, the MCU can access a 64K byte physical address space. The address space includes the same on-chip memory addresses used for single chip mode, in addition to external memory and peripheral devices.

The expansion bus is made up of ports B, C, and F, and the R/$\overline{\text{W}}$ signal. In expanded mode, high order address bits are output on the port B pins, low order address bits on the port F pins, and the data bus on port C. The R/$\overline{\text{W}}$/PG7 pin signals the direction of data transfer on the port C bus.

When internal resources are accessed in expanded mode, the last external address can be held on the output pins (A[15:0]) in order to reduce radio-frequency interference (RFI) emissions. This function is controlled by the FREEZ bit in the CONFIG register. See Section 3.3.2.1 for a description of this bit.

To allow access to slow peripherals, off chip accesses can be extended by one E clock cycle, under control of the STRCH and STRX bits (in the OPT2 and INIT2 registers respectively). The E clock

TPG

stretches externally, but the internal clocks are not affected so that timers and serial systems are not corrupted. See Section 3.3.2.5.

EEPROM data can be protected while in expanded mode, using a security feature described in Section 3.4.4.

3.1.3 Special test -mode

Special test, a variation of the expanded mode, is primarily used during Motorola's internal production testing; however, it is accessible for programming the CONFIG register, programming calibration data into EEPROM, and supporting emulation and debugging during development.

3.1.4 Special bootstrap mode

When the MCU is reset in special bootstrap mode, a small on-chip ROM is enabled at address $BE40–$BFFF. The ROM contains a reset vector and a bootloader program. The MCU fetches the reset vector, then executes the bootloader.

For normal use of the bootloader program, send a synchronization byte $FF to the SCI receiver at either E clock ÷256, or E clock ÷1664 (7812 or 1200 baud respectively, for an E clock of 2 MHz). Then download up to 2048 bytes of program data (which is put into RAM starting at $0080). These characters are echoed through the transmitter. The bootloader program ends the download after a timeout of four character times or 2048 bytes. When loading is complete, the program jumps to location $0080 and begins executing the code. Use of an external pull-up resistor is required when using the SCI transmitter pin (TXD) because port D pins are configured for wired-OR operation by the bootloader. In bootstrap mode, the interrupt vectors point to RAM. This allows the use of interrupts through a jump table.

Further baud rate options are available on the MC68HC11PH8 by using a different value for the synchronization byte, as shown in the Table 3-1.

A special mode exists that allows a low frequency crystal to be used if the PLL is active. In this case, the value on port F is loaded into the SYNR register just after reset, to be used as the multiplication factor for the crystal frequency. If the PLL is not active, then the bootloader runs at the crystal frequency. Refer to Section 2.5 for more information on the operation of the PLL circuitry.

Refer also to Motorola application note ***AN1060, M68HC11 Bootstrap Mode*** (the bootloader mode is similar to that used on the MC68HC11K4).

TPG

Table 3-1 Example bootloader baud rates

Sync. byte	Timeout delay	Baud rates for an E clock of:				
		2.00 MHz	2.10 MHz	3.00 MHz	3.15 MHz	4.00 MHz
$FF	4 char.	7812	8192	11718	12288	15624
$FF	4	1200	1260	1800	1890	2400
$F0	4.9	9600	10080	14400	15120	19200
$FD	17.3	5208	5461	7812	8192	10416
$FD	13	3906	4096	5859	6144	7812

3.2 On-chip memory

The MC68HC11PH8 MCU includes 2K bytes of on-chip RAM, 48K bytes of ROM/*EPROM* and 768 bytes of EEPROM. The bootloader ROM occupies a 512 byte block of the memory map. The CONFIG register is implemented as a separate EEPROM byte.

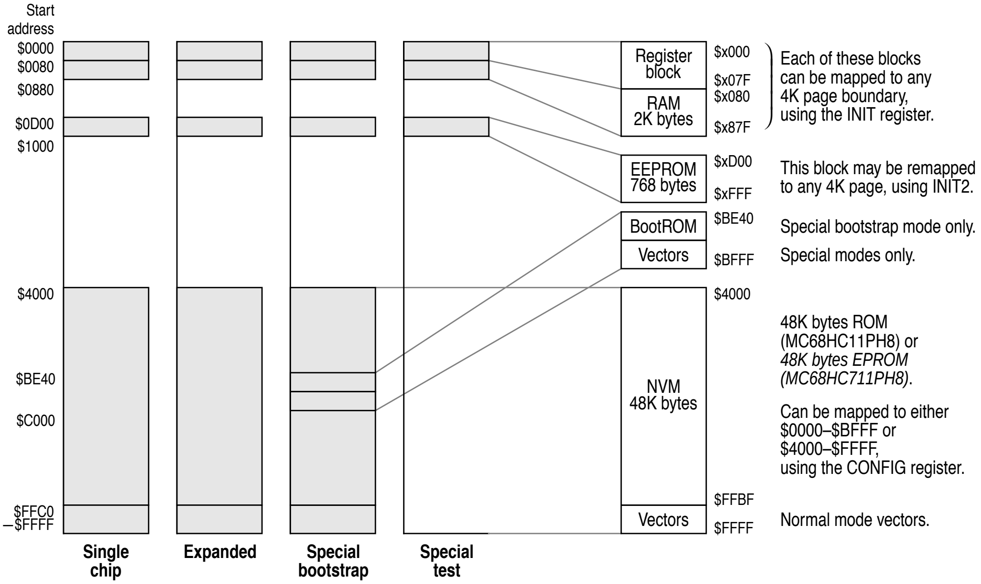

Start address
$0000
$0080
$0880
$0D00
$1000
$4000
$BE40
$C000
$FFC0 –$FFFF
Single chip
Expanded
Special bootstrap
Special test
Register block
RAM 2K bytes
EEPROM 768 bytes
BootROM
Vectors
NVM 48K bytes
Vectors
$x000
$x07F
$x080
$x87F
$xD00
$xFFF
$BE40
$BFFF
$4000
$FFBF
$FFFF
Each of these blocks can be mapped to any 4K page boundary, using the INIT register.
This block may be remapped to any 4K page, using INIT2.
Special bootstrap mode only.
Special modes only.
48K bytes ROM (MC68HC11PH8) or *48K bytes EPROM (MC68HC711PH8)*.
Can be mapped to either $0000–$BFFF or $4000–$FFFF, using the CONFIG register.
Normal mode vectors.

Figure 3-1 MC68HC11PH8/*MC68HC711PH8* memory map

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
TPG

3.2.1 Mapping allocations

Memory locations for on-chip resources are the same for both expanded and single chip modes. The 128-byte register block originates at $0000 after reset and can be placed at any other 4K boundary ($x000) after reset by writing an appropriate value to the INIT register. Refer to Figure 3-1, which shows the memory map.

The on-board 2K byte RAM is initially located at $0080 after reset. The RAM is divided into two sections, of 128 bytes and 1920 bytes. If RAM and registers are both mapped to the same 4K boundary, RAM starts at $x080 and 128 bytes are remapped at $x800–$x87F. Otherwise, RAM starts at $x000. See Figure 3-3.

Remapping is accomplished by writing appropriate values into the two nibbles of the INIT register. See Section 3.3.2.2.

The 768-byte EEPROM is initially located at $0D00 after reset, when EEPROM is enabled in the memory map by the CONFIG register. EEPROM can be placed in any other 4K page ($xD00) by writing to the INIT2 register.

The ROMAD and ROMON bits in the CONFIG register control the position and presence of ROM, *or EPROM*, in the memory map. In special test mode, the ROMON bit is cleared so the ROM is removed from the memory map. In single chip mode, the ROMAD bit is set to one after reset, which enables the ROM at $4000–$FFFF. In expanded mode, the ROM may be enabled from $0000–$BFFF (ROMAD = 0) to allow an external memory to contain the interrupt vectors and initialization code.

In special bootstrap mode, a bootloader ROM is enabled at locations $BE40–$BFFF. The vectors for special bootstrap mode are contained in the bootloader program. The boot ROM occupies a 512 byte block of the memory map, though not all locations are used.

3.2.1.1 RAM

The MC68HC11PH8 has 2K bytes of fully static RAM that are used for storing instructions, variables and temporary data during program execution. RAM can be placed at any 4K boundary in the 64K byte address space by writing an appropriate value to the INIT register.

By default, RAM is initially located at $0080 in the memory map. Direct addressing mode can access the first 128 locations of RAM using a one-byte address operand. Direct mode accesses save program memory space and execution time. Registers can be moved to other boundaries to allow 256 bytes of RAM to be located in direct addressing space. See Figure 3-3.

The on-chip RAM is a fully static memory. RAM contents can be preserved during periods of processor inactivity by either of two methods, both of which reduce power consumption:

TPG

1) During the software-based STOP mode, MCU clocks are stopped, but the MCU continues to draw power from V_{DD}. Power supply current is directly related to operating frequency in CMOS integrated circuits and there is very little leakage when the clocks are stopped. These two factors reduce power consumption while the MCU is in STOP mode.
2) To reduce power consumption to a minimum, V_{DD} can be turned off, and the MODB/VSTBY pin can be used to supply RAM power from either a battery back-up or a second power supply. Although this method requires external hardware, it is very effective. Refer to Section 2 for information about how to connect the stand-by RAM power supply and to Section 10 for a description of low power operation.

3.2.1.2 ROM and EPROM

The MCU has 48K bytes of ROM/*EPROM*. The ROM/*EPROM* array is enabled when the ROMON bit in the CONFIG register is set to one (erased). The ROMAD bit in CONFIG places the ROM/*EPROM* at either $4000–$FFFF (ROMAD = 1) or at $0000–$BFFF (ROMAD = 0) when coming out of reset in expanded mode.

3.2.1.3 Bootloader ROM

The bootloader ROM is enabled at address $BE40–$BFFF during special bootstrap mode. The reset vector is fetched from this ROM and the MCU executes the bootloader firmware. In normal modes, the bootloader ROM is disabled.

3.2.2 Registers

In Table 3-2, a summary of registers and control bits, the registers are shown in ascending order within the 128-byte register block. The addresses shown are for default block mapping ($0000–$007F), however, the INIT register remaps the block to any 4K page ($x000–$x07F). See Section 3.3.2.2.

Table 3-2 Register and control bit assignments (Sheet 1 of 4)

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port A data (PORTA)	$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	undefined
Data direction A (DDRA)	$0001	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	0000 0000
Data direction B (DDRB)	$0002	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	0000 0000
Data direction F (DDRF)	$0003	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	0000 0000
Port B data (PORTB)	$0004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	undefined
Port F data (PORTF)	$0005	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	undefined
Port C data (PORTC)	$0006	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	undefined
Data direction C (DDRC)	$0007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	0000 0000
Port D data (PORTD)	$0008	0	0	PD5	PD4	PD3	PD2	PD1	PD0	undefined
Data direction D (DDRD)	$0009	0	0	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	0000 0000
Port E data (PORTE)	$000A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	undefined
Timer compare force (CFORC)	$000B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	0000 0000
Output compare 1 mask (OC1M)	$000C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	0000 0000
Output compare 1 data (OC1D)	$000D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	0000 0000
Timer count (TCNT) high	$000E	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	0000 0000
Timer count (TCNT) low	$000F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Timer input capture 1 (TIC1) high	$0010	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
Timer input capture 1 (TIC1) low	$0011	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
Timer input capture 2 (TIC2) high	$0012	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
Timer input capture 2 (TIC2) low	$0013	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
Timer input capture 3 (TIC3) high	$0014	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
Timer input capture 3 (TIC3) low	$0015	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
Timer output compare 1 (TOC1) high	$0016	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 1 (TOC1) low	$0017	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 2 (TOC2) high	$0018	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 2 (TOC2) low	$0019	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 3 (TOC3) high	$001A	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 3 (TOC3) low	$001B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 4 (TOC4) high	$001C	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 4 (TOC4) low	$001D	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Capture 4/compare 5 (TI4/O5) high	$001E	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Capture 4/compare 5 (TI4/O5) low	$001F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer control 1 (TCTL1)	$0020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	0000 0000
Timer control 2 (TCTL2)	$0021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	0000 0000
Timer interrupt mask 1 (TMSK1)	$0022	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I	0000 0000

TPG

Table 3-2 Register and control bit assignments (Sheet 2 of 4)

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt flag 1 (TFLG1)	$0023	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F	0000 0000
Timer interrupt mask 2 (TMSK2)	$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	0000 0000
Timer interrupt flag 2 (TFLG2)	$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	0000 0000
Pulse accumulator control (PACTL)	$0026	0	PAEN	PAMOD	PEDGE	0	I4/O5	RTR1	RTR0	0000 0000
Pulse accumulator count (PACNT)	$0027	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
SPI control (SPCR)	$0028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	0000 01uu
SPI status (SPSR)	$0029	SPIF	WCOL	0	MODF	0	0	0	0	0000 0000
SPI data (SPDR)	$002A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
EPROM programming (EPROG) ‡	$002B	MBE	0	ELAT	EXCOL	EXROW	0	0	EPGM	0000 0000
Port pull-up assignment (PPAR)	$002C	0	0	0	HWOIF	HPPUE	GPPUE	FPPUE	BPPUE	0000 1111
LCD control and data (LCDR)	$002D	LCD7	LCD6	LCD5	LCD4	0	0	LCDCK	LCDE	0000 0000
PLL control (PLLCR)	$002E	PLLON	BCS	AUTO	BWC	VCOT	MCS	T16EN	WEN	x010 1010
Synthesizer program (SYNR)	$002F	SYNX1	SYNX0	SYNY5	SYNY4	SYNY3	SYNY2	SYNY1	SYNY0	0000 1011
A/D control & status (ADCTL)	$0030	CCF	0	SCAN	MULT	CD	CC	CB	CA	u0uu uuuu
A/D result 1 (ADR1)	$0031	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
A/D result 2 (ADR2)	$0032	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
A/D result 3 (ADR3)	$0033	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
A/D result 4 (ADR4)	$0034	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
Block protect (BPROT)	$0035	BULKP	0	BPRT4	PTCON	BPRT3	BPRT2	BPRT1	BPRT0	1011 1111
reserved	$0036									
EEPROM mapping (INIT2)	$0037	EE3	EE2	EE1	EE0	STRX	0	M2DL1	M2DL0	0000 0000
System config. options 2 (OPT2)	$0038	LIRDV	CWOM	STRCH	IRVNE	LSBF	SPR2	EXT4X	DISE	x00x 0000
System config. options 1 (OPTION)	$0039	ADPU	CSEL	IRQE	DLY	CME	FCME	CR1	CR0	0001 0000
COP timer arm/reset (COPRST)	$003A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
EEPROM programming (PPROG)	$003B	ODD	EVEN	0	BYTE	ROW	ERASE	EELAT	EEPGM	0000 0000
Highest priority interrupt (HPRIO)	$003C	RBOOT	SMOD	MDA	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0	xxx0 0110
RAM & I/O mapping (INIT)	$003D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	0000 0000
Factory test (TEST1)	$003E	TILOP	PLTST	OCCR	CBYP	DISR	FCM	FCOP	MIDLY	0000 0000
Configuration control (CONFIG)	$003F	ROMAD	FREEZ	CLK4X	PAREN	NOSEC	NOCOP	ROMON	EEON	xxxx xxxx
reserved	$0040									
reserved	$0041									
reserved	$0042									
reserved	$0043									
reserved	$0044									
reserved	$0045									

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Table 3-2 Register and control bit assignments (Sheet 3 of 4)

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
reserved	$0046									
reserved	$0047									
reserved	$0048									
reserved	$0049									
reserved	$004A									
reserved	$004B									
SPI2 control (SP2CR)	$004C	SP2IE	SP2E	GWOM	MSTR2	CPOL2	CPHA2	SP2R1	SP2R0	0000 01uu
SPI2 status (SP2SR)	$004D	SP2IF	WCOL2	0	MODF2	0	0	0	0	0000 0000
SPI2 data (SP2DR)	$004E	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
SPI2 control options (SP2OPT)	$004F	0	0	0	0	LSBF2	SP2R2	0	0	0000 0000
SCI2/MI baud high (S2BDH)	$0050	B2TST	B2SPL	B2RST	S2B12	S2B11	S2B10	S2B9	S2B8	0000 0000
SCI2/MI baud low (S2BDL)	$0051	S2B7	S2B6	S2B5	S2B4	S2B3	S2B2	S2B1	S2B0	0000 0100
SCI2/MI control 1 (S2CR1)	$0052	LOPS2	WOMS2	MIE2	M2	WAKE2	ILT2	PE2	PT2	0000 0000
SCI2/MI control 2 (S2CR2)	$0053	TIE2	TCIE2	RIE2	ILIE2	TE2	RE2	RWU2	SBK2	0000 0000
SCI2/MI status 1 (S2SR1)	$0054	TDRE2	TC2	RDRF2	IDLE2	OR2	NF2	FE2	PF2	1100 0000
SCI2/MI status 2 (S2SR2)	$0055	0	0	0	0	0	0	0	RAF2	0000 0000
SCI2/MI data high (S2DRH)	$0056	R8B	T8B	0	0	0	0	0	0	undefined
SCI2/MI data low (S2DRL)	$0057	R7T7B	R6T6B	R5T5B	R4T4B	R3T3B	R2T2B	R1T1B	R0T0B	undefined
reserved	$0058									
8-bit modulus timer A data (T8ADR)	$0059	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
8-bit modulus timer B data (T8BDR)	$005A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
8-bit modulus timer C data (T8CDR)	$005B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
reserved	$005C									
8-bit modulus timer A control (T8ACR)	$005D	T8AI	T8AF	0	0	0	CSA2	CSA1	CSA0	0000 0000
8-bit modulus timer B control (T8BCR)	$005E	T8BI	T8BF	0	0	PRB	CSB2	CSB1	CSB0	0000 0000
8-bit modulus timer C control (T8CCR)	$005F	T8CI	T8CF	0	0	PRC	CSC2	CSC1	CSC0	0000 0000
Pulse width clock select (PWCLK)	$0060	CON34	CON12	PCKA2	PCKA1	0	PCKB3	PCKB2	PCKB1	0000 0000
Pulse width polarity select (PWPOL)	$0061	PCLK4	PCLK3	PCLK2	PCLK1	PPOL4	PPOL3	PPOL2	PPOL1	0000 0000
Pulse width scale (PWSCAL)	$0062	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Pulse width enable (PWEN)	$0063	TPWSL	DISCP	0	0	PWEN4	PWEN3	PWEN2	PWEN1	0000 0000
Pulse width count 1 (PWCNT1)	$0064	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Pulse width count 2 (PWCNT2)	$0065	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Pulse width count 3 (PWCNT3)	$0066	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Pulse width count 4 (PWCNT4)	$0067	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Pulse width period 1 (PWPER1)	$0068	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111

TPG

Table 3-2 Register and control bit assignments (Sheet 4 of 4)

Register name	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width period 2 (PWPER2)	$0069	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width period 3 (PWPER3)	$006A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width period 4 (PWPER4)	$006B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width duty 1 (PWDTY1)	$006C	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width duty 2 (PWDTY2)	$006D	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width duty 3 (PWDTY3)	$006E	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width duty 4 (PWDTY4)	$006F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
SCI1 baud rate high (SCBDH)	$0070	BTST	BSPL	BRST	SBR12	SBR11	SBR10	SBR9	SBR8	0000 0000
SCI1 baud rate low (SCBDL)	$0071	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	0000 0100
SCI1 control 1 (SCCR1)	$0072	LOOPS	WOMS	0	M	WAKE	ILT	PE	PT	0000 0000
SCI1 control 2 (SCCR2)	$0073	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	0000 0000
SCI1 status 1 (SCSR1)	$0074	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF	1100 0000
SCI1 status 2 (SCSR2)	$0075	0	0	0	0	0	0	0	RAF	0000 0000
SCI1 data high (SCDRH)	$0076	R8	T8	0	0	0	0	0	0	undefined
SCI1 data low (SCDRL)	$0077	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0	undefined
reserved	$0078									
reserved	$0079									
reserved	$007A									
Wired-OR interrupt enable (WOIEH)	$007B	IEH7	IEH6	IEH5	IEH4	IEH3	IEH2	IEH1	IEH0	0000 0000
Port H data (PORTH)	$007C	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	undefined
Data direction H (DDRH)	$007D	DDH7	DDH6	DDH5	DDH4	DDH3	DDH2	DDH1	DDH0	0000 0000
Port G data (PORTG)	$007E	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	undefined
Data direction G (DDRG)	$007F	DDG7	DDG6	DDG5	DDG4	DDG3	DDG2	DDG1	DDG0	0000 0000

KEY

‡ Applies only to EPROM devices

x State on reset depends on mode selected

u State of bit on reset is undefined

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

3.3 System initialization

Registers and bits that control initialization and the basic operation of the MCU are protected against writes except under special circumstances. The following table lists registers that can be written only once after reset, or that must be written within the first 64 cycles after reset.

Table 3-3 Registers with limited write access

Register address	Register name	Must be written in first 64 cycles	Write once only
$x024	Timer interrupt mask register 2 (TMSK2)	[1]	–
$x02D	LCD control and data register (LCDR)	No	[2]
$x035	Block protect register (BPROT)	[3]	–
$x037	EEPROM mapping register (INIT2)	No	Yes
$x038	System configuration options register 2 (OPT2)	No	[4]
$x039	System configuration options register (OPTION)	[5]	–
$x03D	RAM and I/O map register (INIT)	[6]	–

[1] When SMOD = 0, bits 1 and 0 can be written only once, during the first 64 cycles, after which they become read-only. When SMOD = 1, however, these bits can be written at any time. All other bits can be written at any time.

[2] Bit 0 (LCDE) can be written only once.

[3] Bits can be written to zero once and only in the first 64 cycles or in special modes. Bits can be set to one at any time.

[4] Bit 0 (DISE) and bit 1 (EXT4X) can be written only once; bit 4 (IRVNE) can be written only once in single chip and user expanded modes.

[5] Bits 5, 4, 2, 1, and 0 can be written once and only in the first 64 cycles; when SMOD = 1, however, bits 5, 4, 2, 1, and 0 can be written at any time. All other bits can be written at any time.

[6] When SMOD = 0, bits can be written only once, during the first 64 cycles, after which the register becomes read-only. When SMOD = 1, bits can be written at any time.

3.3.1 Mode selection

The four mode variations are selected by the logic states of the mode A (MODA) and mode B (MODB) pins during reset. The MODA and MODB logic levels determine the logic state of special mode (SMOD) and the mode A (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, MODA pin is connected to a logic zero. In expanded mode, MODA is normally connected to V_{DD} through a pull-up resistor of 4.7 kΩ. The MODA pin also functions as the load instruction register ($\overline{\text{LIR}}$) pin when the MCU is not in reset. The open-drain active low $\overline{\text{LIR}}$ output pin drives low during the first E cycle of each instruction, if enabled by the LIRDV bit in the OPT2 register. The MODB pin also functions as the stand-by power input (VSTBY), which allows the RAM contents to be maintained in the absence of V_{DD}.

Refer to Table 3-4, which is a summary of mode pin operation, the mode control bits and the four operating modes.

A normal mode is selected when MODB is logic one 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 zero during reset, the special mode reset vector is fetched from addresses $BFFA–$BFFF and software has access to special test features. Refer to Section 10.

3.3.1.1 HPRIO — Highest priority I-bit interrupt & misc. register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Highest priority interrupt (HPRIO)	$003C	RBOOT	SMOD	MDA	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0	xxx0 0110

Note: RBOOT, SMOD and MDA bits depend on the power-up initialization mode and can only be written in special modes when SMOD = 1. Refer to Table 3-4.

RBOOT — Read bootstrap ROM

1 (set) – Bootloader ROM enabled, at $BE40–$BFFF.

0 (clear) – Bootloader ROM disabled and not in map.

SMOD — Special mode select

1 (set) – Special mode variation in effect.

0 (clear) – Normal mode variation in effect.

Once cleared, cannot be set again.

MDA — Mode select A

1 (set) – Normal expanded or special test mode. (Expanded buses active.)

0 (clear) – Normal single chip or special bootstrap mode. (Ports active.)

Table 3-4 Hardware mode select summary

Inputs		Mode	Control bits in HPRIO (latched at reset)		
MODB	MODA		RBOOT	SMOD	MDA
1	0	Single chip	0	0	0
1	1	Expanded	0	0	1
0	0	Special bootstrap	1	1	0
0	1	Special test	0	1	1

PSEL[4:0] — Priority select bits (refer to Section 10)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

3.3.2 Initialization

Because bits in the following registers control the basic configuration of the MCU, an accidental change of their values could cause serious system problems. The protection mechanism, overridden in special operating modes, requires a write to the protected bits only within the first 64 bus cycles after any reset, or only once after each reset. See Table 3-3.

3.3.2.1 CONFIG — System configuration register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Configuration control (CONFIG)	$003F	ROMAD	FREEZ	CLK4X	PAREN	NOSEC	NOCOP	ROMON	EEON	xxxx xxxx

CONFIG controls the presence and/or location of ROM/*EPROM* and EEPROM in the memory map and enables the COP watchdog system.The FREEZ bit provides a method of reducing RFI emissions in expanded mode, the CLK4X bit enables the 4XOUT pin to output either 4XCLK or the internal EXTAL signal (EXTALi), and the PAREN bit enables pull-ups on certain ports. A security feature that protects data in EEPROM and RAM is available, controlled by the NOSEC bit. Refer to Section 3.4.4.

CONFIG is made up of EEPROM cells and static working latches. The operation of the MCU is controlled directly by these latches and not the EEPROM byte. When programming the CONFIG register, the EEPROM byte is accessed. When the CONFIG register is read, the static latches are accessed.

These bits can be read at any time. The value read is the one latched into the register from the EEPROM cells during the last reset sequence. A new value programmed into this register is not readable until after a subsequent reset sequence.

On the MC68HC711PH8, and on the MC68HC11PH8 if selected by a mask option, the ROMON bit can be written at any time if MDA = 1 (expanded mode or special test mode). It cannot be written in bootstrap mode, and is forced to a logic one in single chip mode.

Other bits in CONFIG can be written at any time if SMOD = 1 (bootstrap or special test mode). If SMOD = 0 (single chip or expanded mode), these bits can only be written using the EEPROM programming sequence, and none of the bits are readable or active until latched via the next reset.

FREEZ is only active in expanded user mode.

ROMAD — ROM mapping control

1 (set) – ROM/*EPROM* addressed from $4000 to $FFFF.

0 (clear) – ROM/*EPROM* addressed from $0000 to $BFFF (expanded mode only).

In single chip mode, reset sets this bit.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

FREEZ — Address bus freeze in expanded user mode

1 (set) – The external bus is only active when externally mapped resources are accessed (expanded mode only).

0 (clear) – Normal operation.

To reduce RFI emissions, the address bus (on ports B and F) is not active while internal resources are being accessed, but will instead freeze on the last external address. At this time, the data bus port C is three-stated (high impedance) with weak pull-ups active, R/$\overline{W}$ is forced high and the E clock enable is pulled low. At reset, the address bus is initialized to $FFFE. Refer to Figure 3-2.

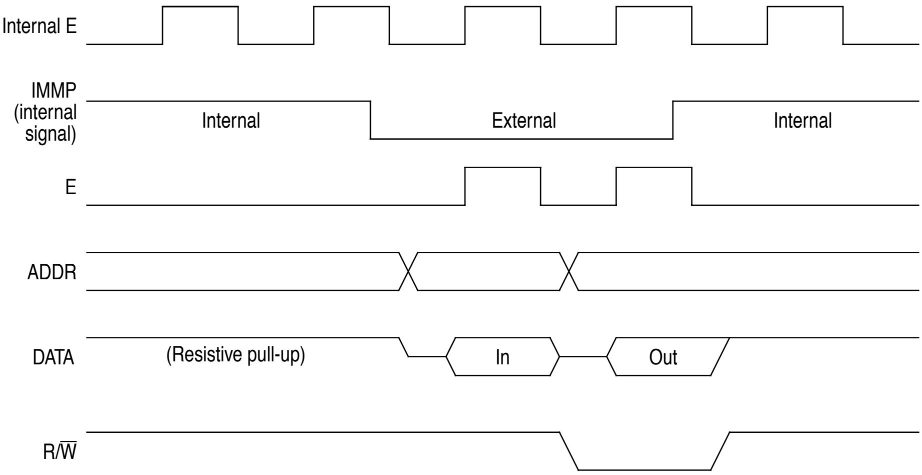


Figure 3-2 Example of expanded mode FREEZ actions

CLK4X — 4X clock enable

1 (set) – 4XCLK or EXTALi driven out on the 4XOUT pin (see Section 3.3.2.5)

0 (clear) – 4XOUT pin disabled.

Note: The 4XOUT pin is not available on 84-pin packaged devices.

PAREN — Pull-up assignment register enable (refer to Section 4)

1 (set) – Pull-ups can be enabled using PPAR.

0 (clear) – All pull-ups disabled (not controlled by PPAR).

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

NOSEC — EEPROM security disabled (refer to Section 3.4.4)

1 (set) – Disable security.

0 (clear) – Enable security.

NOCOP — COP system disable (refer to Section 10)

1 (set) – COP system disabled.

0 (clear) – COP system enabled (forces reset on timeout).

ROMON — ROM enable

1 (set) – ROM/*EPROM* included in the memory map.

0 (clear) – ROM/*EPROM* excluded from the memory map.

In single chip mode, reset sets this bit. In special test mode, reset clears ROMON. *On the MC68HC711PH8*, and on the MC68HC11PH8 if selected by a mask option, ROMON can be modified in expanded and special test modes. In this case, care must be taken to include reset and interrupt vectors in both internal and external memory maps. The routines for altering ROMON should not be located at addresses in the internal ROM/*EPROM* memory range, but rather at different external ROM/*EPROM* addresses or in internal EEPROM.

EEON — EEPROM enable

1 (set) – EEPROM included in the memory map.

0 (clear) – EEPROM is excluded from the memory map.

3.3.2.2 INIT — RAM and I/O mapping register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
RAM & I/O mapping (INIT)	$003D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	0000 0000

The internal registers used to control the operation of the MCU can be relocated on 4K boundaries within the memory space with the use of 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 to only once within the first 64 E clock cycles after a reset. It then becomes a read-only register.

RAM[3:0] — RAM map position

These four bits, which specify the upper hexadecimal digit of the RAM address, control the position of the RAM in the memory map. The RAM can be positioned at the beginning of any 4K page in the memory map. Refer to Table 3-5.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

REG[3:0] — 128-byte register block position

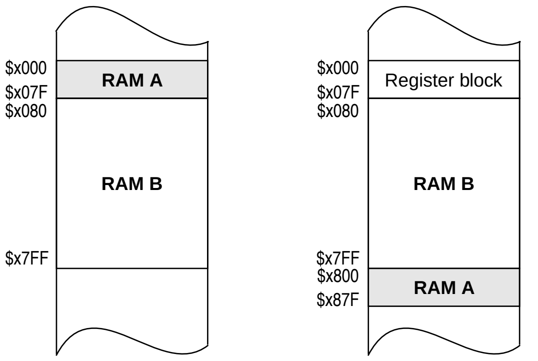
These four bits specify the upper hexadecimal digit of the address for the 128-byte block of internal registers. The register block is positioned at the beginning of any 4K page in the memory map. Refer to Table 3-5.

Table 3-5 RAM and register remapping

RAM[3:0]	Location
0000	$0000–$07FF
0001	$1000–$17FF
0010	$2000–$27FF
0011	$3000–$37FF
0100	$4000–$47FF
0101	$5000–$57FF
0110	$6000–$67FF
0111	$7000–$77FF
1000	$8000–$87FF
1001	$9000–$97FF
1010	$A000–$A7FF
1011	$B000–$B7FF
1100	$C000–$C7FF
1101	$D000–$D7FF
1110	$E000–$E7FF
1111	$F000–$F7FF

REG[3:0]	Location
0000	$0000–$007F
0001	$1000–$107F
0010	$2000–$207F
0011	$3000–$307F
0100	$4000–$407F
0101	$5000–$507F
0110	$6000–$607F
0111	$7000–$707F
1000	$8000–$807F
1001	$9000–$907F
1010	$A000–$A07F
1011	$B000–$B07F
1100	$C000–$C07F
1101	$D000–$D07F
1110	$E000–$E07F
1111	$F000–$F07F

When the memory map has the 128-byte register block mapped at the same location as RAM, the registers have priority and the RAM is relocated to the memory space immediately following the register block. This mapping feature keeps all the RAM available for use. Refer to Figure 3-3, which illustrates the overlap.



Register and RAM mapped to different 4K boundaries.

Register and RAM mapped to the same 4K boundary.

Figure 3-3 RAM and register overlap

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

3.3.2.3 INIT2 — EEPROM mapping and MI BUS delay register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
EEPROM mapping (INIT2)	$0037	EE3	EE2	EE1	EE0	STRX†	0	M2DL1	M2DL0	0000 0000

This register determines the location of EEPROM in the memory map and controls stretching of external accesses. INIT2 may be read at any time but bits 7–4 may be written only once after reset in normal modes (bits 3, 1 and 0 may be written at any time).

EE[3:0] — EEPROM map position

EEPROM is located at $xD00–$xFFF, where x is the hexadecimal digit represented by EE[3:0]. Refer to Table 3-6.

Table 3-6 EEPROM remapping

EE[3:0]	Location	EE[3:0]	Location
0000	$0D00–$0FFF	1000	$8D00–$8FFF
0001	$1D00–$1FFF	1001	$9D00–$9FFF
0010	$2D00–$2FFF	1010	$AD00–$AFFF
0011	$3D00–$3FFF	1011	$BD00–$BFFF
0100	$4D00–$4FFF	1100	$CD00–$CFFF
0101	$5D00–$5FFF	1101	$DD00–$DFFF
0110	$6D00–$6FFF	1110	$ED00–$EFFF
0111	$7D00–$7FFF	1111	$FD00–$FFFF

STRX — Stretch extended†

1 (set) – All external accesses are extended by one E clock cycle.

0 (clear) – Only external access from $0000 to $1FFF (ROMAD set) or from $C000 to $DFFF (ROMAD clear) are extended by one E clock cycle.

This bit only has meaning in expanded mode, and only if the STRCH bit in OPT2 is set (see Section 3.3.2.5).

Bit 2 — Not implemented; always reads zero.

M2DL1, M2DL0 — MI BUS delay select (refer to Section 6)

† *This bit is not present on early versions of the MC68HC711PH8. On those devices, bit 3 is not implemented and always reads zero, and the stretch function is controlled solely by the STRCH bit in OPT2 (see Section 3.3.2.5). Contact your local Motorola Sales Representative for further information.*

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

3.3.2.4 OPTION — System configuration options register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System config. options 1 (OPTION)	$0039	ADPU	CSEL	IRQE	DLY	CME	FCME	CR1	CR0	0001 0000

The 8-bit special-purpose OPTION register sets internal system configuration options during initialization. The time protected control bits IRQE, DLY, FCME and CR[1:0] can be written only once in the first 64 cycles after a reset and then they become read-only bits. This minimizes the possibility of any accidental changes to the system configuration. They may be written at any time in special modes.

ADPU — A/D power-up (refer to Section 9)

1 (set) – A/D system power enabled.

0 (clear) – A/D system disabled, to reduce supply current.

After enabling the A/D power, at least 100µs should be allowed for system stabilization.

CSEL — Clock select (refer to Section 9)

1 (set) – A/D, *EPROM* and EEPROM use internal RC clock source (about 1.5MHz).

0 (clear) – A/D, *EPROM* and EEPROM use system E clock (must be at least 1MHz).

This bit selects the clock source for the on-chip *EPROM*, EEPROM and A/D charge pumps. The on-chip RC clock should be used when the E clock frequency falls below 1MHz.

IRQE — Configure $\overline{\text{IRQ}}$ for falling edge sensitive operation

1 (set) – Falling edge sensitive operation.

0 (clear) – Low level sensitive operation.

DLY — Enable oscillator start-up delay

1 (set) – A stabilization delay is imposed as the MCU is started up from STOP mode (or from power-on reset).

0 (clear) – The oscillator start-up delay is bypassed and the MCU resumes processing within about four bus cycles. A stable external oscillator is required if this option is selected.

DLY is set on reset, so a delay is always imposed as the MCU is started up from power-on reset.

A mask option on the MC68HC11PH8 allows the selection of either a short or long delay time for power-on reset and exit from STOP mode; either 128 or 4064 bus cycles. *This option is not available on the MC68HC711PH8 where the delay time is 4064 bus cycles.*

CME — Clock monitor enable (refer to Section 10)

1 (set) – Clock monitor enabled.

0 (clear) – Clock monitor disabled.

In order to use both STOP and clock monitor, the CME bit should be set before executing STOP, then set again after recovering from STOP.

FCME — Force clock monitor enable (refer to Section 10)

1 (set) – Clock monitor enabled; cannot be disabled until next reset.

0 (clear) – Clock monitor follows the state of the CME bit.

When FCME is set, slow or stopped clocks will cause a clock failure reset sequence. To utilize STOP mode, FCME should always be cleared.

CR[1:0] — COP timer rate select bits (refer to Section 10)

These control bits determine a scaling factor for the watchdog timer.

3.3.2.5 OPT2 — System configuration options register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System config. options 2 (OPT2)	$0038	LIRDV	CWOM	STRCH	IRVNE	LSBF	SPR2	EXT4X	DISE	x00x 000t0

LIRDV — LIR driven

1 (set) – Enable $\overline{\text{LIR}}$ push-pull drive.

0 (clear) – $\overline{\text{LIR}}$ not driven on MODA/LIR pin.

This bit allows power savings in expanded modes by turning off the $\overline{\text{LIR}}$ output (it has no meaning in single chip or bootstrap modes). The $\overline{\text{LIR}}$ pin is driven low to indicate that execution of an instruction has begun. In order to detect consecutive instructions in a high speed application, this signal drives high for a quarter of a cycle to prevent false triggering. An external pull-up is required in expanded modes, while a hardwired VSS connection is possible in single chip modes. LIRDV is reset to zero in single chip modes, and to one in expanded modes.

CWOM — Port C wired-OR mode (refer to Section 4)

1 (set) – Port C outputs are open-drain.

0 (clear) – Port C operates normally.

STRCH — Stretch external accesses

1 (set) – Off-chip accesses are selectively extended by one E clock cycle.

0 (clear) – Normal operation.

When this bit is set, off-chip accesses of selected addresses are extended by one E clock cycle to allow access to slow peripherals. The E clock stretches externally, but the internal clocks are not affected, so that timers and serial systems are not corrupted. The state of the STRX† bit (in the INIT2 register) and the ROMAD bit (in the CONFIG register) determines which address range is affected. See Section 3.3.2.3.

Note: STRCH is cleared on reset; therefore a program cannot execute out of reset in a slow external ROM.

To use this feature, ROMON must be set on reset so that the device starts with internal ROM included in the memory map. STRCH should then be set.

Setting STRX means that all external accesses are stretched. If required (and allowed), ROMON can then be cleared so that internal ROM is not present in the memory map (see Section 3.4.3).

If STRX is cleared, then external accesses from $0000 to $1FFF (ROMAD set) or from $C000 to $DFFF (ROMAD cleared) are stretched.

STRCH has no effect in single chip and boot modes.

IRVNE — Internal read visibility/not E

IRVNE can be written once in any user mode. In **expanded modes**, IRVNE determines whether IRV is on or off (but has no meaning in user expanded secure mode, as IRV must be disabled). In special test mode, IRVNE is reset to one. In normal modes, IRVNE is reset to zero.

1 (set) – Data from internal reads is driven out of the external data bus.

0 (clear) – No visibility of internal reads on external bus.

In **single chip modes** this bit determines whether the E clock drives out from the chip.

1 (set) – E pin is driven low.

0 (clear) – E clock is driven out from the chip.

Refer to the following table for a summary of the operation immediately following reset.

† *The STRX bit is not present on early versions of the MC68HC711PH8; on those devices, setting STRCH means that external accesses either from $0000 to $FFFF or from $C000 to $DFFF are stretched, depending on the state of ROMAD. Contact your local Motorola Sales Representative for further information.*

TPG

Mode	IRVNE after reset	E clock after reset	IRV after reset	IRVNE affects only	IRVNE can be written
Single chip	0	On	Off	E	Once
Expanded	0	On	Off	IRV	Once
Boot	0	On	Off	E	Unlimited
Special test	1	On	On	IRV	Unlimited

LSBF — LSB-first enable (refer to Section 7)

1 (set) – Data is transferred LSB first.

0 (clear) – Data is transferred MSB first.

SPR2 — SPI clock rate select (refer to Section 7)

This bit adds a divide-by-four to the SPI clock chain.

EXT4X — 4XLCK or EXTAL clock output select

This bit can be written once and can be read at any time.

1 (set) – EXTALi clock output on the 4XOUT pin.

0 (clear) – 4XCLK clock output on the 4XOUT pin.

This bit selects which clock is to be output on the 4XOUT pin, when enabled by the CLK4X bit in CONFIG (see Section 3.3.2.1). On reset, or when BCS = 0, 4XCLK (the PLL output) is the same as EXTALi. Refer to Section 2-6. There is a phase delay between EXTALi and 4XOUT.

Note: The 4XOUT pin is not available on 84-pin packaged devices.

DISE — E clock output disable

This bit can be written once and can be read at any time.

1 (set) – No E clock output.

0 (clear) – E clock is output normally.

IRVNE allows E clock to be turned off in single chip modes. DISE has been added for expanded modes, but can be used in every mode. Writing a zero to this bit prevents accidental E clock turn-off in systems requiring this signal.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

3.3.2.6 BPROT — Block protect register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Block protect (BPROT)	$0035	BULKP	0	BPRT4	PTCON	BPRT3	BPRT2	BPRT1	BPRT0	1011 1111

BPROT prevents accidental writes to EEPROM and the CONFIG register. The bits in this register can be written to zero only once during the first 64 E clock cycles after reset in the normal modes; they can be set at any time. Once the bits are cleared, the EEPROM array and the CONFIG register can be programmed or erased. Setting the bits in the BPROT register to logic one protects the EEPROM and CONFIG register until the next reset. Refer to Table 3-7.

BULKP — Bulk erase of EEPROM protect

1 (set) – EEPROM cannot be bulk or row erased.

0 (clear) – EEPROM can be bulk erased normally.

Bit 6 — Not implemented; always reads zero.

BPRT4 — Block protect bit for top 256 bytes of EEPROM (see below)

PTCON — Protect for CONFIG register

1 (set) – CONFIG register cannot be programmed or erased.

0 (clear) – CONFIG register can be programmed or erased normally.

Note that, in special modes, CONFIG may be written regardless of the state of PTCON.

BPRT[4:0] — Block protect bits for EEPROM

1 (set) – Protection is enabled for associated block; it cannot be programmed or erased.

0 (clear) – Protection disabled for associated block.

Each of these five bits protects a block of EEPROM against writing or erasure, as follows:

Table 3-7 EEPROM block protect

Bit name	Block protected	Block size
BPRT0	$xD00–$xD1F	32 bytes
BPRT1	$xD20–$xD5F	64 bytes
BPRT2	$xD60–$xDDF	128 bytes
BPRT3	$xDE0–$xEFF	288 bytes
BPRT4	$xF00–$xFFF	256 bytes

3.3.2.7 TMSK2 — Timer interrupt mask register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt mask 2 (TMSK2)	$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	0000 0000

PR[1:0] are time-protected control bits and can be changed only once and then only within the first 64 bus cycles after reset in normal modes.

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

TOI — Timer overflow interrupt enable (Refer to Section 8)

1 (set) – Interrupt requested when TOF is set.

0 (clear) – TOF interrupts disabled.

RTII — Real-time interrupt enable (Refer to Section 8)

1 (set) – Interrupt requested when RTIF set.

0 (clear) – RTIF interrupts disabled.

PAOVI — Pulse accumulator overflow interrupt enable (Refer to Section 8)

1 (set) – Interrupt requested when PAOVF set.

0 (clear) – PAOVF interrupts disabled.

PAII — Pulse accumulator interrupt enable (Refer to Section 8)

1 (set) – Interrupt requested when PAIF set.

0 (clear) – PAIF interrupts disabled.

Bits [3, 2] — Not implemented; always read zero.

PR[1:0] — Timer prescaler select

These two bits select the prescale rate for the main 16-bit free-running timer system. These bits can be written only once during the first 64 E clock cycles after reset in normal modes, or at any time in special modes. Refer to the following table:

PR[1:0]	Prescale factor
0 0	1
0 1	4
1 0	8
1 1	16

3.4 *EPROM*, EEPROM and CONFIG register

3.4.1 EPROM

Using the on-chip EPROM programming feature requires an external power supply (V_{PPE}). Normal programming is accomplished using the EPROG register. Program EPROM at room temperature only and place an opaque label over the quartz window during and after programming.

The CSEL bit in the OPTION register selects an on-chip oscillator clock for programming the EPROM while operating at frequencies below 1MHz.

The erased state of each EPROM byte is $FF.

3.4.1.1 EPROG — EPROM programming control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
EPROM programming (EPROG)	$002B	MBE	0	ELAT	EXCOL	EXROW	0	0	EPGM	0000 0000

MBE — Multiple byte program enable

This bit may be read or written only in special modes; it will always read zero in normal modes.

1 (set) – Program 12 bytes with the same data.

0 (clear) – Normal programming.

EPROM is made up of three blocks of 16K bytes. When programming, address bits 4 and 7 are ignored, so that 4 addresses per block are programmed simultaneously. Address bits 14 and 15 are also ignored so that a total of twelve addresses are written at once, four in each 16K byte block. For example, with the EPROM mapped to $4000–$FFF, a write to $4026 will actually program $4026, $4036, $40A6, $40B6, $8026, $8036, $80A6, $80B6, $C026, $C036. $C0A6 and $C0B6 (i.e. %xx00 0000 x01x 0110).

Bits [6, 2, 1] *— Not implemented; always read zero.*

ELAT — EPROM latch control

ELAT may be read and written at any time.

*1 (set) – EPROM address and data buses configured for programming.
EPROM cannot be read.*

0 (clear) – EPROM address and data buses configured for normal operation.

When set, this bit causes the address and data for writes to the EPROM to be latched.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

EXCOL — Select extra columns

The EXCOL bit always reads zero in normal modes and may be read or written only in special modes.

1 (set) – User array disabled; extra column selected.

0 (clear) – User array selected.

The extra column may be accessed at bit 7; addresses use bits 15–5, bits 4–0 must be ones.

EXROW — Select extra rows

This bit always reads zero in normal modes and may be read or written only in special modes.

1 (set) – User array disabled; extra rows selected.

0 (clear) – User array selected.

There are six extra rows (two in each block). Addresses use bits 6–0, bits 11–7 must be zeros. (The high nibble determines which 16K block is accessed.)

EPGM — EPROM program command

This bit can be read at any time, but may only be written if ELAT is set.

1 (set) – Programming voltage (V_{PPE}) switched to the EPROM array.

0 (clear) – Programming voltage (V_{PPE}) disconnected from the EPROM array.

Note: If ELAT = 0 (normal operation) then EPGM = 0 (programming voltage disconnected).

3.4.1.2 EPROM programming

The EPROM may be programmed and verified in software, via the MCU, using the following procedure. The ROMON bit in the CONFIG register should be set. To use this method in special bootstrap mode, the external EPROM programming voltage must be applied on pin VPPE. On entry, A contains the data to be programmed and X contains the EPROM address.

```
EPROG  LDAB   #$20
       STAB   $002B  Set ELAT bit (PGM=0) to enable EPROM latches.
       STAA   $0, X  Store data to EPROM address
       LDAB   #$21
       STAB   $002B  Set EPGM bit, with ELAT=1, to enable prog. voltage
       JSR    DLYEP  Delay tEPROG
       CLR    $002B  Turn off programming voltage and set to READ mode
```

User-developed software can be uploaded through the SCI, or an EPROM programming utility resident in the bootstrap ROM can be used. To use the resident utility, bootload a three-byte program into RAM consisting of a single jump instruction to $BF00 (the starting address of a

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

resident EPROM programming utility), along with instructions to set the X and Y index registers to default values. The utility program receives programming data from an external host and puts it in EPROM. The value in IX determines programming delay time; for example, at 4 MHz operation, a delay constant of 8000 in IX will give a 2ms delay time. The value in IY is a pointer to the first address in EPROM to be programmed (normally = $4000). When the utility program is ready to receive programming data, it sends the host an $FF character; then it waits. When the host sees the $FF character, the EPROM programming data is sent, starting with location $4000. After the last byte to be programmed is sent and the corresponding verification data is returned, the programming operation is terminated by resetting the MCU.

3.4.2 EEPROM

The 768-byte on-board EEPROM is initially located from $0D00 to $0FFF after reset in all modes. It can be mapped to any other 4K page by writing to the INIT2 register. The EEPROM is enabled by the EEON bit in the CONFIG register. Programming and erasing are controlled by the PPROG register.

Unlike information stored in ROM, data in the 768 bytes of EEPROM can be erased and reprogrammed under software control. Because programming and erasing operations use an on-chip charge pump driven by V_{DD}, a separate external power supply is not required.

An internal charge pump supplies the programming voltage. Use of the block protect register (BPROT) prevents inadvertent writes to (or erases of) blocks of EEPROM (see Section 3.3.2.6). The CSEL bit in the OPTION register selects an on-chip oscillator clock for programming and erasing the EEPROM while operating at frequencies below 1MHz.

In special modes there is one extra row of EEPROM, which is used for factory testing. Endurance and data retention specifications do not apply to these cells.

The erased state of each EEPROM byte is $FF.

3.4.2.1 PPROG — EEPROM programming control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
EEPROM programming (PPROG)	$003B	ODD	EVEN	0	BYTE	ROW	ERASE	EELAT	EEPGM	0000 0000

Note: Writes to EEPROM addresses are inhibited while EEPGM is one. A write to a different EEPROM location is prevented while a program or erase operation is in progress.

ODD — Program odd rows in half of EEPROM (Test)

EVEN — Program even rows in half of EEPROM (Test)

If both ODD and EVEN are set to one then all odd and even rows in half of the EEPROM will be programmed with the same data, within one programming cycle.

Bit 5 — Not implemented; always reads zero.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

BYTE — EEPROM byte erase mode

1 (set) – Erase only one byte of EEPROM.

0 (clear) – Row or bulk erase mode used.

This bit may be read or written at any time.

ROW — EEPROM row/bulk erase mode (only valid when BYTE = 0)

1 (set) – Erase only one 16 byte row of EEPROM.

0 (clear) – Erase all 768 bytes of EEPROM.

This byte can be read or written at any time.

Table 3-8 Erase mode selection

Byte	Row	Action
0	0	Bulk erase (all 768 bytes)
0	1	Row erase (16 bytes)
1	0	Byte erase
1	1	Byte erase

ERASE — Erase/normal control for EEPROM

1 (set) – Erase mode.

0 (clear) – Normal read or program mode.

This byte can be read or written at any time.

EELAT — EEPROM latch control

1 (set) – EEPROM address and data bus set up for programming or erasing.

0 (clear) – EEPROM address and data bus set up for normal reads.

When the EELAT bit is cleared, the EEPROM can be read as if it were a ROM. The block protect register has no effect during reads. This bit can be read and written at any time.

EEPGM — EEPROM program command

1 (set) – Program or erase voltage switched on to EEPROM array.

0 (clear) – Program or erase voltage switched off to EEPROM array.

This bit can be read at any time but can only be written if EELAT = 1.

Note: If EELAT = 0 (normal operation) then EEPGM = 0 (programming voltage disconnected).

During EEPROM programming, the ROW and BYTE bits of PPROG are not used. If the frequency of the E clock is 1MHz or less, set the CSEL bit in the OPTION register. Remember that the EEPROM must be erased by a separate erase operation before programming. The following example of how to program an EEPROM byte assumes that the appropriate bits in BPROT have been cleared.

```
PROG   LDAB   #$02   EELAT=1
       STAB   $003B  Set EELAT bit
       STAA   $0D00  Store data to EEPROM address
       LDAB   #$03   EELAT=EEPGM=1
       STAB   $003B  Turn on programming voltage
       JSR    DLY10  Delay tEEPROG
       CLR    $003B  Turn off high voltage and set to READ mode
```

3.4.2.2 EEPROM bulk erase

To erase the EEPROM, ensure that the proper bits of the BPROT register are cleared, then complete the following steps using the PPROG register:

1) Write to PPROG with the ERASE, EELAT and appropriate BYTE and ROW bits set.

2) Write to the appropriate EEPROM address with any data. Row erase only requires a write to any location in the row. Bulk erase is accomplished by writing to any location in the array.

3) Write to PPROG with ERASE, EELAT, EEPGM and the appropriate BYTE and ROW bits set.

4) Delay for time t_{EEPROG} (See Section A.5.6).

5) Clear the EEPGM bit in PPROG to turn off the high voltage.

6) Clear the PPROG register to reconfigure the EEPROM address and data buses for normal operation.

The following is an example of how to bulk erase the 768-byte EEPROM. The CONFIG register is not affected in this example.

```
BULKE  LDAB   #$06   EELAT=ERASE=1
       STAB   $003B  Set EELAT bit
       STAA   $0D00  Store data to any EEPROM address
       LDAB   #$07   EELAT=ERASE=EEPGM=1
       STAB   $003B  Turn on programming voltage
       JSR    DLY10  Delay tEEPROG
       CLR    $003B  Turn off high voltage and set to READ mode
```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

3.4.2.3 EEPROM row erase

The following example shows how to perform a fast erase of 16 bytes of EEPROM:

```
ROWE   LDAB  #$0E   ROW=ERASE=EELAT=1
       STAB  $003B  Set to ROW erase mode
       STAB  0,X    Write any data to any address in ROW
       LDAB  #$0F   ROW=ERASE=EELAT=EEPGM=1
       STAB  $003B  Turn on high voltage
       JSR   DLY10  Delay tEEPROG
       CLR   $003B  Turn off high voltage and set to READ mode
```

3.4.2.4 EEPROM byte erase

The following is an example of how to erase a single byte of EEPROM:

```
BYTEE  LDAB  #$16   BYTE=ERASE=EELAT=1
       STAB  $003B  Set to BYTE erase mode
       STAB  0,X    Write any data to address to be erased
       LDAB  #$17   BYTE=ERASE=EELAT=EEPGM=1
       STAB  $003B  Turn on high voltage
       JSR   DLY10  Delay tEEPROG
       CLR   $003B  Turn off high voltage and set to READ mode
```

TPG

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

3.4.3 CONFIG register programming

Because the CONFIG register is implemented with EEPROM cells, use EEPROM procedures to erase and program this register. The procedure for programming is the same as for programming a byte in the EEPROM array, except that the CONFIG register address is used. CONFIG can be programmed or erased (including byte erase) while the MCU is operating in any mode, provided that PTCON in BPROT is clear. To change the value in the CONFIG register, complete the following procedure. Do not initiate a reset until the procedure is complete.

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

CONFIG — System configuration register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Configuration control (CONFIG)	$003F	ROMAD	FREEZ	CLK4X	PAREN	NOSEC	NOCOP	ROMON	EEON	xxxx xxxx

For a description of the bits contained in the CONFIG register refer to Section 3.3.2.1.

CONFIG is made up of EEPROM cells and static working latches. The operation of the MCU is controlled directly by these latches and not the EEPROM byte. When programming the CONFIG register, the EEPROM byte is accessed. When the CONFIG register is read, the static latches are accessed.

These bits can be read at any time. The value read is the one latched into the register from the EEPROM cells during the last reset sequence. A new value programmed into this register is not readable until after a subsequent reset sequence.

On the MC68HC711PH8, and on the MC68HC11PH8 if selected by a mask option, the ROMON bit can be written at any time if MDA = 1 (expanded mode or special test mode). It cannot be written in bootstrap mode, and is forced to a logic one in single chip mode.

Other bits in CONFIG can be written at any time if SMOD = 1 (bootstrap or special test mode). If SMOD = 0 (single chip or expanded mode), these bits can only be written using the EEPROM programming sequence, and none of the bits is readable or active until latched via the next reset.

FREEZ is only active in expanded user mode.

3.4.4 RAM and EEPROM security

The optional security feature protects the contents of EEPROM and RAM from unauthorized access. Data, codes, keys, a program, or a key portion of a program, can be protected against access. To accomplish this, the protection mechanism prevents operation of the device in special test mode. Only resident programs have unlimited access to the internal EEPROM and RAM and can read, write, or transfer the contents of these memories. To maintain RAM and EEPROM security, the following conditions should be satisfied:

– The internal ROM must be enabled and mapped at $4000–$FFFF, by setting the ROMON and ROMAD bits in the CONFIG register. This means that program execution starts after reset under control of the internal ROM (See Section 3.3.2.1).

– Access to external addresses should be restricted to data read or write. Program execution should not point from the internal resources to the external memory map.

– The FREEZ bit in the CONFIG register may be set to prevent internal address visibility on the ports (See Section 3.3.2.1).

Alternatively, EEPROM-only security is possible:

– In expanded mode, program execution is possible in external resources, but the EEPROM read or write access is restricted by internal hardware; instructions must be executed from internal ROM/*EPROM* in the range $4000 to $43FF (with ROMAD set) or in the range $0000 to $03FF (with ROMAD clear). Avoid using indexed addressing in this ROM/*EPROM* range, and clear temporary copies of EEPROM data before returning to the main program.

– As above, the FREEZ bit in the CONFIG register may be set to prevent internal address visibility on the ports (See Section 3.3.2.1).

Note: A mask option on the MC68HC11PH8 determines whether or not the security feature is available *(it is always available on the MC68HC711PH8)*. If the feature is available, then the secure mode can be invoked by programming the NOSEC bit to zero. Otherwise, the NOSEC bit is permanently set to one, disabling security.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

If the security feature is present and enabled and bootstrap mode is selected, then the following sequence is performed by the bootstrap program:

1) Output $FF on the SCI.
2) Turn block protect off. Clear BPROT register.
3) If EEPROM is enabled, erase it all.
4) Verify that the EEPROM is erased; if not, begin sequence again.
5) Write $FF to every RAM byte.
6) Erase the CONFIG register.

If all the above operations are successful, the bootloading process continues as if the device has not been secured.

CONFIG — System configuration register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Configuration control (CONFIG)	$003F	ROMAD	FREEZ	CLK4X	PAREN	NOSEC	NOCOP	ROMON	EEON	xxxx xxxx

For a description of the other bits contained in the CONFIG register refer to Section 3.3.2.1.

NOSEC — EEPROM security disabled

1 (set) – Disable security.

0 (clear) – Enable security.

With security enabled, selection of special test mode is prevented; single chip and user expanded modes may be accessed. If the MODA and MODB pins are configured for special test mode, the part will start in bootstrap mode.

THIS PAGE INTENTIONALLY LEFT BLANK

TPG

4
PARALLEL INPUT/OUTPUT

The MC68HC11PH8 has up to 54 input/output lines and 8 input-only lines, depending on the operating mode. To enhance the I/O functions, the data bus of this microcontroller is non-multiplexed. The following table is a summary of the configuration and features of each port.

Table 4-1 Port configuration

Port	Input pins	Output pins	Bidirectional pins	Alternate functions
A	–	–	8	Timer
B	–	–	8	High order address and LCD segment drivers
C	–	–	8	Data bus
D	–	–	6	SPI1 and SCI1
E	8	–	–	A/D converter
F	–	–	8	Low order address
G	–	–	8	$R/\overline{W}$ on PG7, LCDBP on PG6, SPI2 and SCI2 (with MI bus)
H	–	–	8	PWM and modulus timer clock inputs, keyboard interrupt

Note: Do not confuse pin function with the electrical state of that pin at reset. All general-purpose I/O pins that are configured as inputs at reset are in a high-impedance state and the contents of the port data registers are undefined; in port descriptions, a ‘u’ indicates this condition. The pin function is mode dependent.

4.1 Port A

Port A is an 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, port A pins are shared with timer functions, as shown in the following table.

Pin	Alternate function
PA0	IC3
PA1	IC2
PA2	IC1
PA3	OC5 and/or OC1, or IC4
PA4	OC4 and/or OC1
PA5	OC3 and/or OC1
PA6	OC2 and/or OC1
PA7	PAI and/or OC1

See Section 8 for more information.

On reset the pins are configured as general purpose high-impedance inputs.

4.1.1 PORTA — Port A data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port A data (PORTA)	$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	undefined

This is a read/write register and is not affected by reset. The bits may be read and written at any time, but, when a pin is allocated to its alternate function, a write to the corresponding register bit has no effect on the pin state.

4.1.2 DDRA — Data direction register for port A

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction A (DDRA)	$0001	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	0000 0000

DDA[7:0] — Data direction for port A

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

4.2 Port B

Port B is an 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, port B pins are used as the non-multiplexed high order address pins, as shown in the following table.

Pin	Alternate function
PB0	A8
PB1	A9
PB2	A10
PB3	A11
PB4	A12/LCD4
PB5	A13/LCD5
PB6	A14/ LCD6
PB7	A15/LCD7

In expanded or test mode, the pins become the high order address lines and port B is not included in the memory map.

The state of the pins on reset is mode dependent. In single chip or bootstrap mode, port B pins are high-impedance inputs with selectable internal pull-up resistors (see Section 4.9). In expanded or test mode, port B pins are high order address outputs and PORTB/DDRB are not in the memory map. Alternatively, four LCD segment drivers can be enabled, in all modes, on PB4–PB7 (See Section 2.12).

4.2.1 PORTB — Port B data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port B data (PORTB)	$0004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	undefined

The bits may be read and written at any time and are not affected by reset.

4.2.2 DDRB — Data direction register for port B

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction B (DDRB)	$0002	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	0000 0000

DDB[7:0] — Data direction for port B

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

4.3 Port C

Port C is an 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, port C pins are used as the non-multiplexed data bus pins, as shown in the following table.

Pin	Alternate function
PC0	D0
PC1	D1
PC2	D2
PC3	D3
PC4	D4
PC5	D5
PC6	D6
PC7	D7

In expanded or test mode, the pins become the data bus and port C is not included in the memory map.

The state of the pins on reset is mode dependent. In single chip or bootstrap mode, port C pins are high-impedance inputs. In expanded or test modes, port C pins are the data bus I/O and PORTC/DDRC are not in the memory map.

4.3.1 PORTC — Port C data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port C data (PORTC)	$0006	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	undefined

The bits may be read and written at any time and are not affected by reset.

4.3.2 DDRC — Data direction register for port C

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction C (DDRC)	$0007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	0000 0000

DDC[7:0] — Data direction for port C

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

4.4 Port D

Port D is a 6-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, port D pins are shared with SCI1 and SPI1 functions, as shown in the following table.

Pin	Alternate function	
PD0	RXD1	See Section 5 for more information.
PD1	TXD1	
PD2	MISO1	See Section 7 for more information.
PD3	MOSI1	
PD4	SCK1	
PD5	$\overline{SS1}$	

On reset the pins are configured as general purpose high-impedance inputs.

4.4.1 PORTD — Port D data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port D data (PORTD)	$0008	0	0	PD5	PD4	PD3	PD2	PD1	PD0	undefined

This is a read/write register and is not affected by reset. The bits may be read and written at any time, but, when a pin is allocated to its alternate function, a write to the corresponding register bit has no effect on the pin state.

4.4.2 DDRD — Data direction register for port D

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction D (DDRD)	$0009	0	0	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	0000 0000

Bits [7:6] — Reserved; always read zero

DDD[5:0] — Data direction for port D

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

4.5 Port E

Port E is an 8-bit input-only port. In addition to their input capability, port E pins are shared with A/D functions, as shown in the following table.

Pin	Alternate function
PE0	AD0
PE1	AD1
PE2	AD2
PE3	AD3
PE4	AD4
PE5	AD5
PE6	AD6
PE7	AD7

See Section 9 for more information.

On reset the pins are configured as general purpose high-impedance inputs.

4.5.1 PORTE — Port E data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port E data (PORTE)	$000A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	undefined

This is a read-only register and is not affected by reset. The bits may be read at any time.

Note: As port E shares pins with the A/D converter, a read of this register may affect any conversion currently in progress, if it coincides with the sample portion of the conversion cycle. Hence, normally port E should not be read during the sample portion of any conversion.

4.6 Port F

Port F is an 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, port F pins are used as the non-multiplexed low order address pins, as shown in the following table.

Pin	Alternate function
PF0	A0
PF1	A1
PF2	A2
PF3	A3
PF4	A4
PF5	A5
PF6	A6
PF7	A7

In expanded or test mode, the pins become the low order address and port F is not included in the memory map.

The state of the pins on reset is mode dependent. In single chip or bootstrap mode, port F pins are high-impedance inputs with selectable internal pull-up resistors (see Section 4.9). In expanded or test modes, port F pins are low order address outputs and PORTF/DDRF are not in the memory map.

4.6.1 PORTF — Port F data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port F data (PORTF)	$0005	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	undefined

The bits may be read and written at any time and are not affected by reset.

4.6.2 DDRF — Data direction register for port F

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction F (DDRF)	$0003	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	0000 0000

DDF[7:0] — Data direction for port F

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

4.7 Port G

Port G is an 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, port G pins are shared with R/$\overline{W}$, LCD, SCI2 (with MI-bus) and SPI2 functions, as shown in the following table.

Pin	Alternate function
PG0	RXD2
PG1	TXD2
PG2	MISO2
PG3	MOSI2
PG4	SCK2
PG5	$\overline{SS2}$
PG6	LCDBP
PG7	R/$\overline{W}$

PG0–PG1: See Section 5 for more information.

PG2–PG5: See Section 7 for more information.

PG6–PG7: See Section 2 for more information.

Pins PG[6:0] are high-impedance inputs with software selectable pull-up resistors, as is PG7 in single chip and bootstrap modes (see Section 4.9). In expanded or test modes, PG7 is the R/$\overline{W}$ output.

4.7.1 PORTG — Port G data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port G data (PORTG)	$007E	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	undefined

This is a read/write register and is not affected by reset. The bits may be read and written at any time, but, when a pin is allocated to its alternate function, a write to the corresponding register bit has no effect on the pin state.

4.7.2 DDRG — Data direction register for port G

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction G (DDRG)	$007F	DDG7	DDG6	DDG5	DDG4	DDG3	DDG2	DDG1	DDG0	0000 0000

DDG[7:0] — Data direction for port G

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

4.8 Port H

Port H is an 8-bit bidirectional port, with both data and data direction registers. In addition to their I/O capability, port H pins are shared with modulus timer and PWM functions, as shown in the following table. Each port H pin configured as an input can be used as a keyboard interrupt, if enabled (See Section 4.8.3).

Pin	Alternate function
PH0	PW1
PH1	PW2
PH2	PW3
PH3	PW4
PH4	–
PH5	–
PH6	Modulus timer C clock input
PH7	Modulus timer B clock input

See Section 8 for more information.

On reset the pins are configured as general purpose high-impedance inputs with selectable internal pull-ups (see Section 4.9).

4.8.1 PORTH — Port H data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port H data (PORTH)	$007C	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	undefined

This is a read/write register and is not affected by reset. The bits may be read and written at any time, but when one of the pins PH[3:0] is allocated to its alternate function of PWM channel, a write to the corresponding register bit has no affect on the pin state.

4.8.2 DDRH — Data direction register for port H

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Data direction H (DDRH)	$007D	DDH7	DDH6	DDH5	DDH4	DDH3	DDH2	DDH1	DDH0	0000 0000

DDH[7:0] — Data direction for port H

1 (set) – The corresponding pin is configured as an output.

0 (clear) – The corresponding pin is configured as an input.

4.8.3 Wired-OR interrupt

4.8.3.1 WOIEH — WOI enable (WOIEH)

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Wired-OR interrupt enable (WOIEH)	$007B	IEH7	IEH6	IEH5	IEH4	IEH3	IEH2	IEH1	IEH0	0000 0000

IEHx — Port H pin x wired-OR interrupt enable

1 (set) – PHx wired-OR interrupt enabled.

0 (clear) – PHx wired-OR interrupt disabled.

With the wired-OR interrupt function enabled, any port H pin configured as an input may be used as a keyboard interrupt. A high to low transition on an enabled pin (with all other enabled pins high) will result in an interrupt. The wired-OR interrupt flag (bit 4 of the port pull-up assignment register) indicates that an interrupt has occurred (see Section 4.9.1). A wired-OR interrupt can wake the MCU from STOP or WAIT mode.

4.9 Internal pull-up resistors

Four of the ports (B, F, G and H) have internal, software selectable pull-up resistors under control of the port pull-up assignment register (PPAR).

4.9.1 PPAR — Port pull-up assignment register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Port pull-up assignment (PPAR)	$002C	0	0	0	HWOIF	HPPUE	GPPUE	FPPUE	BPPUE	0000 1111

Bits [7:5] — Not implemented; always read zero.

HWOIF — Port H wired-OR interrupt flag

1 (set) – Port H keyboard interrupt request.

0 (clear) – No port H keyboard interrupt request.

This bit is cleared by a write to the PPAR register with HWOIF **set**. When this function is used, care must be taken when changing pull-up enable bits to prevent accidental clearing of this flag.

xPPUE — Port x pin pull-up enable

These bits control the on-chip pull-up devices connected to all the pins on I/O ports B, F, G and H. They are collectively enabled or disabled via the PAREN bit in the CONFIG register (see Section 4.10.2).

1 (set) – Port x pin on-chip pull-up devices enabled.

0 (clear) – Port x pin on-chip pull-up devices disabled.

Note: FPPUE and BPPUE have no effect in expanded mode since ports F and B are dedicated address bus or LCD outputs.

Note: When the SCI2 receiver is enabled, the associated pull-up on port G is disabled.

4.10 System configuration

One bit in each of the following registers is directly concerned with the configuration of the I/O ports. For full details on the other bits in the registers, refer to the appropriate section.

4.10.1 OPT2 — System configuration options register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System config. options 2 (OPT2)	$0038	LIRDV	CWOM	STRCH	IRVNE	LSBF	SPR2	EXT4X	DISE	x00x 0000

LIRDV — LIR driven (refer to Section 3)

1 (set) – Enable LIR push-pull drive.

0 (clear) – LIR not driven on MODA/LIR pin.

CWOM — Port C wired-OR mode

1 (set) – Port C outputs are open-drain.

0 (clear) – Port C operates normally.

STRCH — Stretch external accesses (refer to Section 3)

1 (set) – Off-chip accesses are extended by one E clock cycle.

0 (clear) – Normal operation.

IRVNE — Internal read visibility/not E (refer to Section 3)

1 (set) – Data from internal reads is driven out of the external data bus.

0 (clear) – No visibility of internal reads on external bus.

In **single chip mode** this bit determines whether the E clock drives out from the chip.

1 (set) – E pin is driven low.

0 (clear) – E clock is driven out from the chip.

LSBF — LSB first enable (refer to Section 7)

1 (set) – SPI1 data is transferred LSB first.

0 (clear) – SPI1 data is transferred MSB first.

SPR2 — SPI1 clock rate select (refer to Section 7)

EXT4X — 4XLCK or EXTAL clock output select (refer to Section 3

1 (set) – EXTALi clock output on the 4XOUT pin.

0 (clear) – 4XCLK clock output on the 4XOUT pin.

DISE— E clock output disable (refer to Section 3)

1 (set) – No E clock output.

0 (clear) – E clock is output normally.

4.10.2 CONFIG — System configuration register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Configuration control (CONFIG)	$003F	ROMAD	FREEZ	CLK4X	PAREN	NOSEC	NOCOP	ROMON	EEON	xxxx xxxx

ROMAD — ROM/*EPROM* mapping control (refer to Section 3)

1 (set) – ROM/*EPROM* addressed from $4000 to $FFFF.

0 (clear) – ROM/*EPROM* addressed from $0000 to $BFFF (expanded mode only).

FREEZ — Expanded user mode address bus freeze (refer to Section 3)

1 (set) – The external bus is only active when externally mapped resources are accessed (expanded mode only).

0 (clear) – Normal operation.

CLK4X — 4X clock enable (refer to Section 3)

1 (set) – 4XCLK or EXTALi driven out on the 4XOUT pin.

0 (clear) – 4XOUT pin disabled.

PAREN — Pull-up assignment register enable

1 (set) – Pull-ups can be enabled using PPAR register.

0 (clear) – All pull-ups disabled.

NOSEC — EEPROM security disabled (refer to Section 3)

1 (set) – Disable security.

0 (clear) – Enable security.

NOCOP — COP system disable (refer to Section 10)

1 (set) – COP system disabled.

0 (clear) – COP system enabled (forces reset on timeout).

ROMON — ROM/*EPROM* enable (refer to Section 3)

1 (set) – ROM/*EPROM* present in the memory map.

0 (clear) – ROM/*EPROM* disabled from the memory map.

EEON — EEPROM enable (refer to Section 3)

1 (set) – EEPROM is present in the memory map.

0 (clear) – EEPROM is disabled from the memory map.

5

SERIAL COMMUNICATIONS INTERFACE[†]

The serial communications interface (SCI) is a universal asynchronous receiver transmitter (UART). It has a non-return to zero (NRZ) format (one start, eight or nine data, and one stop bit) that is compatible with standard RS-232 systems.

The SCI shares I/O with two of port D's pins:

Pin	Alternate function
PD0	RXD1
PD1	TXD1

The SCI transmit and receive functions are enabled by TE and RE respectively, in SCCR2.

The SCI features enabled on this MCU include: 13-bit modulus prescaler, idle line detect, receiver-active flag, transmitter and receiver hardware parity. A block diagram of the enhanced baud rate generator is shown in Figure 5-1. See Table 5-1 for example baud rate control values.

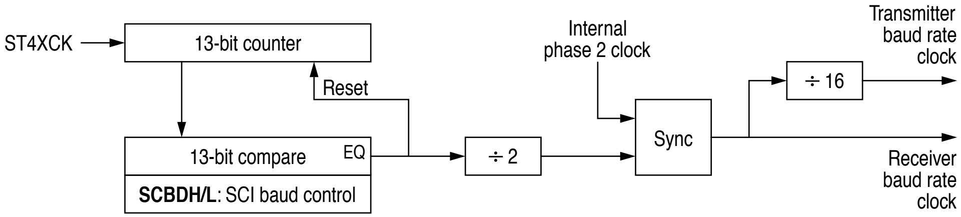


Figure 5-1 SCI baud rate generator circuit diagram

[†] The MC68HC11PH8 contains two serial communications interfaces, both having similar operation. For ease of reference, a full description of SCI1 (PD0/RXD1, PD1/TXD1) is given first, followed by a summary of SCI2 (Section 5.8), detailing its differences.

5.1 Data format

The serial data format requires the following conditions:

- An idle-line condition before transmission or reception of a message.
- A start bit, logic zero, transmitted or received, that indicates the start of each character.
- Data that is transmitted and received least significant bit (LSB) first.
- A stop bit, logic one, 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.)
- A break (defined as the transmission or reception of a logic zero for some multiple number of frames).

Selection of the word length is controlled by the M bit of SCCR1.

5.2 Transmit operation

The SCI transmitter includes a parallel data register (SCDRH/SCDRL) and a serial shift register. The contents of the shift register can only be written through the parallel data register. This double buffered operation allows a character to be shifted out serially while another character is waiting in the parallel data register to be transferred into the shift register. The output of the 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 5-2, shows the transmit serial shift register and the buffer logic at the top of the figure.

5.3 Receive operation

During receive operations, the transmit sequence is reversed. The serial shift register receives data and transfers it to the parallel receive data registers (SCDRH/SCDRL) as a complete word. This double buffered operation allows a character to be shifted in serially while another character is still in the serial data registers. An advanced data recovery scheme distinguishes valid data from noise in the serial data stream. The data input is selectively sampled to detect receive data, and majority sampling logic determines the value and integrity of each bit.

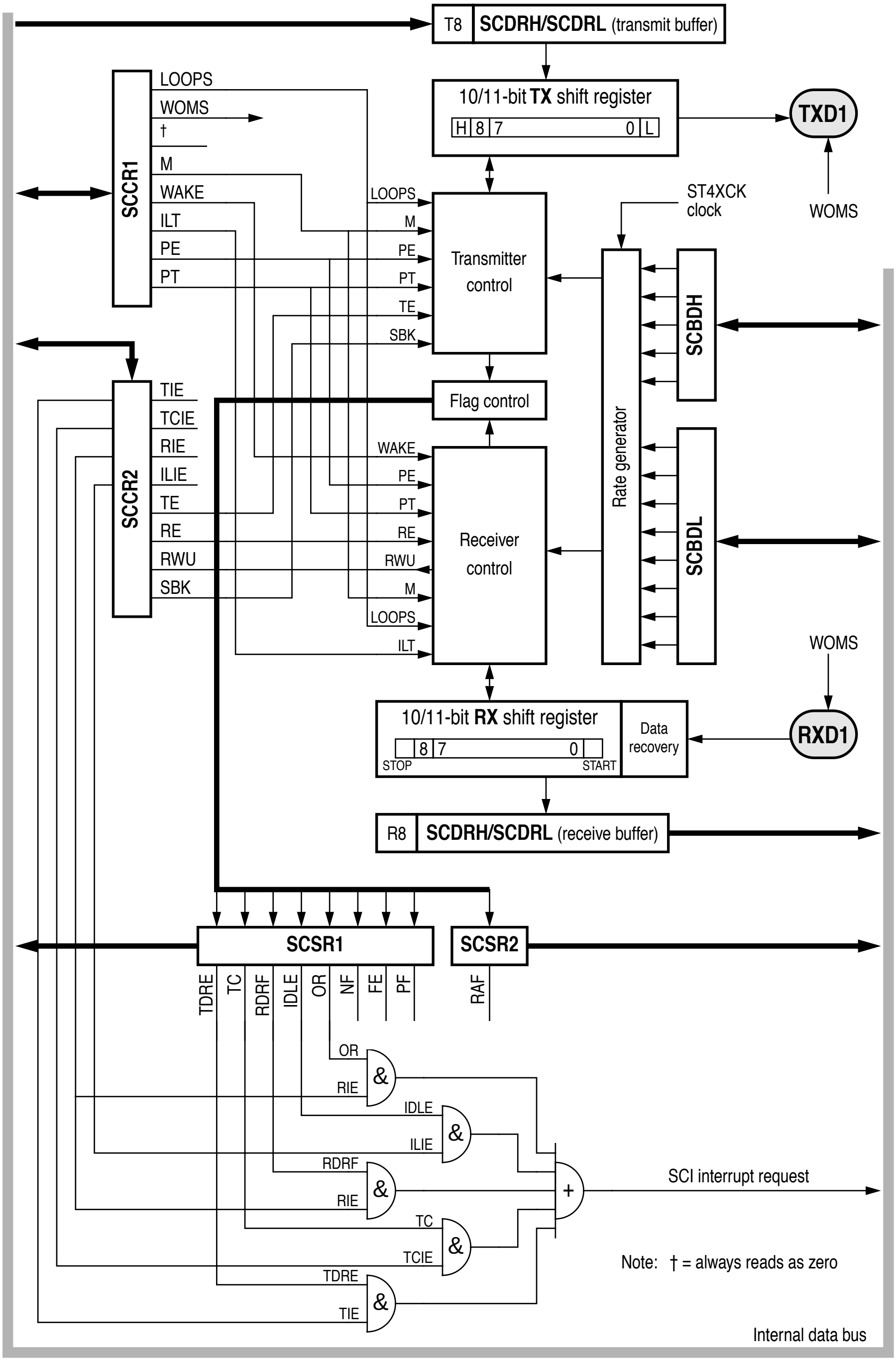


Figure 5-2 SCI1 block diagram

5.4 Wake-up feature

The wake-up feature reduces SCI service overhead in multiple receiver systems. Software for each receiver evaluates the first character or frame of each message. All receivers are placed in wake-up mode by writing a one to the RWU bit in the SCCR2 register. When RWU is set, the receiver-related status flags (RDRF, IDLE, OR, NF, FE, and PF) are inhibited (cannot be set). Although RWU can be cleared by a software write to SCCR2, to do so would be unusual. Normally RWU is set by software and is cleared automatically with hardware. Whenever a new message begins, logic alerts the dormant receivers to wake up and evaluate the initial character of the new message.

Two methods of wake-up are available: idle-line wake-up and address mark wake-up. During idle-line wake-up, a dormant receiver activates as soon as the RXD line becomes idle. In the address mark wake-up, logic one in the most significant bit (MSB) of a character activates all sleeping receivers. To use either receiver wake-up method, establish a software addressing scheme to allow the transmitting devices to direct messages to individual receivers or to groups of receivers. This addressing scheme can take any form as long as all transmitting and receiving devices are programmed to understand the same scheme.

5.4.1 Idle-line wake-up

Clearing the WAKE bit in SCCR1 register enables idle-line wake-up mode. In idle-line wake-up mode, all receivers are active (RWU bit in SCCR2 = 0) when each message begins. The first frames of each message are addressing frames. Each receiver in the system evaluates the addressing frames of a message to determine if the message is intended for that receiver. When a receiver finds that the message is not intended for it, it sets the RWU bit. Once set, the RWU control bit disables all but the necessary receivers for the remainder of the message, thus reducing software overhead for the remainder of that message. As soon as an idle line is detected by receiver logic, hardware automatically clears the RWU bit so that the first frames of the next message can be evaluated by all receivers in the system. This type of receiver wake-up requires a minimum of one idle frame time between messages, and no idle time between frames within a message.

5.4.2 Address-mark wake-up

Setting the WAKE bit in SCCR1 register enables address-mark wake-up mode. The address-mark wake-up method uses the MSB of each frame to differentiate between address information (MSB = 1) and actual message data (MSB = 0). All frames consist of seven information bits (eight bits if M bit in SCCR1 = 1) and an MSB which, when set to one, indicates an address frame. The first frames of each message are addressing frames. Receiver logic evaluates these marked frames to determine the receivers for which that message is intended. When a receiver finds that the message is not intended for it, it sets the RWU bit. Once set, the RWU control bit disables all but the necessary receivers for the remainder of the message, thus reducing software overhead for the remainder of that message. When the next message begins, its first frame will have the MSB set which will automatically clear the RWU bit and indicate that this is an addressing frame. This frame is always the first frame received after wake-up because the RWU bit is cleared before the stop bit for the first frame is received. This method of wake-up allows messages to include idle times, however, there is a loss in efficiency due to the extra bit time required for the address bit in each frame.

5.5 SCI error detection

Four error conditions can occur during SCI operation. These error conditions are: serial data register overrun, received bit noise, framing, and parity error. Four bits (OR, NF, FE, and PF) in serial communications status register 1 (SCSR1) indicate if one of these error conditions exists.

The overrun error (OR) bit is set when the next byte is ready to be transferred from the receive shift register to the serial data registers (SCDRH/SCDRL) and the registers are already full (RDRF bit is set). When an overrun error occurs, the data that caused the overrun is lost and the data that was already in serial data registers is not disturbed. The OR is cleared when the SCSR is read (with OR set), followed by a read of the SCI data registers.

The noise flag (NF) bit is set if there is noise on any of the received bits, including the start and stop bits. The NF bit is not set until the RDRF flag is set. The NF bit is cleared when the SCSR is read (with FE equal to one) followed by a read of the SCI data registers.

When no stop bit is detected in the received data character, the framing error (FE) bit is set. FE is set at the same time as the RDRF. If the byte received causes both framing and overrun errors, the processor only recognizes the overrun error. The framing error flag inhibits further transfer of data into the SCI data registers until it is cleared. The FE bit is cleared when the SCSR is read (with FE equal to one) followed by a read of the SCI data registers.

The parity error flag (PF) is set if received data has incorrect parity. The flag is cleared by a read of SCSR1 with PE set, followed by a read of SCDR.

5.6 SCI registers

There are eight addressable registers in the SCI. SCBDH, SCBDL, SCCR1, and SCCR2 are control registers. The contents of these registers control functions and indicate conditions within the SCI. The status registers SCSR1 and SCSR2 contain bits that indicate certain conditions within the SCI. SCDRH and SCDRL are SCI data registers. These double buffered registers are used for the transmission and reception of data, and are used to form the 9-bit data word for the SCI. If the SCI is being used with 7 or 8-bit data, only SCDRL needs to be accessed. Note that if 9-bit data format is used, the upper register should be written first to ensure that it is transferred to the transmitter shift register with the lower register.

5.6.1 SCBDH, SCBDL — SCI baud rate control registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI1 baud rate high (SCBDH)	$0070	BTST	BSPL	BRST	SBR12	SBR11	SBR10	SBR9	SBR8	0000 0000
SCI1 baud rate low (SCBDL)	$0071	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	0000 0100

The contents of this register determine the baud rate of the SCI.

BTST — Baud register test (Test mode only)

BSPL — Baud rate counter split (Test mode only)

BRST — Baud rate reset (Test mode only)

SBR[12:0] — SCI baud rate selects

Use the following formula to calculate SCI baud rate. Refer to the table of baud rate control values for example rates:

$$\text{SCI baud rate} = \frac{\text{ST4XCK}}{16 \times (2\text{BR})}$$

where the baud rate control value (BR) is the contents of SCBDH/L (BR = 1, 2, 3,... 8191).

For example, to obtain a baud rate of 1200 with a ST4XCK frequency of 12MHz, the baud register (SCBDH/L) should contain $0138 (see Table 5-1).

The clock rate generator is disabled if BR = 0, or if neither the receiver nor transmitter is enabled (both RE and TE in SCCR2 are cleared).

Writes to the baud rate registers will only be successful if the last (or only) byte written is SCBDL. The use of an STD instruction is recommended as it guarantees that the bytes are written in the correct order.

Note: ST4XCK may be the output of the PLL circuit or it may be the EXTAL input of the MCU (see Section 2.5, Figure 8-1 and Figure 8-2).

Table 5-1 Example SCI baud rate control values

Target baud rate	ST4XCK frequency					
	8 MHz		12 MHz		16 MHz	
	Dec value	Hex value	Dec value	Hex value	Dec value	Hex value
110	2272	$08E0	3409	$0D51	4545	$11C1
150	1666	$0682	2500	$09C4	3333	$0D05
300	833	$0341	1250	$04E2	1666	$0682
600	416	$01A0	625	$0271	833	$0341
1200	208	$00D0	312	$0138	416	$01A0
2400	104	$0068	156	$009C	208	$00D0
4800	52	$0034	78	$004E	104	$0068
9600	26	$001A	39	$0027	52	$0034
19200	13	$000D	20	$0014	26	$001A
38400					13	$000D

5.6.2 SCCR1 — SCI control register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI1 control 1 (SCCR1)	$0072	LOOPS	WOMS	0	M	WAKE	ILT	PE	PT	0000 0000

The SCCR1 register provides the control bits that determine word length and select the method used for the wake-up feature.

LOOPS — SCI loop mode enable

1 (set) – SCI transmit and receive are disconnected from TXD and RXD pins, and transmitter output is fed back into the receiver input.

0 (clear) – SCI transmit and receive operate normally.

Both the transmitter and receiver must be enabled to use the LOOP mode. When the LOOP mode is enabled, the TXD pin is driven high (idle line state) if the transmitter is enabled.

WOMS — Wired-OR mode for SCI pins (PD1, PD0)

1 (set) – TXD and RXD are open drains if operating as outputs.

0 (clear) – TXD and RXD operate normally.

Bit 5 — Not implemented; always reads zero

M — Mode (select character format)

1 (set) – Start bit, 9 data bits, 1 stop bit.

0 (clear) – Start bit, 8 data bits, 1 stop bit.

WAKE — Wake-up by address mark/idle

1 (set) – Wake-up by address mark (most significant data bit set).

0 (clear) – Wake-up by IDLE line recognition.

ILT — Idle line type

1 (set) – Long (SCI counts ones only after stop bit).

0 (clear) – Short (SCI counts consecutive ones after start bit).

This bit determines which of two types of idle line detection method is used by the SCI receiver. In short mode the stop bit and any bits that were ones before the stop bit will be considered as part of that string of ones, possibly resulting in erroneous or premature detection of an idle line condition. In long mode the SCI system does not begin counting ones until a stop bit is received.

PE — Parity enable

1 (set) – Parity enabled.

0 (clear) – Parity disabled.

PT — Parity type

1 (set) – Parity odd (an odd number of ones causes parity bit to be zero, an even number of ones causes parity bit to be one).

0 (clear) – Parity even (an even number of ones causes parity bit to be zero, an odd number of ones causes parity bit to be one).

5.6.3 SCCR2 — SCI control register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI1 control 2 (SCCR2)	$0073	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	0000 0000

The SCCR2 register provides the control bits that enable or disable individual SCI functions.

TIE — Transmit interrupt enable

1 (set) – SCI interrupt requested when TDRE status flag is set.

0 (clear) – TDRE interrupts disabled.

TCIE — Transmit complete interrupt enable

1 (set) – SCI interrupt requested when TC status flag is set.

0 (clear) – TC interrupts disabled.

RIE — Receiver interrupt enable

1 (set) – SCI interrupt requested when RDRF flag or the OR status flag is set.

0 (clear) – RDRF and OR interrupts disabled.

ILIE — Idle line interrupt enable

1 (set) – SCI interrupt requested when IDLE status flag is set.

0 (clear) – IDLE interrupts disabled.

TE — Transmitter enable

1 (set) – Transmitter enabled.

0 (clear) – Transmitter disabled.

RE — Receiver enable

1 (set) – Receiver enabled.

0 (clear) – Receiver disabled.

RWU — Receiver wake-up control

1 (set) – Wake-up enabled and receiver interrupts inhibited.

0 (clear) – Normal SCI receiver.

SBK — Send break

1 (set) – Break codes generated as long as SBK is set.

0 (clear) – Break generator off.

5.6.4 SCSR1 — SCI status register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI1 status 1 (SCSR1)	$0074	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF	1100 0000

The bits in SCSR1 indicate certain conditions in the SCI hardware and are automatically cleared by special acknowledge sequences.

TDRE — Transmit data register empty flag

1 (set) – SCDR empty.

0 (clear) – SCDR busy.

This flag is set when SCDR is empty. Clear the TDRE flag by reading SCSR1 with TDRE set and then writing to SCDR.

TC — Transmit complete flag

1 (set) – Transmitter idle.

0 (clear) – Transmitter busy.

This flag is set when the transmitter is idle (no data, preamble, or break transmission in progress). Clear the TC flag by reading SCSR1 with TC set and then writing to SCDR.

RDRF — Receive data register full flag

1 (set) – SCDR full.

0 (clear) – SCDR empty.

Once cleared, IDLE is not set again until the RXD line has been active and becomes idle again. RDRF is set if a received character is ready to be read from SCDR. Clear the RDRF flag by reading SCSR1 with RDRF set and then reading SCDR.

IDLE — Idle line detected flag

1 (set) – RXD line is idle.

0 (clear) – RXD line is active.

This flag is set if the RXD line is idle. Once cleared, IDLE is not set again until the RXD line has been active and becomes idle again. The IDLE flag is inhibited when RWU = 1. Clear IDLE by reading SCSR1 with IDLE set and then reading SCDR.

OR — Overrun error flag

1 (set) – Overrun detected.

0 (clear) – No overrun.

OR is set if a new character is received before a previously received character is read from SCDR. Clear the OR flag by reading SCSR1 with OR set and then reading SCDR.

NF — Noise error flag

1 (set) – Noise detected.

0 (clear) – Unanimous decision.

NF is set if the majority sample logic detects anything other than a unanimous decision. Clear NF by reading SCSR1 with NF set and then reading SCDR.

FE — Framing error

1 (set) – Zero detected.

0 (clear) – Stop bit detected.

FE is set when a zero is detected where a stop bit was expected. Clear the FE flag by reading SCSR1 with FE set and then reading SCDR.

PF — Parity error flag

1 (set) – Incorrect parity detected.

0 (clear) – Parity correct.

PF is set if received data has incorrect parity. Clear PF by reading SCSR1 with PE set and then reading SCDR.

5.6.5 SCSR2 — SCI status register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI1 status 2 (SCSR2)	$0075	0	0	0	0	0	0	0	RAF	0000 0000

In the SCSR2 only bit 0 is used, to indicate receiver active. The other seven bits always read zero.

Bits [7:1] — Not implemented; always read zero

RAF — Receiver active flag (read only)

1 (set) – A character is being received.

0 (clear) – A character is not being received.

5.6.6 SCDRH, SCDRL — SCI data high/low registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI1 data high (SCDRH)	$0076	R8	T8	0	0	0	0	0	0	undefined
SCI1 data low (SCDRL)	$0077	R7T7	R6T6	R5T5	R4T4	R3T3	R2T2	R1T1	R0T0	undefined

SCDRH/SCDRL is a parallel register that performs two functions. It is the receive data register when it is read, and the transmit data register when it is written. Reads access the receive data buffer and writes access the transmit data buffer. Data received or transmitted is double buffered.

If the SCI is being used with 7 or 8-bit data, only SCDRL needs to be accessed. Note that if 9-bit data format is used, the upper register should be written first to ensure that it is transferred to the transmitter shift register with the lower register.

R8 — Receiver bit 8

Ninth serial data bit received when SCI is configured for a nine data bit operation

T8 — Transmitter bit 8

Ninth serial data bit transmitted when SCI is configured for a nine data bit operation

Bits [5:0] — Not implemented; always read zero

R/T[7:0] — Receiver/transmitter data bits [7:0]

SCI data is double buffered in both directions.

5.7 Status flags and interrupts

The SCI transmitter has two status flags. These status flags can be read by software (polled) to tell when certain conditions exist. Alternatively, a local interrupt enable bit can be set to enable each of these status conditions to generate interrupt requests. Status flags are automatically set by hardware logic conditions, but must be cleared by software. This provides an interlock mechanism that enables logic to know when software has noticed the status indication. The software clearing sequence for these flags is automatic — functions that are normally performed in response to the status flags also satisfy the conditions of the clearing sequence.

TDRE and TC flags are normally set when the transmitter is first enabled (TE set to one). The TDRE flag indicates there is room in the transmit queue to store another data character in the transmit data register. The TIE bit is the local interrupt mask for TDRE. When TIE is zero, TDRE must be polled. When TIE and TDRE are one, an interrupt is requested.

The TC flag indicates the transmitter has completed the queue. The TCIE bit is the local interrupt mask for TC. When TCIE is zero, TC must be polled; when TCIE is one and TC is one, an interrupt is requested.

Writing a zero to TE requests that the transmitter stop when it can. The transmitter completes any transmission in progress before shutting down. Only an MCU reset can cause the transmitter to stop and shut down immediately. If TE is cleared when the transmitter is already idle, the pin reverts to its general purpose I/O function (synchronized to the bit-rate clock). If anything is being transmitted when TE is cleared, that character is completed before the pin reverts to general purpose I/O, but any other characters waiting in the transmit queue are lost. The TC and TDRE flags are set at the completion of this last character, even though TE has been disabled.

5.7.1 Receiver flags

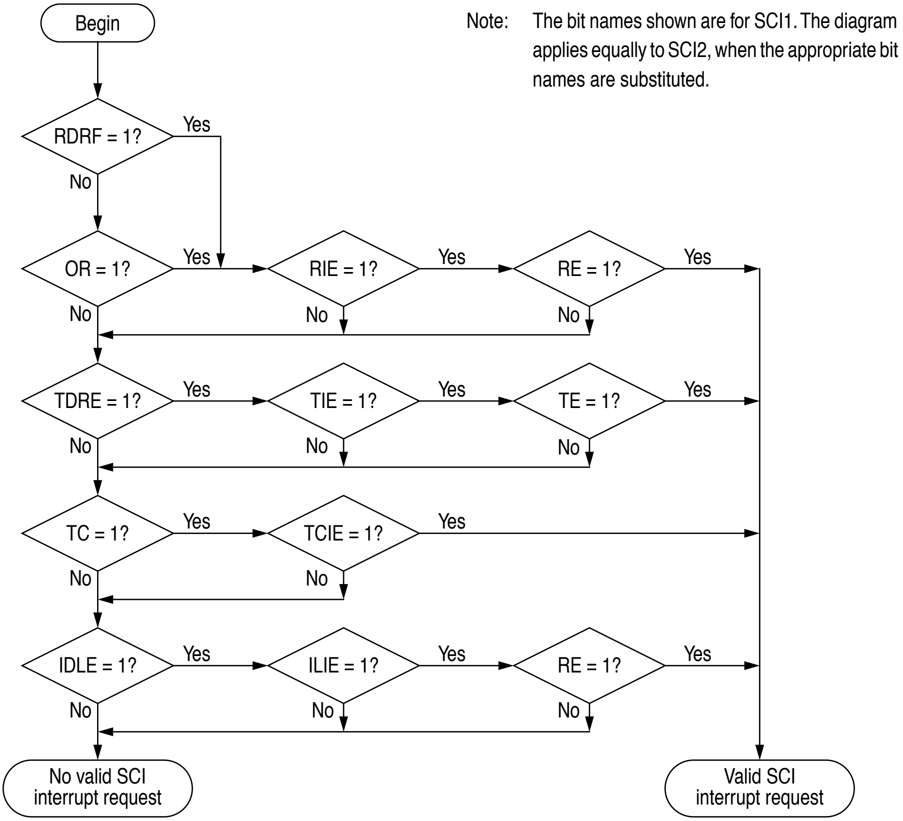
The SCI receiver has seven status flags, three of which can generate interrupt requests. The status flags are set by the SCI logic in response to specific conditions in the receiver. These flags can be read (polled) at any time by software. Refer to Figure 5-3, which shows SCI interrupt arbitration.

When an overrun takes place, the new character is lost, and the character that was in its way in the parallel receive data register (RDR) is undisturbed. RDRF is set when a character has been received and transferred into the parallel RDR. The OR flag is set instead of RDRF if overrun occurs. A new character is ready to be transferred into the RDR before a previous character is read from the RDR.

The NF, FE and PF flags provide additional information about the character in the RDR, but do not generate interrupt requests.

The receiver active flag (RAF) indicates that the receiver is busy.

The last receiver status flag and interrupt source come from the IDLE flag. The RXD line is idle if it has constantly been at logic one for a full character time. The IDLE flag is set only after the RXD line has been busy and becomes idle. This prevents repeated interrupts for the time RXD remains idle.



Note: The bit names shown are for SCI1. The diagram applies equally to SCI2, when the appropriate bit names are substituted.

Figure 5-3 Interrupt source resolution within SCI

5.8 SCI2

In addition to the subsystem described in the above paragraphs (SCI1), the MC68HC11PH8 has another, similar, SCI module (SCI2). This system is identical to SCI1, with the following exceptions:

- SCI2 shares I/O with two port G pins:

Pin	Alternate function
PG0	RXD2
PG1	TXD2

- The SCI2 transmit and receive functions are enabled by TE2 and RE2 respectively, in S2CR2.
- SCI1 functions and data are handled by a register block at $0070–$0077. The corresponding registers for SCI2 are at addresses $0050–$0057, as described in the following sections.
- The SCI2 baud rate register is at address $0050/51.
- In addition to the SCI functions,SCI2 is also used for MI BUS, controlled by bit 5 of S2CR1. Refer to Section 6 for full details of MI BUS operation.

5.8.1 S2BDH, S2BDL — SCI2 baud rate control registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI2/MI baud high (S2BDH)	$0050	B2TST	B2SPL	B2RST	S2B12	S2B11	S2B10	S2B9	S2B8	0000 0000
SCI2/MI baud low (S2BDL)	$0051	S2B7	S2B6	S2B5	S2B4	S2B3	S2B2	S2B1	S2B0	0000 0100

The contents of this register determine the baud rate for SCI2. For details of the bits and the corresponding baud rates, see Section 5.6.1. This register also controls the MI BUS clock rate (see Section 6).

5.8.2 S2CR1 — SCI2 control register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI2/MI control 1 (S2CR1)	$0052	LOPS2	WOMS2	MIE2	M2	WAKE2	ILT2	PE2	PT2	0000 0000

The S2CR1 register provides the control bits that determine word length and select the method used for the wake-up feature. Bit 5 has an MI BUS control function detailed below (for details of the other bits see Section 5.6.2).

WOMS2 — Wired-OR mode for SCI pins (PG1, PG0)

1 (set) – TXD2 and RXD2 are open drains if operating as inputs.

0 (clear) – TXD2 and RXD2 operate normally.

MIE2 — Motorola interface bus enable 2

1 (set) – MI BUS is enabled for this subsystem.

0 (clear) – The SCI functions normally.

When MIE2 is set, the SCI2 registers, bits and pins assume the functionality required for MI BUS.

5.8.3 S2CR2 — SCI2 control register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI2/MI control 2 (S2CR2)	$0053	TIE2	TCIE2	RIE2	ILIE2	TE2	RE2	RWU2	SBK2	0000 0000

The S2CR2 register provides the control bits that enable or disable individual SCI functions. For details of the bits, see Section 5.6.3.

5.8.4 S2SR1 — SCI2 status register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI2/MI status 1 (S2SR1)	$0054	TDRE2	TC2	RDRF2	IDLE2	OR2	NF2	FE2	PF2	1100 0000

The bits in S2SR1 indicate certain conditions in the SCI hardware and are automatically cleared by special acknowledge sequences. For details of the bits, see Section 5.6.4.

5.8.5 S2SR2 — SCI2 status register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI2/MI status 2 (S2SR2)	$0055	0	0	0	0	0	0	0	RAF2	0000 0000

In the S2SR2 only bit 0 is used, to indicate receiver active (see Section 5.6.5 for details). The other seven bits always read zero.

5.8.6 S2DRH, S2DRL — SCI2 data high/low registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI2/MI data high (S2DRH)	$0056	R8B	T8B	0	0	0	0	0	0	undefined
SCI2/MI data low (S2DRL)	$0057	R7T7B	R6T6B	R5T5B	R4T4B	R3T3B	R2T2B	R1T1B	R0T0B	undefined

S2DRH/S2DRL is a parallel register that performs two functions. It is the receive data register when it is read, and the transmit data register when it is written. Reads access the receive data buffer and writes access the transmit data buffer. Data received or transmitted is double buffered. See Section 5.6.6 for more details.

THIS PAGE INTENTIONALLY LEFT BLANK

6
MOTOROLA INTERCONNECT BUS (MI BUS)

The Motorola Interconnect Bus (MI BUS) is a serial communications protocol which supports distributed real-time control efficiently and with a high degree of noise immunity, at a typical bit rate for the data transfer of 20kHz. The MI BUS is suitable for medium speed networks requiring very low cost multiplex wiring; only one wire is required to connect to slave devices.[†]

The MI BUS uses a push-pull sequence to transfer data. The master device, which in this case is the MC68HC11PH8, sends a push field to the slave devices connected to the bus. The push field contains data plus an address that is recognized by one of the slaves. The slave addressed returns data which the master pulls from the MI BUS over the same wire. Specific details of the message format are covered later in this section. The MCU (master) can take the bus at any time, with a start bit that violates the rules of Manchester biphase encoding. Up to eight slave devices may be addressed by the MI BUS. Other features of MI BUS include message validation, error detection, and default value setting.

On the MC68HC11PH8 the MI BUS module shares the same pins on port G as the SCI2 module. Data is transmitted (or ‘pushed’) via the TXD pin, and received (‘pulled’) via the RXD pin. While data is being pushed, RXD will be disconnected from the receiver circuitry. The message frame is handled automatically in hardware. The MCU register interface is similar to that for the SCI.

Pin	Alternate function
PG0	RXD2
PG1	TXD2

MI BUS functions are enabled by MIE2 in S2CR1

[†] Related information on Motorola’s MI BUS is contained in the following Motorola publications:
EB409/D — The MI BUS and Product family for Multiplexing Systems
AN475/D — Single Wire MI BUS Controlling Stepper Motors
BR477/D — Smart Mover – Stepper Motors with Integrated Serial Bus Controller

6.1 Push-pull sequence

Communication between the MCU and the slave device always utilizes the same frame organization. First, the MCU sends serial data to the selected device. This data field is called the ‘push field’. At the end of the push field, the selected device automatically sends back to the MCU the data held during the push sequence. The MCU reads the serial data sent by the selected device. This data is called the ‘pull field’ and contains status information followed by the end-of-frame information from the selected device.

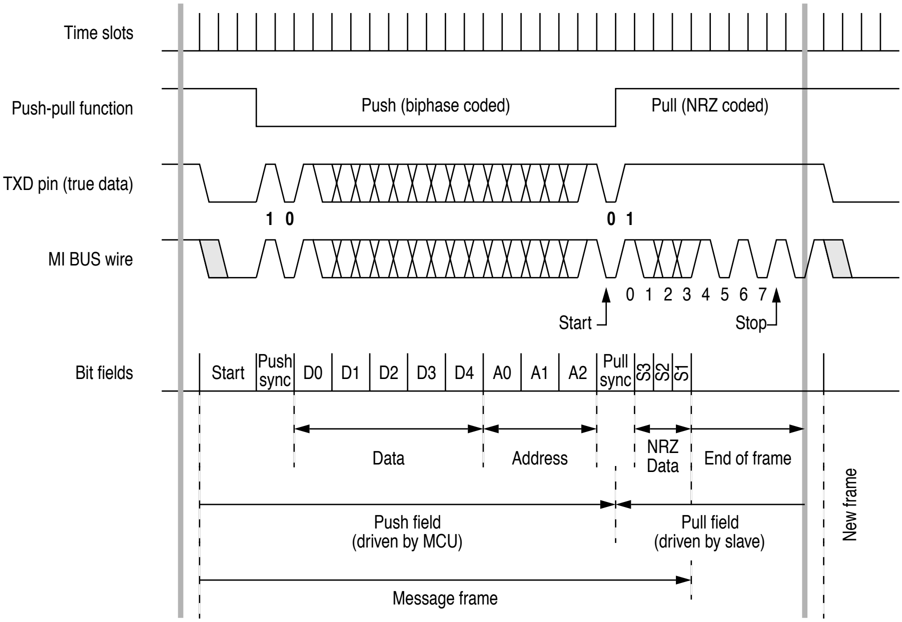


Figure 6-1 MI BUS timing

6.1.1 The push field

The push field consists of a start bit, a push synchronization bit, a push data field and a push address field. The start consists of three time slots having the dominant logical state ‘0’. The start marks the beginning of the message frame by violation of the rule of the Manchester code. The push synchronization bit consists of a biphase coded ‘0’. Biphase coding will be discussed later. The push data field consists of five bits of biphase coded data. The push address consists of three bits of biphase coded data. Data and address are written to the lower byte of the SCI data register (S2DRL). The push data occupies the lower five bits and the push address occupies the upper three bits of the register.

6.1.2 The pull field

The pull field consists of a pull synchronization bit, a pull data field and an end of frame. The pull synchronization bit is a biphase coded ‘1’ and is initiated by the MCU during the time slot after the last address bit of the push field. The pull data field consists of an NRZ coded transmission, each bit taking one time slot. Once shifted in, the pull data is stored in the lower byte of the SCI data register (S2DRL). The end-of-frame field is a square wave signal having a typical frequency of 20 kHz ± 1% tolerance (i.e. the bit rate of the push field) when the data sent to the selected device is valid.

6.2 Biphase coding

Manchester biphase L coding is used for the push field bits. Each bit requires two time slots to encode the logic value of the bit. This encoding allows the detection of a single error at the time slot level. Bits are encoded as follows:

1 (set) – In the first time slot, the logic level is set to zero, followed by a logic level one in the second time slot;

0 (clear) – In the first time slot, the logic level is set to one, followed by a logic level zero in the second time slot.

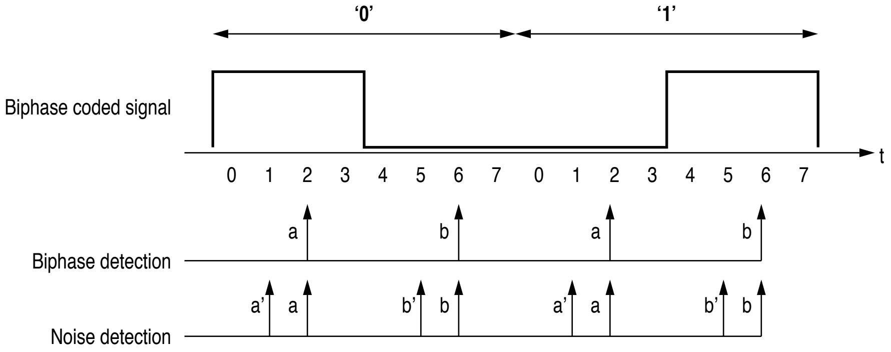


Figure 6-2 Biphase coding and error detection

6.3 Message validation

The communication between the MCU and the selected device is valid when the MCU reads a pull data field having correct codes (excluding the codes ‘111’ and ‘000’) followed by a square wave signal, having a frequency of 20kHz, contained in the end-of-frame information.

An MI BUS error is detected when the pull field contains the code ‘111’ followed by the end-of-frame permanently tied to logical state ‘1’. This means that the communication between the MCU and the selected device was not accomplished.

6.3.1 Controller detected errors

There are three different MI BUS error types which are detected by the selected slave device and are not mutually exclusive. The MCU cannot determine which error occurred.

- **Noise error** Slave devices take two samples in each time slot of the biphase encoded push field. An error occurs when the two samples for each time slot are not the same logical level.
- **Biphase error** Slave devices receiving the push field detect the biphase code. An error occurs when the two time slots of the biphase code do not yield a logical exclusive-OR function.
- **Field error** A field error is detected when the fixed-form of the push field is violated.

6.3.2 MCU detected errors

There is a fourth error that can be detected by the MCU. This error causes the noise flag (NF) to be asserted in the S2SR1 register during the push field sequence.

- **Bit error** A bit error can be detected by the MCU during the push field. The MI BUS serial system monitors the bus via on-chip hardware at the RXD pin at the same time as sending data. A bit error is detected at that bit time when the value monitored is different from the bit value sent.

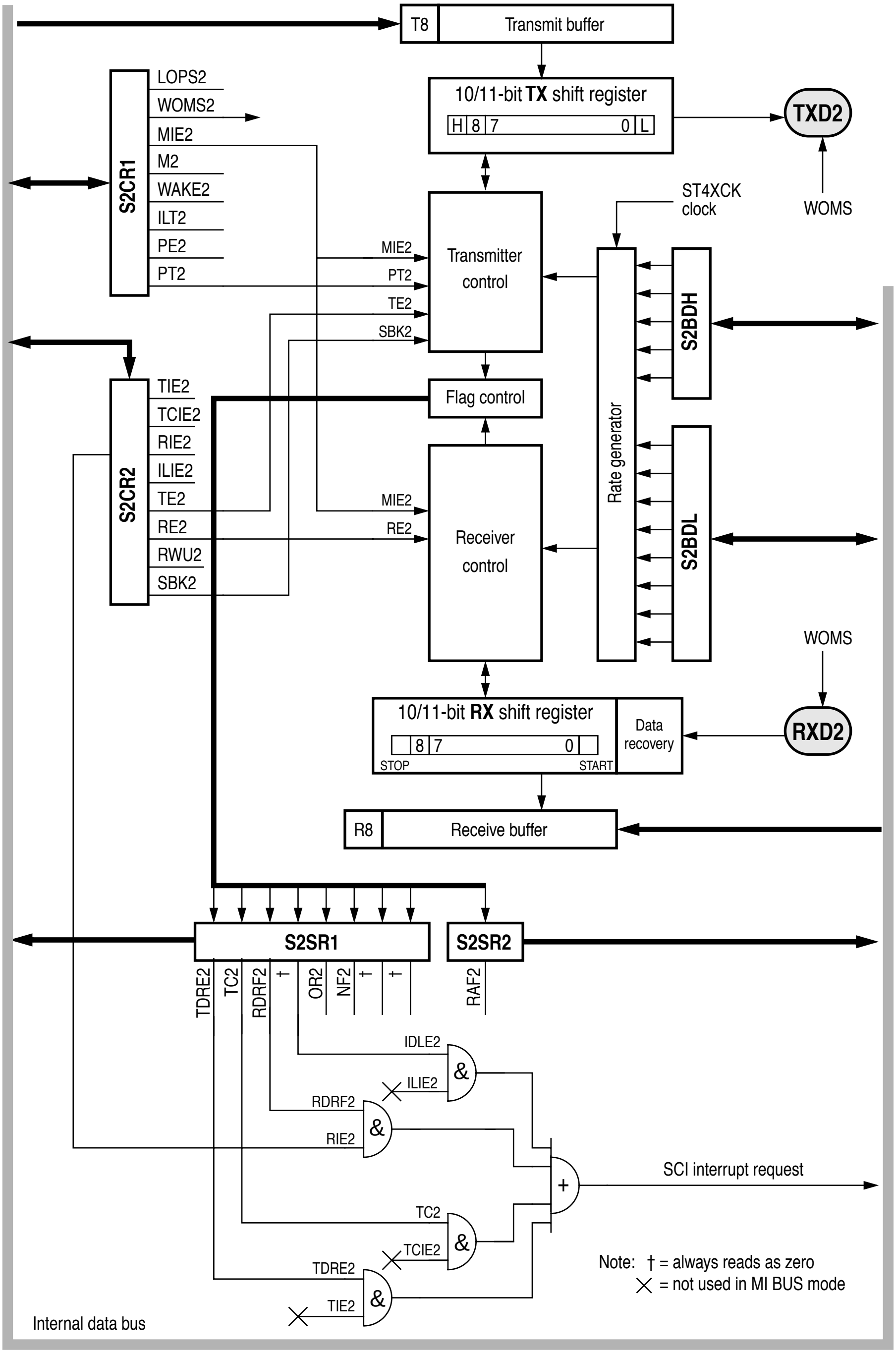


Figure 6-3 MI BUS block diagram

6.4 Interfacing to MI BUS

Physically the MI BUS consists of only a single wire. In the example shown in Figure 6-4, only a single transistor and a few passive components are required to connect up the MC68HC11PH8 for full MI BUS operation.

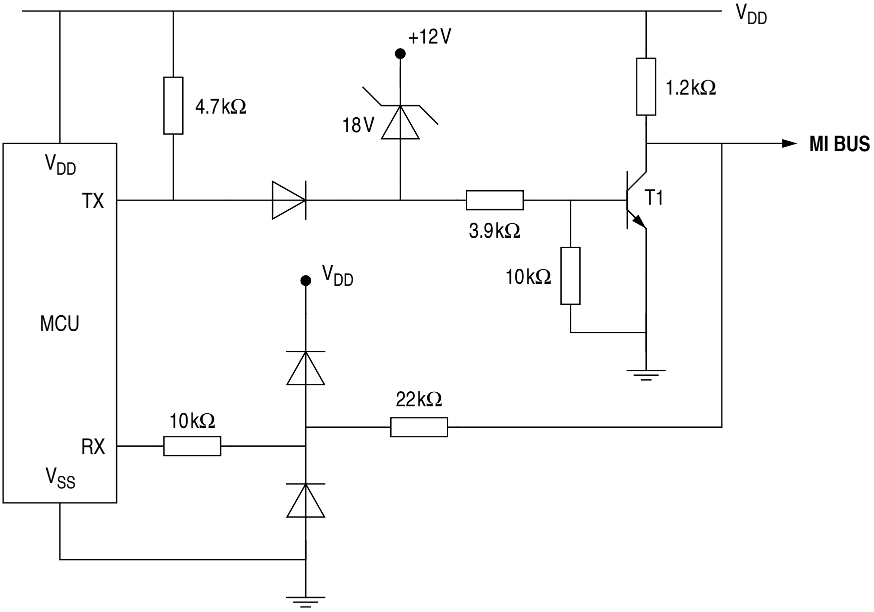


Figure 6-4 A typical interface between the MC68HC11PH8 and the MI BUS

The transistor serves both to drive the MI BUS during the push field and to protect the MCU TX pin from voltage transients generated in the wiring. Without the transistor, EMI could damage the TX pin. Similarly, the input pin (RX) is protected from EMI by clamping it to the MCU supply rails with two diodes. When a load dump occurs, the zener diode (18V) is switched on and hence turns the transistor on; this generates the logic ‘0’ state on the MI BUS. After eight time slots (200ms) of continuous ‘0’ state, all devices on the MI BUS will have their outputs disabled.

The MI BUS line can take two states, recessive or dominant. The recessive state (‘1’) is represented by 5V, through a pull-up resistor of 10kΩ. The dominant state (‘0’) is represented by a maximum 0.3V (V_{CESAT} of the transistor, T1).

The bus load depends on the number of devices on the bus. Each device has a pull-up resistor of 10kΩ. An external termination resistor is used to stabilize the load resistance of the bus at 600Ω.

6.5 MI BUS clock rate

The MI BUS clock rate is set via the SCI baud registers. To use the MI BUS, the ST4XCK clock frequency that drives the SCI clock generator must be selected to match the minimum resolution of the MI BUS logic. This is expressed by the following formula:

ST4XCK = 16 • 2n • (2 • Push_field_bit_rate) = 16 • 2n • 40kHz = n • 1280kHz

where ‘n’ is an integer and 20kHz is the minimum Push field bit rate for the MI BUS. Values for ST4XCK could be 1280kHz, 2560kHz, …, n • 1280kHz. The value ‘n’ is the modulus for the MI BUS baud register (see Section 6.6.2). ST4XCK may be the output of the PLL circuit or it may be the EXTAL input of the MCU. Refer to Section 2.5.

6.6 SCI2/MI BUS registers

MI BUS operation is controlled by the same group of registers as is used for the SCI. However the functions of some of the bits are modified when in MI BUS mode. A description of the registers, as applicable to the MI BUS function, is given here.

Note: In MI BUS mode, bits that have no meaning are reserved by Motorola, and are not described.

6.6.1 INIT2 — EEPROM mapping and MI BUS delay register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
EEPROM mapping (INIT2)	$0037	EE3	EE2	EE1	EE0	STRX	0	M2DL1	M2DL0	0000 0000

This register sets the MI BUS delay time. INIT2 may be read at any time but bits 7–4 may be written only once after reset in normal modes (bits 3, 1 and 0 may be written at any time).

EE[3:0] — EEPROM map position (Refer to Section 3.3.2.3.)

EEPROM is located at $xD00–$xFFF, where x is the hexadecimal digit represented by EE[3:0].

STRX — Stretch extended (Refer to Section 3.3.2.3)

1 (set) – All external accesses are extended by one E clock cycle.

0 (clear) – Only external access from $0000 to $1FFF (ROMAD set) or from $C000 to $DFFF (ROMAD clear) are extended by one E clock cycle.

Bit 2 — Not implemented, always read zero.

M2DL1:M2DL0 — MI BUS delay select

These bits are used to set up the delay for the start of the NRZ receive for MI BUS operation as shown (for a 20 kHz bit rate) in the following table.

M2DL1	M2DL0	Delay factor	Delay time[1]
0	0	1	1.5625 µs[2]
0	1	2	3.1250 µs
1	0	3	4.6875 µs
1	1	4	6.2500 µs

[1] 20 kHz bit rate requires 25 µs (40 kHz) time slots.

[2] 25 µs ÷ 16

6.6.2 S2BDH, S2BDL — MI BUS clock rate control registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI2/MI baud high (S2BDH)	$0050	B2TST	B2SPL	B2RST	S2B12	S2B11	S2B10	S2B9	S2B8	0000 0000
SCI2/MI baud low (S2BDL)	$0051	S2B7	S2B6	S2B5	S2B4	S2B3	S2B2	S2B1	S2B0	0000 0100

The contents of this register determine the clock rate for MI BUS.

S2B[12:0] — SCI baud rate/ MI BUS clock rate selects

Use the following formula to calculate MI BUS clock rate. Refer to the table of baud rate control values (see Table 5-1) for example rates:

$$\text{MI BUS clock rate } = \frac{\text{ST4XCK}}{16 \times (2\text{BR})}$$

where the baud rate control value (BR) is the contents of S2BDH/L (BR = 1, 2, 3,... 8191).

The clock rate generator is disabled if BR = 0, or if neither the receiver nor transmitter is enabled (both RE and TE in SCCR2 are cleared).

Writes to the baud rate registers will only be successful if the last (or only) byte written is SCBDL. The use of an STD instruction is recommended as it guarantees that the bytes are written in the correct order.

Note: ST4XCK may be the output of the PLL circuit or it may be the EXTAL input of the MCU. Selection is made by the MCS bit in the PLLCR (see Section 2.5).

6.6.3 S2CR1 — MI BUS control register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI2/MI control 1 (S2CR1)	$0052	–	WOMS2	MIE2	–	–	–	–	PT2	0000 0000

WOMS2 — Wired-OR mode for MI BUS pins (PG0, PG1)

1 (set) – TXD2 and RXD2 are open drains if operating as outputs.

0 (clear) – TXD2 and RXD2 operate normally.

MIE2 — Motorola interface bus enable 2

1 (set) – MI BUS is enabled for this subsystem.

0 (clear) – The SCI functions normally.

When MIE2 is set, the SCI2 registers, bits and pins assume the functionality required for MI BUS.

PT2 — MI BUS TX polarity (See Section 5.6.2)

1 (set) – MI BUS transmit pin will send inverted data.

0 (clear) – MI BUS transmit pin functions normally.

This control bit allows for different driver interfaces between the MCU and the MI BUS wire.

6.6.4 S2CR2 — MI BUS control register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI2/MI control 2 (S2CR2)	$0053	—	—	RIE2	—	TE2	RE2	—	SBK2	0000 0000

RIE2 — Receiver interrupt enable 2

1 (set) – MI BUS interrupt requested when RDRF2 flag is set.

0 (clear) – RDRF2 and OR2 interrupts disabled.

TE2 — Transmitter enable 2

1 (set) – Transmitter enabled and port pin dedicated to the MI BUS.

0 (clear) – Transmitter disabled.

RE2 — Receiver enable 2

1 (set) – Port pin dedicated to the MI BUS; the receiver is enabled by a pull sync and is inhibited during a push field.

0 (clear) – Receiver disabled.

SBK2 — Send break 2

1 (set) – MI transmit line is set low for 20 time slots.

0 (clear) – No action.

When an MI BUS wire is held low for eight or more time slots an internal circuit on any slave device connected to the bus may reset or preset the device with default values.

6.6.5 S2SR1 — MI BUS status register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI2/MI status 1 (S2SR1)	$0054	—	—	RDRF2	—	OR2	NF2	—	—	1100 0000

The bits in S2SR1 indicate certain conditions in the MI BUS hardware and are automatically cleared by special acknowledge sequences. The receive related flag bits in S2SR1 (RDRF2, OR2 and NF2) are cleared by a read of this register followed by a read of the transmit/receive data register. However, only those bits that were set when S2SR1 was read will be cleared by the subsequent read of the transmit/receive data register.

RDRF2 — Receive data register full flag 2

1 (set) – Contents of the receiver serial shift register have been transferred to the receiver data register.

0 (clear) – Contents of the receiver serial shift register have not been transferred to the receiver data register.

This bit is set when the contents of the receiver serial shift register have been transferred to the receiver data register.

The EOF (end-of-frame) during an MI BUS pull-field is a continuous square wave, which will result in multiple RDRFs. This may be dealt with in any of the following ways:

- By clearing the RIE2 mask, ignoring unneeded RDRF2s, initiating a push field, waiting for TDRE2† and then clearing the RDRF2;
- By clearing the RE2 bit when a pull field is complete, followed by setting the RE2 bit after the TDRE2† flag associated with the next push field is asserted;
- By disabling the MI BUS.

OR2 — Bit error 2

1 (set) – A bit error has been detected.

0 (clear) – No bit error has been detected.

This bit is set when a push field bit value on the MI BUS does not match the bit value that was sent. This is known as an MI BUS bit error. OR2 does not generate an interrupt request in MI BUS mode.

† Note that TDRE2 and TC2 will both behave in the same way as during normal SCI transmissions. The MI BUS will still be receiving when the TC2 bit becomes set, hence any queued transmission will not start until the current pull field has finished.
See also Section 5.6.4.

NF2 — Noise error flag 2

1 (set) – Noise detected.

0 (clear) – No noise detected.

This bit is set when noise is detected on the receive line during an MI BUS pull field.

6.6.6 S2SR2 — MI BUS2 status register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI/MI 2 status 2 (S2SR2)	$0055	0	0	0	0	0	0	0	RAF2	0000 0000

RAF2 — Receiver active flag (read only)

1 (set) – A character is being received.

0 (clear) – A character is not being received.

6.6.7 S2DRL — MI BUS2 data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SCI/MI 2 data low (S2DRL)	$0057	R7T7B	R6T6B	R5T5B	R4T4B	R3T3B	R2T2B	R1T1B	R0T0B	undefined
		0	1	0	1	S1	S2	S3	1	Pull field
		A2	A1	A0	D4	D3	D2	D1	D0	Push field

This register forms the 8-bit data/address word for the MI push field and contains the 3-bit data word received as the MI pull field.

R/T[7:0] — Receiver/transmitter data bits [7:0]

READ: Reads access the three bits of pull field data (stored in bits 3–1) of the read-only MI BUS receive data register. Bits [7:4, 0] are a fixed data pattern when a valid status and end-of-frame is returned. A valid status is represented by the following data pattern: 0101 xxx1 (bits 7–0), where ‘xxx’ is the status. All ones in the receive data register indicate that an error occurred on the MI BUS. Bits are received LSB first by the MCU, and the status bits map as shown in the above table.

WRITE: Writes access the eight bits of the write-only MI BUS transmit data register. MI BUS devices require a 5-bit data pattern followed by a 3-bit address pattern to be sent during the push field. The data pattern is mapped to the lowest five bits of the data register and the address to the highest three bits, as shown in the above table. Thus MI-data[4:0] is written to S2DRL[4:0] and MI-address[2:0] is written to S2DRL[7:5].

7
SERIAL PERIPHERAL INTERFACE[†]

The serial peripheral interface (SPI), an independent serial communications subsystem, allows the MCU to communicate synchronously with peripheral devices, such as transistor-transistor logic (TTL) shift registers, liquid crystal (LCD) display drivers, analog-to-digital converter subsystems, and 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, with data rates as high as one half of the E clock rate when configured as a master and as fast as the E clock rate when configured as a slave.

The SPI shares I/O with four of port D's pins and is enabled by SPE in the SPCR:

Pin	Alternate function
PD2	MISO1
PD3	MOSI1
PD4	SCK1
PD5	$\overline{\text{SS1}}$

7.1 Functional description

The central element in the SPI system is the block containing the shift register and the read data buffer (see Figure 7-1). 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).

[†] The MC68HC11PH8 contains two serial peripheral interfaces having similar operation. For ease of reference, a full description of SPI1 is given first, followed by a summary of SPI2 (Section 7.6).

7.2 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 7-2.

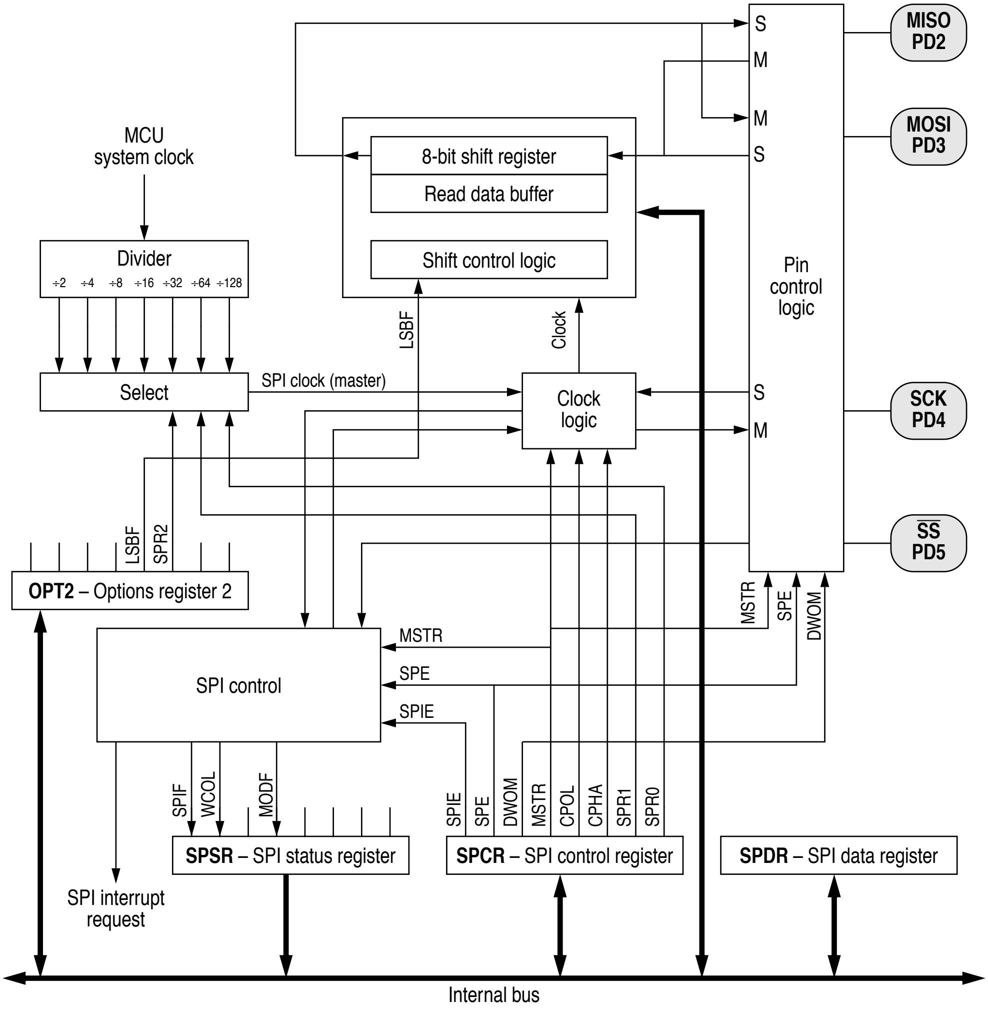
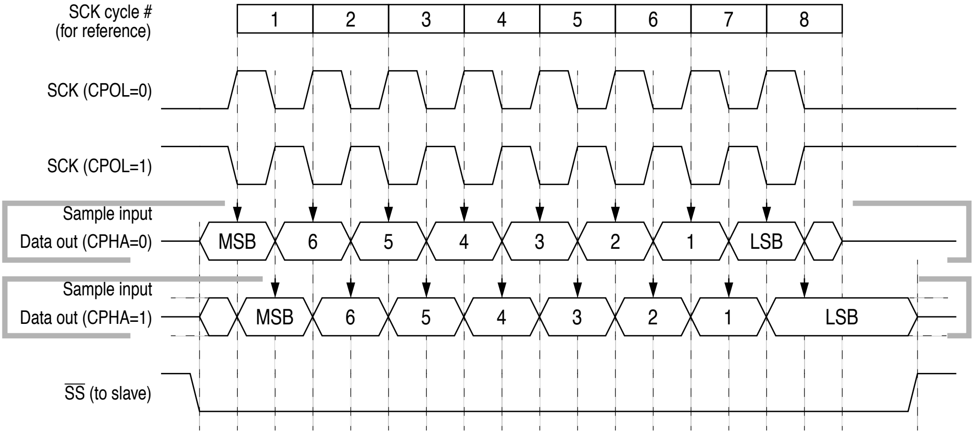


Figure 7-1 SPI block diagram



Note: this figure shows the LSBF=0 (default) case. If LSBF=1, data is transferred in the reverse order (LSB first).

Figure 7-2 SPI transfer format

7.2.1 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 zero, the $\overline{SS}$ line must be deasserted 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 one, the $\overline{SS}$ line can remain low between successive transfers.

7.3 SPI signals

The following paragraphs contain descriptions of the four SPI signals: master in slave out (MISO), master out slave in (MOSI), serial clock (SCK), and 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.

7.3.1 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.

7.3.2 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.

7.3.3 Serial clock

SCK, an input to a slave device, is generated by the master device and synchronizes data movement in and out of the device through the MOSI and MISO lines. Master and slave devices are capable of exchanging a byte of information during a sequence of eight clock cycles.

There are four possible timing relationships that can be chosen by using control bits CPOL and CPHA in the serial peripheral control register (SPCR). Both master and slave devices must operate with the same timing. The SPI clock rate select bits, SPR[1:0], in the SPCR of the master device, select the clock rate. In a slave device, SPR[1:0] have no effect on the operation of the SPI.

7.3.4 Slave select

The slave select $\overline{SS}$ input of a slave device must be externally asserted before a master device can exchange data with the slave device. $\overline{SS}$ must be low before data transactions begin and must stay low for the duration of the transaction.

The $\overline{SS}$ line of the master must be held high. If it goes low, a mode fault error flag (MODF) is set in the serial peripheral status register (SPSR). To disable the mode fault circuit, write a one in bit 5 of the port D data direction register. This sets the $\overline{SS}$ pin to act as a general-purpose output, rather than a dedicated input to the slave select circuit, thus inhibiting the mode fault flag. The other three lines are dedicated to the SPI whenever the serial peripheral interface is on.

The state of the master and slave CPHA bits affects the operation of $\overline{SS}$. CPHA settings should be identical for master and slave. When CPHA = 0, the shift clock is the OR of $\overline{SS}$ with SCK. In this clock phase mode, $\overline{SS}$ must go high between successive characters in an SPI message. When CPHA = 1, $\overline{SS}$ can be left low between successive SPI characters. In cases where there is only one SPI slave MCU, its $\overline{SS}$ line can be tied to V_{SS} as long as only CPHA = 1 clock mode is used.

7.4 SPI system errors

Two kinds of system errors can be detected by the SPI system. The first type of error arises in a multiple-master system when more than one SPI device simultaneously tries to be a master. This error is called a mode fault. The second type of error, write collision, indicates that an attempt was made to write data to the SPDR while a transfer was in progress.

When the SPI system is configured as a master and the $\overline{SS}$ input line goes to active low, a mode fault error has occurred — usually because two devices have attempted to act as master at the same time. In the case where more than one device is concurrently configured as a master, there is a chance of contention between two pin drivers. For push-pull CMOS drivers, this contention can cause permanent damage. The mode fault detection circuitry attempts to protect the device by disabling the drivers. The MSTR control bit in the SPCR and all four DDRD control bits associated with the SPI are cleared and an interrupt is generated (subject to masking by the SPIE control bit and the I bit in the CCR).

Other precautions may need to be taken to prevent driver damage. If two devices are made masters at the same time, the mode fault detector does not help protect either one unless one of them selects the other as slave. The amount of damage possible depends on the length of time both devices attempt to act as master.

A write collision error occurs if the SPDR is written while a transfer is in progress. Because the SPDR is not double buffered in the transmit direction, writes to SPDR cause data to be written directly into the SPI shift register. Because this write corrupts any transfer in progress, a write collision error is generated. The transfer continues undisturbed, and the write data that caused the error is not written to the shifter.

A write collision is normally a slave error because a slave has no control over when a master initiates a transfer. A master knows when a transfer is in progress, so there is no reason for a master to generate a write-collision error, although the SPI logic can detect write collisions in both master and slave devices.

The SPI configuration determines the characteristics of a transfer in progress. For a master, a transfer begins when data is written to SPDR and ends when SPIF is set. For a slave with CPHA equal to zero, a transfer starts when $\overline{SS}$ goes low and ends when $\overline{SS}$ returns high. In this case, SPIF is set at the middle of the eighth SCK cycle when data is transferred from the shifter to the parallel data register, but the transfer is still in progress until $\overline{SS}$ goes high. For a slave with CPHA equal to one, transfer begins when the SCK line goes to its active level, which is the edge at the beginning of the first SCK cycle. The transfer ends when SPIF is set, for a slave in which CPHA=1.

7.5 SPI registers

The three SPI registers, SPCR, SPSR, and SPDR, provide control, status, and data storage functions. Refer to the following information for a description of how these registers are organized.

7.5.1 SPCR — SPI control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI control (SPCR)	$0028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	0000 01uu

SPIE — Serial peripheral interrupt enable

1 (set) – A hardware interrupt sequence is requested each time SPIF or MODF is set.

0 (clear) – SPI interrupts are inhibited.

Set the SPIE bit to a one to request a hardware interrupt sequence each time the SPIF or MODF status flag is set. SPI interrupts are inhibited if this bit is clear or if the I bit in the condition code register is one.

SPE — Serial peripheral system enable

1 (set) – Port D [5:2] is dedicated to the SPI.

0 (clear) – Port D has its default I/O functions and the clock generator is stopped.

When the SPE bit is set, the port D pins 2, 3, 4, and 5 are dedicated to the SPI functions and lose their general purpose I/O functions. When the SPI system is enabled and expects any of PD[4:2] to be inputs then those pins will be inputs regardless of the state of the associated DDRD bits. If any of PD[4:2] are expected to be outputs then those pins will be outputs only if the associated DDRD bits are set. However, if the SPI is in the master mode, DDD5 determines whether PD5 is an error detect input (DDD5 = 0) or a general-purpose output (DDD5 = 1).

DWOM — Port D wired-OR mode

1 (set) – Port D [5:2] buffers configured for open-drain outputs.

0 (clear) – Port D [5:2] buffers configured for normal CMOS outputs.

MSTR — Master mode select

1 (set) – Master mode

0 (clear) – Slave mode

CPOL — Clock polarity

1 (set) – SCK is active low.

0 (clear) – SCK is active high.

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 7-2 and Section 7.2.1.

CPHA — Clock phase

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 7-2 and Section 7.2.1.

SPR1 and SPR0 — SPI clock rate selects

These two bits select the SPI clock rate, as shown in Table 7-1. Note that SPR2 is located in the OPT2 register, and that its state on reset is zero.

Table 7-1 SPI clock rates

SPR[2:0]	E clock divide ratio	SPI clock frequency (≡ baud rate) for:		
		E = 2 MHz	E = 3 MHz	E = 4 MHz
0 0 0	2	1.0 MHz	1.5 MHz	2.0 MHz
0 0 1	4	500 kHz	750kHz	1.0 MHz
0 1 0	16	125 kHz	187.5 kHz	250 kHz
0 1 1	32	62.5 kHz	93.7 kHz	125 kHz
1 0 0	8	250 kHz	375 kHz	500 kHz
1 0 1	16	125 kHz	187.5 kHz	250 kHz
1 1 0	64	31.3 kHz	46.9 kHz	62.5 kHz
1 1 1	128	15.6 kHz	23.4 kHz	31.3 kHz

7.5.2 SPSR — SPI status register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI status (SPSR)	$0029	SPIF	WCOL	0	MODF	0	0	0	0	0000 0000

SPIF — SPI interrupt complete flag

1 (set) – Data transfer to external device has been completed.

0 (clear) – No valid completion of data transfer.

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

1 (set) – Write collision.

0 (clear) – No write collision.

Clearing the WCOL bit is accomplished by reading the SPSR (with WCOL set) followed by an access of SPDR. Refer to Section 7.3.4 and Section 7.4.

MODF — Mode fault

1 (set) – Mode fault.

0 (clear) – No mode fault.

To clear the MODF bit, read the SPSR (with MODF set), then write to the SPCR. Refer to Section 7.3.4 and Section 7.4.

Bits [5, 3:0] — Not implemented; always read zero.

7.5.3 SPDR — SPI data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI data (SPDR)	$002A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined

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.

SPI is double buffered in and single buffered out.

7.5.4 OPT2 — System configuration options register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System config. options 2 (OPT2)	$0038	LIRDV	CWOM	STRCH	IRVNE	LSBF	SPR2	EXT4X	DISE	x00x 0000

LIRDV — LIR driven (refer to Section 3)

1 (set) – Enable LIR push-pull drive.

0 (clear) – LIR not driven on MODA/LIR pin.

CWOM — Port C wired-OR mode (refer to Section 4)

1 (set) – Port C outputs are open-drain.

0 (clear) – Port C operates normally.

STRCH — Stretch external accesses (refer to Section 3)

1 (set) – Off-chip accesses are extended by one E clock cycle.

0 (clear) – Normal operation.

IRVNE — Internal read visibility/not E (refer to Section 3)

1 (set) – Data from internal reads is driven out of the external data bus.

0 (clear) – No visibility of internal reads on external bus.

In **single chip mode** this bit determines whether the E clock drives out from the chip.

1 (set) – E pin is driven low.

0 (clear) – E clock is driven out from the chip.

LSBF — LSB first enable

1 (set) – SPI1 data is transferred LSB first.

0 (clear) – SPI1 data is transferred MSB first.

If this bit is set, data, which is usually transferred MSB first, is transferred LSB first. LSBF does not affect the position of the MSB and LSB in the data register. Reads and writes of the data register always have MSB in bit 7.

SPR2 — SPI clock rate select

When set, SPR2 adds a divide-by-4 prescaler to the SPI clock chain. With the two bits in the SPCR, this bit specifies the SPI clock rate. Refer to Table 7-1.

EXT4X — 4XCLK or EXTAL clock output select (refer to Section 3)

1 (set) – EXTALi clock output on the 4XOUT pin.

0 (clear) – 4XCLK clock output on the 4XOUT pin.

DISE — E clock output disable (refer to Section 3)

1 (set) – No E clock output.

0 (clear) – E clock is output normally.

7.6 SPI2

In addition to the subsystem described in the above paragraphs (SPI1), the MC68HC11PH8 has another SPI module (SPI2). This system is identical to SPI1, with the following exceptions:

- SPI2 shares I/O with four port G pins:

Pin	Alternate function
PG2	MISO2
PG3	MOSI2
PG4	SCK2
PG5	$\overline{\text{SS2}}$

- SPI1 functions and data are handled by a register block at \$0028–\$002A along with the system configuration options register 2 at address \$0038. The corresponding registers for SPI2 are at addresses \$004C–\$004E along with the SPI2 control options register at address \$004F. The SPI2 registers are described in the following sections.

7.6.1 SP2CR — SPI2 control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI2 control (SP2CR)	$004C	SP2IE	SP2E	GWOM	MSTR2	CPOL2	CPHA2	SP2R1	SP2R0	0000 01uu

For details of the functions of bits 2,3,4,6 and 7, see Section 7.5.1.

GWOM — Port G wired-OR mode

1 (set) – Port G [5:2] buffers configured for open-drain outputs.

0 (clear) – Port G [5:2] buffers configured for normal CMOS outputs.

SP2R1 and SP2R0 — SPI2 clock rate selects

These two bits, along with the SP2R2 bit, select the SPI clock rate as shown in Table 7-1. Note that SP2R2 is located in the SP2OPT register, and that its state on reset is zero.

7.6.2 SP2SR — SPI2 status register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI2 status (SP2SR)	$004D	SP2IF	WCOL2	0	MODF2	0	0	0	0	0000 0000

For a description of bits 4,6 and 7, see Section 7.5.2.

7.6.3 SP2DR — SPI2 data register

	Address	bit 7	Bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI2 data (SP2DR)	$004E	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined

For a description of this register, see Section 7.5.3.

7.6.4 SP2OPT — SPI2 control options register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
SPI2 control options (SP2DR)	$004F	0	0	0	0	LSBF2	SP2R2	0	0	0000 0000

For a description of bits 2 and 3, see Section 7.5.4.

THIS PAGE INTENTIONALLY LEFT BLANK

8
TIMING SYSTEM

The MC68HC11PH8 has three timing modules: a 16-bit timer system (incorporating pulse accumulator, RTI and COP), a pulse width modulation (PWM) system and an 8-bit modulus timing system comprising timers A, B and C.

8.1 16-bit timer

The M68HC11 timing system is composed of several clock divider chains. The main clock divider chain includes a 16-bit free-running counter, which is driven by a programmable prescaler. The main timer's programmable prescaler provides one of the four clocking rates to drive the 16-bit counter. Two prescaler control bits select the prescale rate. The prescaler output divides the system clock by 1, 4, 8, or 16. Taps from this main clocking chain drive circuitry may be used to generate the slower clocks used by the pulse accumulator, the real-time interrupt (RTI), the computer operating properly (COP) watchdog subsystems and the LCD module. Refer to Figure 8-1 and Figure 8-2.

All main timer system activities can be referenced to the free-running counter. The counter begins incrementing from $0000 as the MCU comes out of reset, and continues to the maximum count, $FFFF. At the maximum count, the counter rolls over to $0000, sets an overflow flag and continues to increment. As long as the MCU is running in a normal operating mode, there is no way to reset, change or interrupt the counting, unless, for reduced power consumption and if the PLL is in operation, the 16-bit counter is disabled under control of the T16EN bit (see Section 8.1.1.1). The capture/compare subsystem features three input capture channels, four output compare channels and one channel that can be selected to perform either input capture or output compare. Each of the input capture functions has its own 16-bit input capture register (time capture latch) and each of the output compare functions has its own 16-bit compare register. All timer functions, including the timer overflow and RTI, have their own interrupt controls and separate interrupt vectors. See Table 8-1 for related frequencies and periods.

The pulse accumulator contains an 8-bit counter and edge select logic. The pulse accumulator can operate in either event counting mode or gated time accumulation mode. During event counting mode, the pulse accumulator's 8-bit counter increments when a specified edge is detected on an input pin. During gated time accumulation mode, an internal clock source (ST4XCK/2^8) increments the 8-bit counter while an input signal has a predetermined logic level.

The real-time interrupt (RTI) is a programmable periodic interrupt circuit that permits pacing of the execution of software routines by selecting one of four interrupt rates. It may be clocked by the 16-bit timer (ST4XCK/2^{15}) or by the underflow of 8-bit modulus timer A (CLK64), depending on whether or not the PLL system is active (see Figure 8-1, Figure 8-2 and Section 8.1.5).

The COP watchdog clock input may be tapped off from the free-running counter chain (ST4XCK/2^{17}), or may be the underflow of the 8-bit modulus timer A (CLK64/4), depending on whether or not the PLL system is active (see Figure 8-1, Figure 8-2 and Section 8.1.6). The COP automatically times out unless it is serviced within a specific time by a program reset sequence. If the COP is allowed to time out, a reset is generated, which drives the $\overline{\text{RESET}}$ pin low to reset the MCU and the external system (see Section 10).

The LCD module can drive up to four LCD segments, and may be clocked by the 16-bit timer (ST4XCK/2^{18}), or by the underflow of the 8-bit modulus timer A (CLK64 or CLK64/8), depending on whether or not the PLL system is active (see Figure 8-1, Figure 8-2 and Section 2).

Table 8-1 Timer resolution and capacity

	Clock					
Control bits PR[1:0]	4.0MHz 1.0MHz 1000ns	8.0MHz 2.0MHz 500ns	12.0MHz 3.0MHz 333ns	16.0MHz 4.0MHz 250ns	ST4XCK ST4XCK/4 4/ST4XCK	Crystal[1] Clock Period
0 0	1.0µs 65.536ms	500ns 32.768ms	333ns 21.845ms	250ns 16.384ms	4/ST4XCK 2^{18}/ST4XCK	– resolution – overflow
0 1	4.0µs 262.14ms	2.0µs 131.07ms	1.333µs 87.381ms	1.0µs 65.536ms	16/ST4XCK 2^{20}/ST4XCK	– resolution – overflow
1 0	8.0µs 524.29ms	4.0µs 262.14ms	2.667µs 174.76ms	2.0µs 131.07ms	32/ST4XCK 2^{21}/ST4XCK	– resolution – overflow
1 1	16.0µs 1049 ms	8.0µs 524.29ms	5.333µs 349.53ms	4.0µs 262.14ms	64/ST4XCK 2^{22}/ST4XCK	– resolution – overflow

[1] Crystal frequencies are valid only if the PLL is not active.

8.1.1 Timer enable control

The 16-bit timer may be enabled or disabled under control of the T16EN bit in the PLL control register.

8.1.1.1 PLLCR — PLL control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
PLL control (PLLCR)	$002E	PLLON	BCS	AUTO	BWC	VCOT	MCS	T16EN	WEN	1010 1010

PLLON — PLL on (See Section 2.5.4.1)

1 (set) – Switch PLL on.

0 (clear) – Switch PLL off.

BCS — Bus clock select (See Section 2.5.4.1)

1 (set) – VCOOUT output drives the clock circuit (4XCLK).

0 (clear) – EXTALi drives the clock circuit (4XCLK).

AUTO — Automatic bandwidth control (See Section 2.5.4.1)

1 (set) – Automatic bandwidth control selected.

0 (clear) – Manual bandwidth control selected.

BWC — Bandwidth control (See Section 2.5.4.1)

1 (set) – High bandwidth control selected.

0 (clear) – Low bandwidth control selected.

VCOT — VCO test (Test mode only, see Section 2.5.4.1)

1 (set) – Loop filter operates as specified by AUTO and BWC.

0 (clear) – Low bandwidth mode of the PLL filter is disabled.

MCS — Module clock select (See Section 2.5.4.1)

1 (set) – 4XCLK is the source for the SCI and timer divider chain.

0 (clear) – EXTALi is the source for the SCI and timer divider chain.

T16EN — 16-bit timer clock enable

1 (set) – 16-bit timer clock enabled.

0 (clear) – 16-bit timer clock disabled.

Power consumption may be reduced by disabling the 16-bit timer clock. This bit cannot be cleared whilst VDDSYN is low, as then the 16-bit timer provides the clock source for the COP and RTI. When VDDSYN is high, the 8-bit modulus timer A supplies the clock source for the COP and RTI functions, which are therefore independent from the 16-bit timer clock. Reset sets this bit.

WEN — WAIT enable (See Section 2.5.4.1)

1 (set) – Low-power WAIT mode selected (PLL set to 'idle' in WAIT mode).

0 (clear) – Do not alter the 4XCLK during WAIT mode.

8.1.2 Timer structure

The timer functions share I/O with all eight pins of port A:

Pin	Alternate function
PA0	IC3
PA1	IC2
PA2	IC1
PA3	OC5 and/or OC1, or IC4
PA4	OC4 and/or OC1
PA5	OC3 and/or OC1
PA6	OC2 and/or OC1
PA7	PAI and/or OC1

Figure 8-3 shows the capture/compare system block diagram. The port A pin control block includes logic for timer functions and for general-purpose I/O. For pins PA3, PA2, PA1 and PA0, this block contains both the edge-detection logic and the control logic that enables the selection of which edge triggers an input capture. The digital level on PA[3:0] can be read at any time (read PORTA register), even if the pin is being used for the input capture function. Pins PA[6:3] are used either for general-purpose I/O, or as output compare pins. When one of these pins is being used for an output compare function, it cannot be written directly as if it were a general-purpose output. Each of the output compare functions (OC[5:2]) is related to one of the port A output pins. Output compare 1 (OC1) has extra control logic, allowing it optional control of any combination of the PA[7:3] pins. The PA7 pin can be used as a general-purpose I/O pin, as an input to the pulse accumulator or as an OC1 output pin.

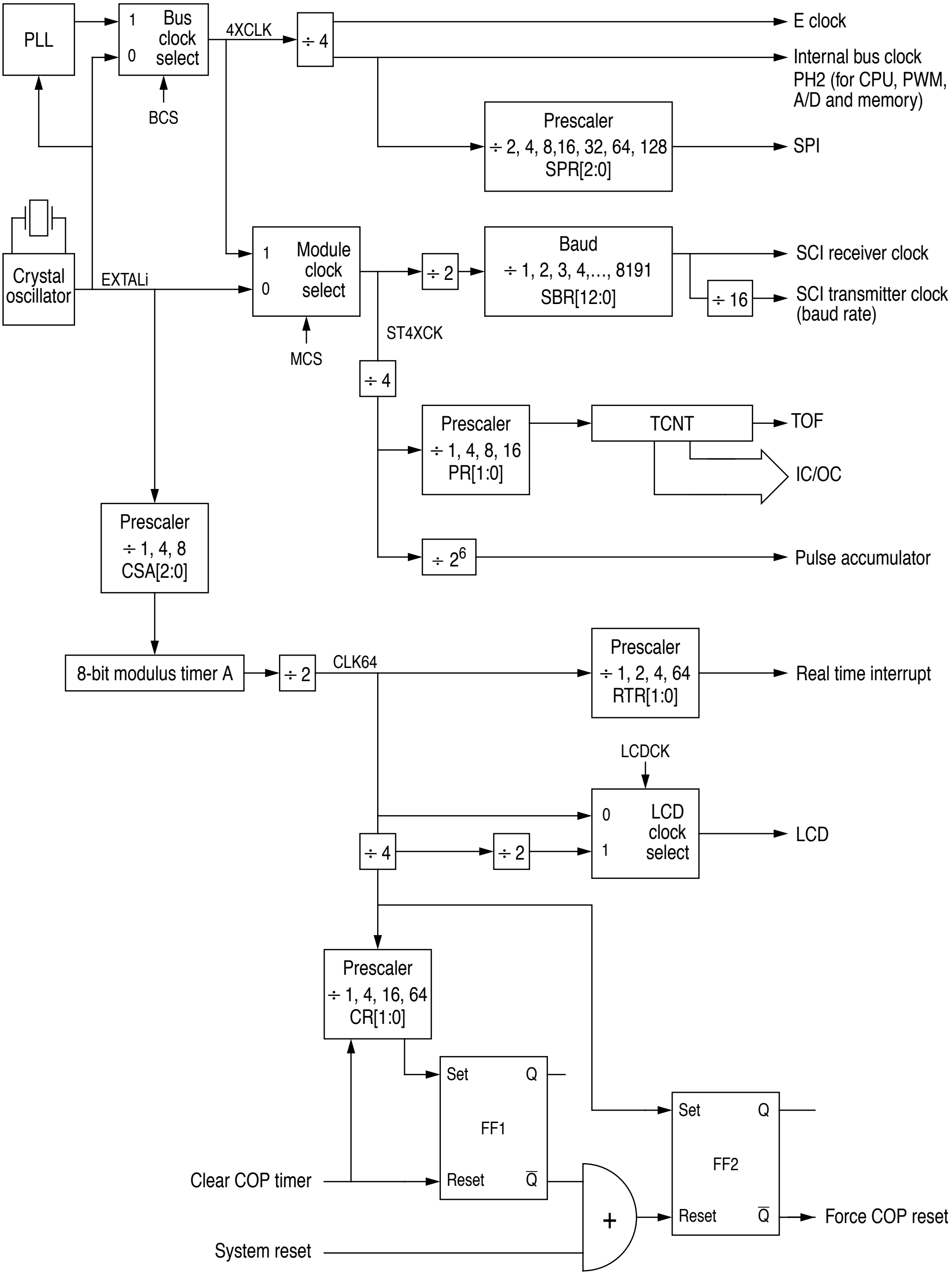


Figure 8-1 Timer clock divider chains (PLL enabled — VDDSYN high)

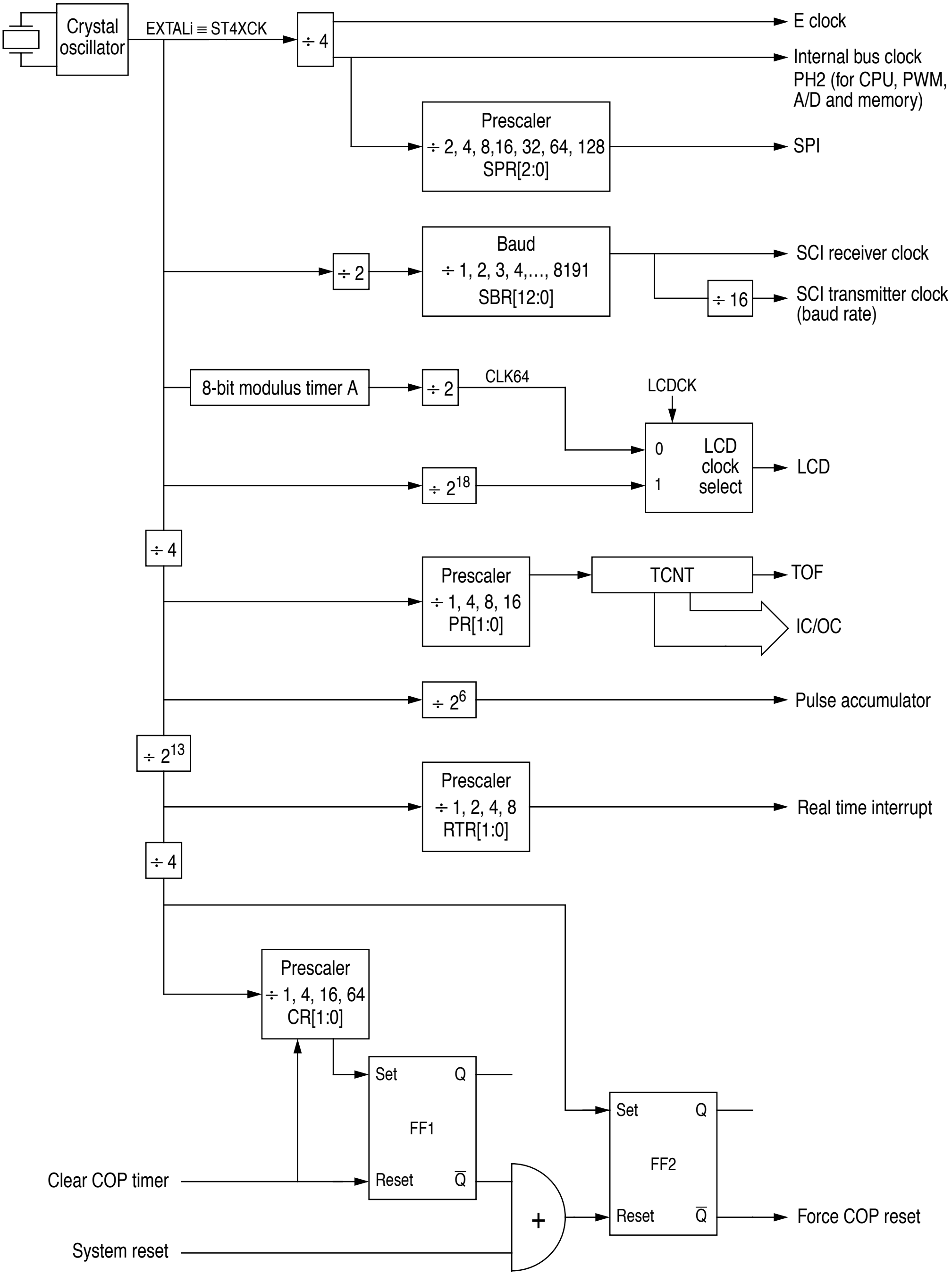
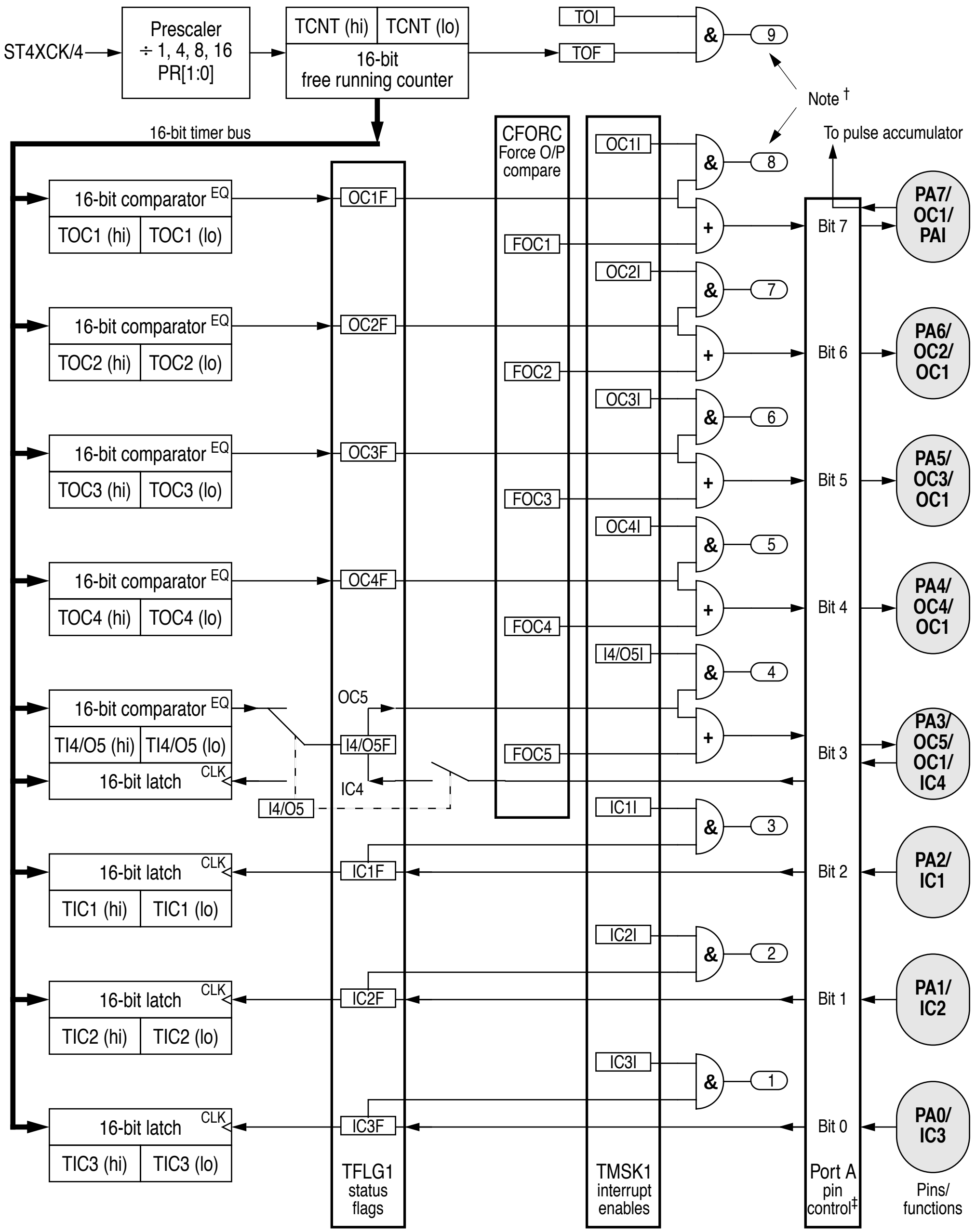


Figure 8-2 Timer clock divider chains (PLL disabled — VDDSYN low)



† Interrupt requests 1–9 (these are further qualified by the I-bit in the CCR)

‡ Port A pin actions are controlled by OC1M, OC1D, PACTL, TCTL1 and TCTL2 registers

Figure 8-3 Capture/compare block diagram

8.1.3 Input capture

The input capture function records the time an external event occurs by latching the value of the free-running counter when a selected edge is detected at the associated timer input pin. Software can store latched values and use them to compute the periodicity and duration of events. For example, by storing the times of successive edges of an incoming signal, software can determine the period and pulse width of a signal. To measure period, two successive edges of the same polarity are captured. To measure pulse width, two alternate polarity edges are captured.

In most cases, input capture edges are asynchronous with respect to the internal timer counter, which is clocked relative to an internal clock (PH2). These asynchronous capture requests are synchronized with PH2 so that latching occurs on the opposite half cycle of PH2 from when the timer counter is being incremented. This synchronization process introduces a delay from when the edge occurs to when the counter value is detected. Because these delays cancel out when the time between two edges is being measured, the delay can be ignored. When an input capture is being used with an output compare, there is a similar delay between the actual compare point and when the output pin changes state.

The control and status bits that implement the input capture functions are contained in the PACTL, TCTL2, TMSK1, and TFLG1 registers.

To configure port A bit 3 as an input capture, clear the DDA3 bit of the DDRA 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 DDA3 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.

8.1.3.1 TCTL2 — Timer control register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer control 2 (TCTL2)	$0021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	0000 0000

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.

EDGxB and EDGxA — Input capture edge control

EDGxB	EDGxA	Configuration
0	0	ICx disabled
0	1	ICx captures on rising edges only
1	0	ICx captures on falling edges only
1	1	ICx captures on any edge

There are four pairs of these bits. Each pair is cleared 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.

8.1.3.2 TIC1–TIC3 — Timer input capture registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer input capture 1 (TIC1) high	$0010	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
Timer input capture 1 (TIC1) low	$0011	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
Timer input capture 2 (TIC2) high	$0012	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
Timer input capture 2 (TIC2) low	$0013	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
Timer input capture 3 (TIC3) high	$0014	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	undefined
Timer input capture 3 (TIC3) low	$0015	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined

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. Input capture values can be read from a pair of 8-bit read-only registers. A read of the high-order byte of an input capture register pair inhibits a new capture transfer for one bus cycle. If a double-byte read instruction, such as LDD, is used to read the captured value, coherency is assured. When a new input capture occurs immediately after a high-order byte read, transfer is delayed for an additional cycle but the value is not lost.

The TICx registers are not affected by reset.

8.1.3.3 TI4/O5 — Timer input capture 4/output compare 5 register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Capture 4/compare 5 (TI4/O5) high	$001E	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Capture 4/compare 5 (TI4/O5) low	$001F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111

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 one. To use it as an output compare register, set the I4/O5 bit to a logic level zero. Refer to Section 8.1.8.1.

The TI4/O5 register pair resets to ones ($FFFF).

8.1.4 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 latency. To generate an output signal of a specific frequency and duty cycle, repeat this pulse-generating procedure.

There are four 16-bit read/write output compare registers: TOC1, TOC2, TOC3, and TOC4, and the TI4/O5 register, which 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.

OC1 is different from the other output compares in that a successful OC1 compare can affect any or all five of the OC pins. The OC1 output action taken when a match is found is controlled by two 8-bit registers with three bits unimplemented: the output compare 1 mask register, OC1M, and the output compare 1 data register, OC1D. OC1M specifies which port A outputs are to be used, and OC1D specifies what data is placed on these port pins.

8.1.4.1 TOC1–TOC4 — Timer output compare registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer output compare 1 (TOC1) high	$0016	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 1 (TOC1) low	$0017	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 2 (TOC2) high	$0018	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 2 (TOC2) low	$0019	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 3 (TOC3) high	$001A	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 3 (TOC3) low	$001B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Timer output compare 4 (TOC4) high	$001C	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	1111 1111
Timer output compare 4 (TOC4) low	$001D	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111

All output compare registers are 16-bit read-write. Each is initialized to $FFFF at reset. If an output compare register is not used for an output compare function, it can be used as a storage location. A write to the high-order byte of an output compare register pair inhibits the output compare function for one bus cycle. This inhibition prevents inappropriate subsequent comparisons. Coherency requires a complete 16-bit read or write. However, if coherency is not needed, byte accesses can be used.

For output compare functions, write a comparison value to output compare registers TOC1–TOC4 and TI4/O5. When TCNT value matches the comparison value, specified pin actions occur.

All TOCx register pairs reset to ones ($FFFF).

8.1.4.2 CFORC — Timer compare force register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer compare force (CFORC)	$000B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	0000 0000

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.

FOC[1:5] — Force output compares

1 (set) – A forced output compare action will occur on the specified pin.

0 (clear) – No action.

Bits [2:0] — Not implemented; always read zero

8.1.4.3 OC1M — Output compare 1 mask register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Output compare 1 mask (OC1M)	$000C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	0000 0000

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 PA7–PA3.

OC1M[7:3] — Output compare masks for OC1

1 (set) – OC1 is configured to control the corresponding pin of port A.

0 (clear) – OC1 will not affect the corresponding port A pin.

Bits [2:0] — Not implemented; always read zero.

8.1.4.4 OC1D — Output compare 1 data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Output compare 1 data (OC1D)	$000D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	0000 0000

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

OC1D[7:3] — Output compare data for OC1

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

Bits [2:0] — Not implemented; always read zero

8.1.4.5 TCNT — Timer counter register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer count (TCNT) high	$000E	(bit 15)	(14)	(13)	(12)	(11)	(10)	(9)	(bit 8)	0000 0000
Timer count (TCNT) low	$000F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000

The 16-bit read-only TCNT register contains the prescaled value of the 16-bit timer. A full counter read addresses the more significant byte (MSB) first. A read of this address causes the less significant byte (LSB) to be latched into a buffer for the next CPU cycle so that a double-byte read returns the full 16-bit state of the counter at the time of the MSB read cycle.

TCNT resets to $0000.

8.1.4.6 TCTL1 — Timer control register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer control 1 (TCTL1)	$0020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	0000 0000

The bits of this register specify the action taken as a result of a successful OCx compare.

OM[2:5] — Output mode
OL[2:5] — Output level

OMx	OLx	Action taken on successful compare
0	0	Timer disconnected from OCx pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to 0
1	1	Set OCx output line to 1

These control bit pairs are encoded to specify the action taken after a successful OCx compare. OC5 functions only if the I4/O5 bit in the PACTL register is clear.

8.1.4.7 TMSK1 — Timer interrupt mask register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt mask 1 (TMSK1)	$0022	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I	0000 0000

Use this 8-bit register to enable or inhibit the timer input capture and output compare interrupts.

Note: Bits in TMSK1 correspond bit for bit with flag bits in TFLG1. Ones in TMSK1 enable the corresponding interrupt sources.

OC1I–OC4I — Output compare x interrupt enable

1 (set) – OCx interrupt is enabled.

0 (clear) – OCx interrupt is disabled.

If the OCxI enable bit is set when the OCxF flag bit is set, a hardware interrupt sequence is requested.

I4/O5I — Input capture 4/output compare 5 interrupt enable

1 (set) – IC4/OC5 interrupt is enabled.

0 (clear) – IC4/OC5 interrupt is disabled.

When I4/O5 in PACTL is set, I4/O5I is the input capture 4 interrupt enable bit.
When I4/O5 in PACTL is zero, I4/O5I is the output compare 5 interrupt enable bit.

IC1I–IC3I — Input capture x interrupt enable

1 (set) – ICx interrupt is enabled.

0 (clear) – ICx interrupt is disabled.

If the ICxI enable bit is set when the ICxF flag bit is set, a hardware interrupt sequence is requested.

8.1.4.8 TFLG1 — Timer interrupt flag register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt flag 1 (TFLG1)	$0023	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F	0000 0000

Bits in this register indicate when timer system events have occurred. Coupled with the bits of TMSK1, the bits of TFLG1 allow the timer subsystem to operate in either a polled or interrupt driven system. Clear flags by writing a one to the corresponding bit position(s).

Note: Bits in TFLG1 correspond bit for bit with flag bits in TMSK1. Ones in TMSK1 enable the corresponding interrupt sources.

OC1F–OC4F — Output compare x flag

1 (set) – Counter has reached the preset output compare x value.

0 (clear) – Counter has not reached the preset output compare x value.

These flags are set each time the counter matches the corresponding output compare x values.

I4/O5F — Input capture 4/output compare 5 flag

Set by IC4 or OC5, depending on the function enabled by I4/O5 bit in PACTL

IC1F–IC3F — Input capture x flag

1 (set) – Selected edge has been detected on corresponding port pin.

0 (clear) – Selected edge has not been detected on corresponding port pin.

These flags are set each time a selected active edge is detected on the ICx input line

8.1.4.9 TMSK2 — Timer interrupt mask register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt mask 2 (TMSK2)	$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	0000 0000

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.

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

TOI — Timer overflow interrupt enable

1 (set) – Timer overflow interrupt requested when TOF is set.

0 (clear) – TOF interrupts disabled.

RTII — Real-time interrupt enable (refer to Section 8.1.5)

1 (set) – Real time interrupt requested when RTIF is set.

0 (clear) – Real time interrupts disabled.

PAOVI — Pulse accumulator overflow interrupt enable (refer to Section 8.1.8)

PAII — Pulse accumulator input edge interrupt enable (refer to Section 8.1.8)

Bits [3, 2] — Not implemented; always read zero.

PR[1:0] — Timer prescaler select

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

These bits are used to select the prescaler divide-by ratio. In normal modes, PR[1:0] can only be written once, and the write must be within 64 cycles after reset. See Table 8-1 for specific timing values.

8.1.4.10 TFLG2 — Timer interrupt flag register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt flag 2 (TFLG2)	$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	0000 0000

Bits in this register indicate when certain timer system events have occurred. 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. Clear flags by writing a one to the corresponding bit position(s).

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

TOF — Timer overflow interrupt flag

1 (set) – TCNT has overflowed from $FFFF to $0000.

0 (clear) – No timer overflow has occurred.

RTIF — Real time (periodic) interrupt flag (refer to Section 8.1.5)

1 (set) – RTI period has elapsed.

0 (clear) – RTI flag has been cleared.

PAOVF — Pulse accumulator overflow interrupt flag (refer to Section 8.1.8)

PAIF — Pulse accumulator input edge interrupt flag (refer to Section 8.1.8.)

Bits [3:0] — Not implemented; always read zero

8.1.5 Real-time interrupt

The real-time interrupt (RTI) feature, used to generate hardware interrupts at a fixed periodic rate, has two possible clock sources. When the PLL clock generation is not used (VDDSYN low), the RTI function is clocked by the 16-bit free-running counter (ST4XCK/2^{15}). When the PLL clock generation is used (VDDSYN high), the RTI clock source is the underflow of the 8-bit modulus timer A (CLK64). This ensures that the RTI interrupt rate is unaffected by changes made to the bus speed by the PLL circuit. See Figure 8-1 and Figure 8-2. The RTI clock rate is controlled and configured by two bits (RTR1 and RTR0) in the pulse accumulator control (PACTL) register. The different rates available are a product of the source frequency and the value of bits RTR[1:0]. If VDDSYN is low, the source frequency, ST4XCK/2^{15}, can be divided by 1,2,4 or 8. If VDDSYN is high, the source frequency, CLK64, can be divided by 1,2,4 or 64. Refer to Table 8-2 and Table 8-3 which show examples of periodic real-time interrupt rates. The RTII bit in the TMSK2 register enables the interrupt capability.

Table 8-2 RTI periodic rates (PLL disabled)

RTR[1:0]	ST4XCK = 12MHz	ST4XCK = 8MHz	ST4XCK = 4MHz	ST4XCK = xMHz
0 0	2.731 ms	4.096 ms	8.192 ms	2^{15}/ST4XCK
0 1	5.461 ms	8.192 ms	16.384 ms	2^{16}/ST4XCK
1 0	10.923 ms	16.384 ms	32.768 ms	2^{17}/ST4XCK
1 1	21.845 ms	32.768 ms	65.536 ms	2^{18}/ST4XCK

Table 8-3 RTI periodic rates (PLL enabled)

RTR[1:0]	EXTALi = 640kHz	EXTALi = 32.768kHz	EXTALi = 32kHz	EXTALi = xkHz
0 0	0.4 ms	7.81 ms	8.0 ms	2^{8}/EXTALi
0 1	0.8 ms	15.63 ms	16.0 ms	2^{9}/EXTALi
1 0	1.6 ms	31.25 ms	32.0 ms	2^{10}/EXTALi
1 1	25.6 ms	500 ms	512 ms	2^{14}/EXTALi

Note: The values in Table 8-3 assume that the 8-bit modulus timer is loaded to give an EXTALi/2^{8} prescaler value. Other prescaler values are possible, in the range EXTALi/4 to EXTALi/4080 (see Section 8.3.1).

Either clock source causes the time between successive RTI timeouts to be a constant that is independent of the software latency associated with flag clearing and service. For this reason, an RTI period starts from the previous timeout, not from when RTIF is cleared.

Every timeout causes the RTIF bit in TFLG2 to be set, and if RTII is set, an interrupt request is generated. After reset, one entire RTI period elapses before the RTIF flag is set for the first time. Refer to the TMSK2, TFLG2, and PACTL registers.

8.1.5.1 TMSK2 — Timer interrupt mask register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt mask 2 (TMSK2)	$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	0000 0000

This register contains the real-time interrupt enable bit.

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

TOI — Timer overflow interrupt enable (refer to Section 8.1.4.9)

1 (set) – Timer overflow interrupt requested when TOF is set.

0 (clear) – TOF interrupts disabled.

RTII — Real-time interrupt enable

1 (set) – Real time interrupt requested when RTIF is set.

0 (clear) – Real time interrupts disabled.

PAOVI — Pulse accumulator overflow interrupt enable (refer to Section 8.1.8)

PAII — Pulse accumulator input edge (refer to Section 8.1.8)

Bits[3, 2] — Not implemented; always reads zero

PR[1, 0] — Timer prescaler select (refer to Section 8.1.4.9)

8.1.5.2 TFLG2 — Timer interrupt flag register 2

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt flag 2 (TFLG2)	$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	0000 0000

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

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

TOF — Timer overflow interrupt flag (refer to Section 8.1.4.10)

1 (set) – TCNT has overflowed from $FFFF to $0000.

0 (clear) – No timer overflow has occurred.

RTIF — Real-time interrupt flag

1 (set) – RTI period has elapsed.

0 (clear) – RTI flag has been cleared.

The RTIF status bit is automatically set to one at the end of every RTI period.

PAOVF — Pulse accumulator overflow interrupt flag (refer to Section 8.1.8)

PAIF — Pulse accumulator input edge interrupt flag (refer to Section 8.1.8)

Bits [3:0] — Not implemented; always read zero

8.1.5.3 PACTL — Pulse accumulator control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse accumulator control (PACTL)	$0026	0	PAEN	PAMOD	PEDGE	0	I4/O5	RTR1	RTR0	0000 0000

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.

Bits [7, 3] — Not implemented; always read zero

PAEN — Pulse accumulator system enable (refer to Section 8.1.8)

1 (set) – Pulse accumulator enabled.

0 (clear) – Pulse accumulator disabled.

PAMOD — Pulse accumulator mode (refer to Section 8.1.8)

1 (set) – Gated time accumulation mode.

0 (clear) – Event counter mode.

PEDGE — Pulse accumulator edge control (refer to Section 8.1.8)

This bit has different meanings depending on the state of the PAMOD bit.

I4/O5 — Input capture 4/output compare 5 (refer to Section 8.1.8)

1 (set) – Input capture 4 function is enabled (no OC5).

0 (clear) – Output compare 5 function is enabled (no IC4).

RTR[1:0] — RTI interrupt rate select

These two bits determine the rate at which the RTI system requests interrupts. The RTI system is driven either by CLK64 or by an ST4XCK/2^{15} clock rate that is compensated so it is independent of the timer prescaler. These two control bits select an additional division factor. Refer to Table 8-2 and Table 8-3.

8.1.6 Computer operating properly watchdog function

There are two possible clock sources for the COP function (see Figure 8-1 and Figure 8-2). When PLL clock generation is not used (VDDSYN low), the clocking chain for the COP function is tapped off from the main timer divider chain (ST4XCK/2^{17}). When the PLL clock generation is used (VDDSYN high), the COP function is clocked by the underflow of the 8-bit modulus timer A (CLK64/4). The CR[1:0] bits in the OPTION register and the NOCOP bit in the CONFIG register control and configure the COP function. One additional register, COPRST, is used to arm and clear the COP watchdog reset system. Refer to Section 10 for a more detailed discussion of the COP function.

8.1.7 LCD module

There are three possible clock sources for the LCD module, under control of the LCDCK bit and depending on the state of VDDSYN. When LCDCK = 0, the LCD module is clocked by the output of 8-bit modulus timer A (CLK64). When LCDCK = 1, the LCD module is clocked by CLK64/8 if PLL clock generation is used (VDDSYN high), and by ST4XCK/2^{18} if PLL clock generation is not used (VDDSYN low). Refer to Figure 8-1, Figure 8-2 and Section 2.12.

8.1.8 Pulse accumulator

The MC68HC11PH8 has an 8-bit counter that can be configured to operate either as a simple event counter, or for gated time accumulation, depending on the state of the PAMOD bit in the PACTL register. Refer to the pulse accumulator block diagram, Figure 8-4.

In the event counting mode, the 8-bit counter is clocked to increasing values by an external pin. The maximum clocking rate for the external event counting mode is the E clock divided by two. In gated time accumulation mode, a free-running ST4XCK/2^8 signal drives the 8-bit counter, but only while the external PAI pin is activated. Refer to Table 8-4. The pulse accumulator counter can be read or written at any time.

Table 8-4 Pulse accumulator timing

Crystal frequency[1]	ST4XCK/4 clock	Cycle time	2^8/ST4XCK	PACNT overflow
4.0 MHz	1.0 MHz	1000 ns	64 µs	16.384 ms
8.0 MHz	2.0 MHz	500 ns	32 µs	8.192 ms
12.0 MHz	3.0 MHz	333 ns	21.33 µs	5.461 ms
16.0 MHz	4.0 MHz	250 ns	16.0 µs	4.096 ms

[1] Crystal frequency values are only valid if the PLL is not active.

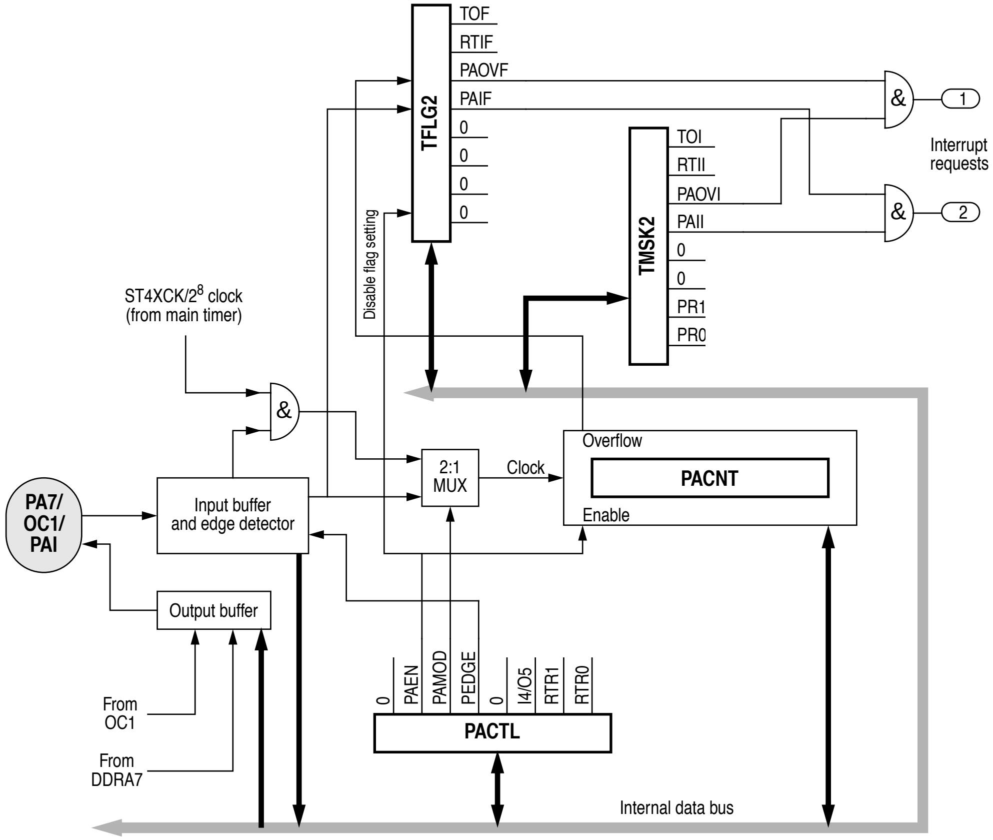


Figure 8-4 Pulse accumulator block diagram

Pulse accumulator control bits are located within the PACTL, TMSK2 and TFLG2 registers, as described in the following paragraphs.

8.1.8.1 PACTL — Pulse accumulator control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse accumulator control (PACTL)	$0026	0	PAEN	PAMOD	PEDGE	0	I4/O5	RTR1	RTR0	0000 0000

Four of this register's bits control an 8-bit pulse accumulator system. Another bit enables either the OC5 function or the IC4 function, while two other bits select the rate for the real-time interrupt system.

Bits [7, 3] — Not implemented; always read zero

PAEN — Pulse accumulator system enable

1 (set) – Pulse accumulator enabled.

0 (clear) – Pulse accumulator disabled.

PAMOD — Pulse accumulator mode

1 (set) – Gated time accumulation mode.

0 (clear) – Event counter mode.

PEDGE — Pulse accumulator edge control

This bit has different meanings depending on the state of the PAMOD bit, as shown:

PAMOD	PEDGE	Action of clock
0	0	PAI falling edge increments the counter.
0	1	PAI rising edge increments the counter.
1	0	A zero on PAI inhibits counting.
1	1	A one on PAI inhibits counting.

I4/O5 — Input capture 4/output compare 5

1 (set) – Input capture 4 function is enabled (no OC5).

0 (clear) – Output compare 5 function is enabled (no IC4)

RTR[1:0] — RTI interrupt rate selects (refer to Section 8.1.5)

8.1.8.2 PACNT — Pulse accumulator count register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse accumulator count (PACNT)	$0027	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined

This 8-bit read/write register contains the count of external input events at the PAI input, or the accumulated count. In gated time accumulation mode, PACNT is readable even if PAI is not active. 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.

8.1.8.3 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.

8.1.8.4 TMSK2 — Timer interrupt mask 2 register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt mask 2 (TMSK2)	$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	0000 0000

8.1.8.5 TFLG2 — Timer interrupt flag 2 register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Timer interrupt flag 2 (TFLG2)	$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	0000 0000

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 one in the corresponding data bit position (bit 5) of the TFLG2 register. The PAOVI control bit allows the pulse accumulator overflow to be configured for polled or interrupt-driven operation and does not affect the state of PAOVF. When PAOVI is zero, 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.

PAII and PAIF — Pulse accumulator input edge interrupt enable and flag

The PAIF status bit is automatically set each time a selected edge is detected at the PA7/PAI/OC1 pin. To clear this status bit, write to the TFLG2 register with a one in the corresponding data bit position (bit 4). The PAII control bit allows the pulse accumulator input edge detect to be configured for polled or interrupt-driven operation but does not affect setting or clearing the PAIF bit. When PAII is zero, pulse accumulator input interrupts are inhibited, and the system operates in a polled mode. In this mode, the PAIF bit must be polled by user software to determine when an edge has occurred. When the PAII control bit is set, a hardware interrupt request is generated each time PAIF is set. Before leaving the interrupt service routine, software must clear PAIF.

8.2 Pulse-width modulation (PWM) timer

The PWM timer subsystem provides up to four 8-bit pulse-width modulated waveforms on the port H pins. Channel pairs can be concatenated to create 16-bit PWM outputs. Three clock sources (A, B, and S) and a flexible clock select scheme give the PWM a wide range of frequencies.

Pin	Alternate function
PH0	PW1
PH1	PW2
PH2	PW3
PH3	PW4

Four control registers configure the PWM outputs — PWCLK, PWPOL, PWSCAL, and PWEN. The PWCLK register selects the prescale value for the PWM clock sources and enables the 16-bit PWM functions. The PWPOL register determines each channel’s polarity and selects the clock source for each channel. The PWSCAL register derives a user-scaled clock based on the A clock source, and the PWEN register enables the PWM channels.

Each channel also has a separate 8-bit counter, period register, and duty cycle register. The period and duty cycle registers are double buffered so that if they are changed while the channel is enabled, the change does not take effect until the counter rolls over or the channel is disabled. A new period or duty cycle can be forced into effect immediately by writing to the period or duty cycle register and then writing to the counter.

With PWMs configured for 8-bit mode and E equal to 4MHz, PWM signals can be produced from 40 kHz (1% duty cycle resolution) to less than 10 cycles per second (approximately 0.4% duty cycle resolution). By configuring the PWMs for 16-bit mode with E equal to 4MHz, PWM periods greater than one minute are possible.

In 16-bit mode, duty cycle resolution of up to 15 parts per million can be achieved (at a PWM frequency of 60Hz). In the same system, a PWM frequency of 1kHz corresponds to a duty cycle resolution of 0.025%.

8.2.1 PWM timer block diagram

Figure 8-5 shows the block diagram of the PWM timer subsystem. Three different clock sources are selectable and provide inputs to the control registers. Each of the four channels has a counter, a period register, and a duty register. The waveform output is the result of a match between the period register (PWPERx) and the value in the counter (PWCNTx). The duty register (PWDTYx) changes the state of the output during the period to determine the duty cycle.

8.2.2 PWCLK — PWM clock prescaler and 16-bit select register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width clock select (PWCLK)	$0060	CON34	CON12	PCKA2	PCKA1	0	PCKB3	PCKB2	PCKB1	0000 0000

This register contains bits for selecting the 16-bit PWM options and the prescaler values for the clocks.

8.2.2.1 16-bit PWM function

The PWCLK register contains two control bits, each of which is used to concatenate a pair of PWM channels into one 16-bit channel. Channels 3 and 4 are concatenated with the CON34 bit, and channels 1 and 2 are concatenated with the CON12 bit.

When the 16-bit concatenated mode is selected, the clock source is determined by the low order channel. Channel 2 is the low order channel when channels 1 and 2 are concatenated. Channel 4 is the low order channel when channels 3 and 4 are concatenated. The pins associated with channels 1 and 3 can be used for general-purpose I/O when 16-bit PWM mode is selected.

Channel 1 registers are the high order byte of the double-byte channel when channels 1 and 2 are concatenated. Channel 3 registers are the high order byte of the double-byte channel when channels 3 and 4 are concatenated. Reads of the high order byte cause the low order byte to be latched for one cycle to guarantee that double byte reads are accurate. Writes to the low byte of the counter cause reset of the entire counter. Writes to the upper bytes of the counter have no effect.

CON34 — Concatenate channels 3 and 4

1 (set) – Channels 3 and 4 are concatenated into one 16-bit PWM channel.

0 (clear) – Channels 3 and 4 are separate 8-bit PWMs.

When concatenated, channel 3 is the high-order byte and the channel 4 pin (PH3) is the output.

CON12 — Concatenate channels 1 and 2

1 (set) – Channels 1 and 2 are concatenated into one 16-bit PWM channel.

0 (clear) – Channels 1 and 2 are separate 8-bit PWMs.

When concatenated, channel 1 is the high-order byte and the channel 2 pin (PH1) is the output.

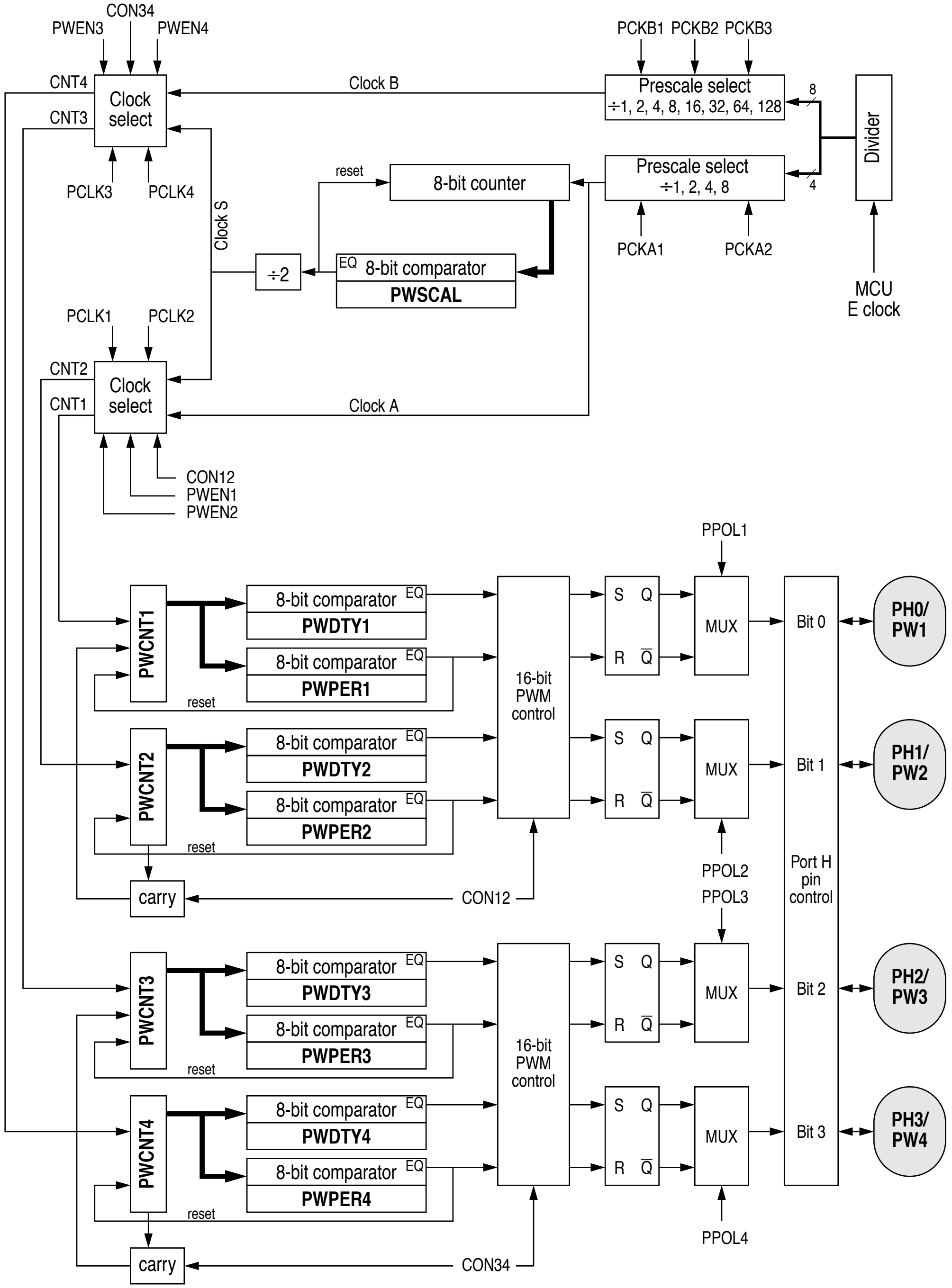


Figure 8-5 PWM timer block diagram

8.2.2.2 Clock prescaler selection

The three available clocks are clock A, clock B, and clock S (scaled). Clock A can be software selected to be E, E/2, E/4, or E/8. Clock B can be software selected to be E, E/2, E/4,..., E/128. The scaled clock (clock S) uses clock A as an input and divides it with a reloadable counter. The rates available are software selectable to be clock A/2, down to clock A /512.

The clock source portion of the block diagram shows the three clock sources and how the scaled clock is created. Clock A is an input to an 8-bit counter which is then compared to a user programmable scale value. When they match, this circuit has an output that is divided by two and the counter is reset.

Each PWM timer channel can be driven by one of two clocks. Refer to Figure 8-5.

PCKA[2:1] — Prescaler for clock A

Determines the frequency of clock A. Refer to Table 8-5.

Bit 3 — Not implemented; always reads zero

PCKB[3:1] — Prescaler for clock B

Determines the frequency of clock B. Refer to Table 8-5.

Table 8-5 Clock A and clock B prescalers

PCKA[2:1]	Clock A
0 0	E
0 1	E/2
1 0	E/4
1 1	E/8

PCKB[3:1]	Clock B
0 0 0	E
0 0 1	E/2
0 1 0	E/4
0 1 1	E/8
1 0 0	E/16
1 0 1	E/32
1 1 0	E/64
1 1 1	E/128

8.2.3 PWPOL — PWM timer polarity & clock source select register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width polarity select (PWPOL)	$0061	PCLK4	PCLK3	PCLK2	PCLK1	PPOL4	PPOL3	PPOL2	PPOL1	0000 0000

PCLK[4:3] — Pulse width channel 4/3 clock select

1 (set) – Clock S is source.

0 (clear) – Clock B is source.

PCLK[2:1] — Pulse width channel 2/1 clock select

1 (set) – Clock S is source.

0 (clear) – Clock A is source.

PPOL[4:1] — Pulse width channel x polarity

1 (set) – PWM channel x output is high at the beginning of the clock cycle and goes low when duty count is reached.

0 (clear) – PWM channel x output is low at the beginning of the clock cycle and goes high when duty count is reached.

Each channel has a polarity bit that allows a cycle to start with either a high or a low level. This is shown on the block diagram, Figure 8-5, as a selection of either the Q output or the $\overline{Q}$ output of the PWM output flip flop. When one of the bits in the PWPOL register is set, the associated PWM channel output is high at the beginning of the clock cycle, then goes low when the duty count is reached.

8.2.4 PWSCAL — PWM timer prescaler register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width scale (PWSCAL)	$0062	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000

Scaled clock S is generated by dividing clock A by the value in PWSCAL, then dividing the result by two. If PWSCAL = $00, clock A is divided by 256, then divided by two to generate clock S.

8.2.5 PWEN — PWM timer enable register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width enable (PWEN)	$0063	TPWSL	DISCP	0	0	PWEN4	PWEN3	PWEN2	PWEN1	0000 0000

Each timer has an enable bit to start its waveform output. Writing any of these PWENx bits to one causes the associated port line to become an output regardless of the state of the associated DDR bit. This does not change the state of the DDR bit and when PWENx returns to zero the DDR bit again controls I/O state. On the front end of the PWM timer the clock is connected to the PWM circuit by the PWENx enable bit being high. There is a synchronizing circuit to guarantee that the clock will only be enabled or disabled at an edge.

PWEN contains 4 PWM enable bits — one for each channel. When an enable bit is set to one, the pulse modulated signal becomes available at the associated port pin.

TPWSL — PWM scaled clock test bit (Test mode only)

1 (set) – Clock S output to PWSCAL register (Test only).

0 (clear) – Normal operation.

When TPWSL is one, clock S from the PWM timer is output to PWSCAL register. Normal writing to the PWSCAL register still functions.

DISCP — Disable compare scaled E clock (Test mode only)

1 (set) – Match of period does not reset associated count register (Test only).

0 (clear) – Normal operation.

Bits [5:4] — Not implemented; always read zero

PWEN[4:1] — Pulse width channels 4–1

1 (set) – Channel enabled on the associated port pin.

0 (clear) – Channel disabled.

8.2.6 PWCNT1–4 — PWM timer counter registers 1 to 4

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width count 1 (PWCNT1)	$0064	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Pulse width count 2 (PWCNT2)	$0065	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Pulse width count 3 (PWCNT3)	$0066	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000
Pulse width count 4 (PWCNT4)	$0067	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	0000 0000

Each channel has its own counter which can be read at any time without affecting the count or the operation of the PWM channel. Writing to a counter causes it to be reset to $00; this is generally done before the counter is enabled. A counter may also be written to whilst it is enabled; this may cause a truncated PWM period.

8.2.7 PWPER1–4 — PWM timer period registers 1 to 4

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width period 1 (PWPER1)	$0068	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width period 2 (PWPER2)	$0069	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width period 3 (PWPER3)	$006A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width period 4 (PWPER4)	$006B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111

There is one period register for each channel. The value in this register determines the period of the associated PWM timer channel. PWPERx is connected internally to a buffer which compares directly with the counter register. The period value in PWPERx is loaded into the buffer when the counter is cleared by the termination of the previous period or by a write to the counter. This register can be written at any time, and the written value will take effect from the start of the next PWM timer cycle. Reads of this register return the most recent value written.

8.2.8 PWDTY1–4 — PWM timer duty cycle registers 1 to 4

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Pulse width duty 1 (PWDTY1)	$006C	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width duty 2 (PWDTY2)	$006D	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width duty 3 (PWDTY3)	$006E	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111
Pulse width duty 4 (PWDTY4)	$006F	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	1111 1111

There is one duty register for each channel. The value in this register determines the duty cycle of the associated PWM timer channel. PWDTYx is compared to the counter contents and if they are equal, a match occurs and the output goes to the state defined by the associated polarity bit. If the register is written while the channel is enabled, then the new value is held in a buffer until the counter rolls over or the channel is disabled. Reads of this register return the most recent value written.

Note: If PWDTYx ≥ PWPERx then there will be no change of state due to the duty cycle value. In addition, if the duty register is set to $00, then the output will always be in the state which would normally be result from the duty change of state (see also Section 8.2.9).

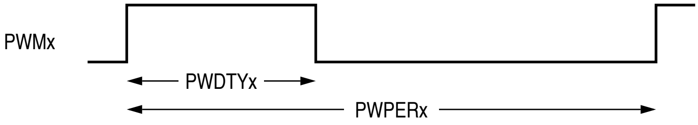


Figure 8-6 PWM duty cycle

8.2.9 Boundary cases

The following boundary conditions apply to the values stored in the PWDTYx and PWPERx registers and the PPOLx bits:

- If PWDTYx = $00, PWPERx > $00 and PPOLx = 0 then the output is always high.
- If PWDTYx = $00, PWPERx > $00 and PPOLx = 1 then the output is always low.
- If PWDTYx ≥ PWPERx and PPOLx = 0 then the output is always low.
- If PWDTYx ≥ PWPERx and PPOLx = 1 then the output is always high.
- If PWPERx = $00 and PPOLx = 0 then the output is always low.
- If PWPERx = $00 and PPOLx = 1 then the output is always high.

8.3 8-bit modulus timers

The MC68HC11PH8 has three 8-bit modulus timers: A, B and C. These timers can generate a wide range of low-frequency, periodic interrupts. In addition, timer A may be used as a clock source for the LCD segments, the COP watchdog system and the real time interrupt (RTI), in applications using PLL clock generation.

Each modulus timer consists of an 8-bit down-counter, an 8-bit modulus register and a load mechanism. Only the 8-bit modulus register may be written to in software. Each timer is configured by an associated control register that enables and flags interrupts, and selects the clock source for the down-counter. Figure 8-7 provides a block diagram of the modulus timers.

8.3.1 Modulus timer operation

The down-counter in each timer contains a value which is decremented at a preselected clock rate. When this counter value reaches $00 (‘underflow’), an output pulse is generated and, if enabled, a hardware interrupt is requested. At this point, a new value is loaded into the down-counter from the modulus register; at the next clock, the counter will contain this value minus one.

Note: For all three timers, modulus register values of $00 or $01 should be avoided. Because modulus timer A is used to clock the COP monitor and cannot be stopped, the loading mechanism on modulus timer A is inhibited for values of $00 or $01; at underflow, the counter will roll over from $00 to $FF and continue decrementing.

The frequency that is output from the timer is equal to the clock frequency divided by the value in the 8-bit modulus register; a modulus register value of n generates a modulus timer underflow every n input clocks. Therefore, the modulus timer can divide an input frequency by any value from 2 to 255. In addition, the timer A clock output is further divided by two to give the CLK64 signal. There are several software-selectable input clocks for the modulus timers (see Section 8.3.2 and Figure 8-7). For example, the modulus timer A clock source can be EXTALi, EXTALi/4 or EXTALi/8. Consequently, CLK64 can vary in frequency from EXTALi/4 (EXTALi ÷ 2 ÷ 2) to EXTALi/4080 (EXTALi/8 ÷ 255 ÷ 2). The following table provides an example of how to obtain a 64 Hz frequency from various EXTALi values, using timer A.

EXTALi	Modulus timer A		
	Input clock	Modulus register	Output
32 kHz	EXTALi	$FA	64 Hz
32.768 kHz	EXTALi/4	$40	64 Hz
38.4 kHz	EXTALi/4	$46	64 Hz

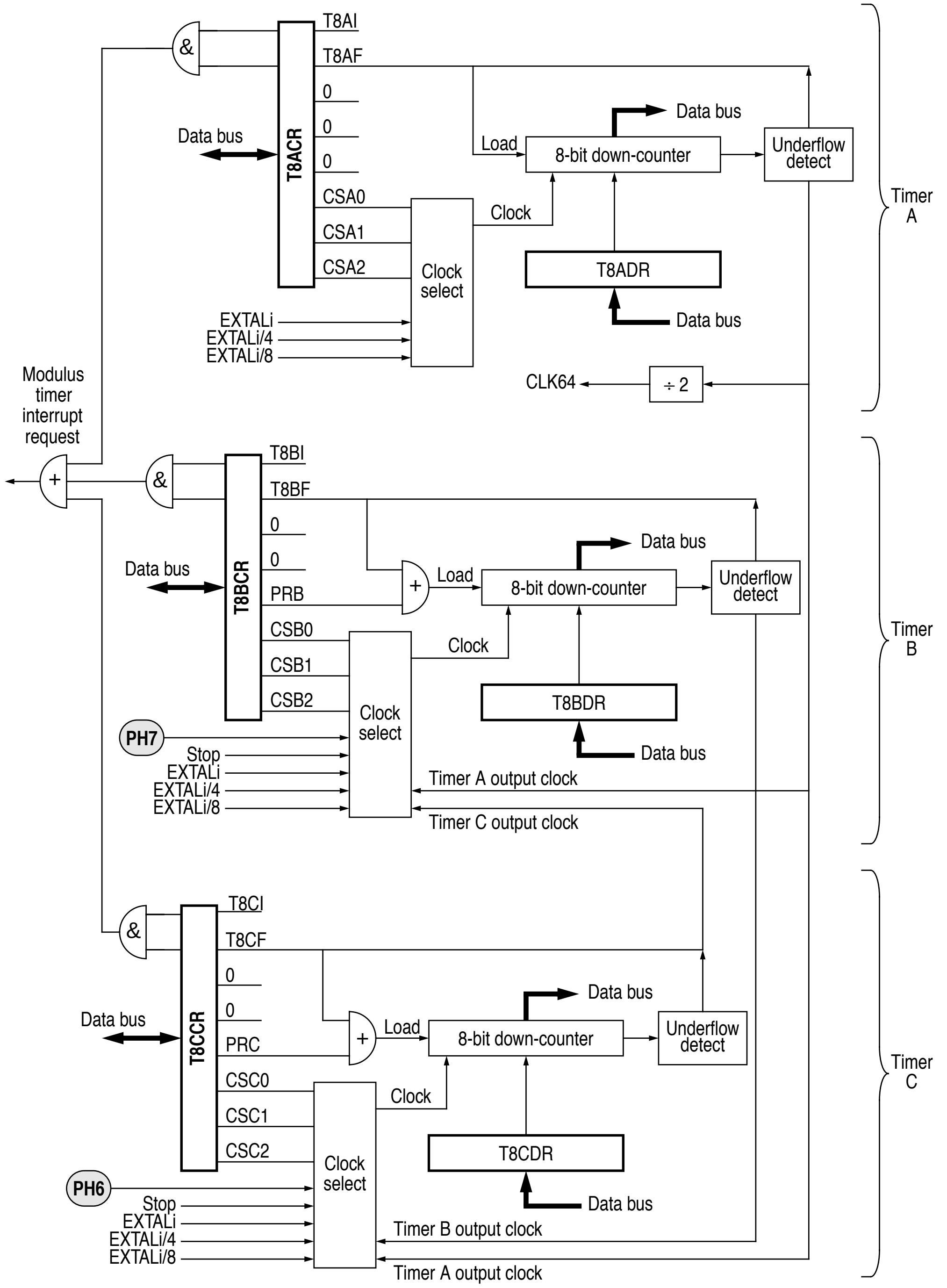


Figure 8-7 8-bit modulus timer system

The modulus register can be written to at any time without affecting the down-counter; care must be taken by the programmer to ensure that a new value is written to the modulus register before the down-counter reaches $00, unless the previous value is to be reloaded. A read of the modulus register with the timer running will access a latched value of the down-counter. This latch guarantees a stable value during the read and is updated with each timer clock. To determine the value in the modulus register itself (for timers B and C), the timer must be stopped by the appropriate clock selection (see Section 8.3.2), then a one must be written to the preset bit in the control register. The down-counter now contains the modulus register value.

Note: Because it is used to clock the COP watchdog in applications using PLL clock generation, timer A cannot be stopped (therefore, the value in modulus register A cannot be determined).

Note: The timer preset bits only have an effect if the timer is stopped under hardware control (timers B and C only).

The recommended procedure for configuring timers B and C is as follows:

1) Stop the timer by writing %000 to the relevant clock select control bits.
2) Set the modulus register by writing the required value to the timer data register.
3) Write a one to the timer preset bit.
4) Select the desired clock source to start the counter decrementing.

8.3.2 Clock rate selection

A number of clock rates can be selected in software for each of the three modulus timers. The selection is controlled by bits 0-2 in the timer control register, as shown in the tables below. Timer A may be used as a prescaler for timers B and C (see Figure 8-7). Similarly timer B may be used as a prescaler for timer C, and vice versa.

Table 8-6 Modulus timers clock sources

CSA[2:0]	Timer A clock source
0 0 0	EXTALi/8
0 0 1	EXTALi
0 1 0	EXTALi/4
0 1 1	EXTALi/8
1 0 0	EXTALi/8
1 0 1	EXTALi/8
1 1 0	EXTALi/8
1 1 1	EXTALi/8

CSB[2:0]	Timer B clock source
0 0 0	Stopped
0 0 1	EXTALi
0 1 0	EXTALi/4
0 1 1	EXTALi/8
1 0 0	Timer A underflow
1 0 1	Timer C underflow
1 1 0	Rising edge PH7
1 1 1	Stopped

CSC[2:0]	Timer C clock source
0 0 0	Stopped
0 0 1	EXTALi
0 1 0	EXTALi /4
0 1 1	EXTALi /8
1 0 0	Timer A underflow
1 0 1	Timer B underflow
1 1 0	Rising edge PH6
1 1 1	Stopped

Warning: Selecting EXTALi as a clock source when the PLL function is not used, i.e. when the bus frequency is EXTALi/4, could lead to read or write errors in the timer registers.

8.3.2.1 T8ADR — 8-bit modulus timer A data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
8-bit modulus timer A data (T8ADR)	$0059	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	1111 1111

This 8-bit register contains the value that will be loaded into the timer A down-counter on the next underflow. At reset, the timer A clock source is the EXTALi clock divided by 8, and the modulus register is initialized to its highest value.

Because timer A is used to clock the COP watchdog in applications using the PLL clock generation, it is not possible to stop timer A. For the same reason, a write of values $00 or $01 to this register will not be loaded from the modulus register to the counter.

8.3.2.2 T8ACR — 8-bit modulus timer A control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
8-bit modulus timer A control (T8ACR)	$005D	T8AI	T8AF	0	0	0	CSA2	CSA1	CSA0	0000 0000

T8AI — 8-bit timer A interrupt enable

1 (set) – Hardware interrupt requested when T8AF flag set.

0 (clear) – Interrupt disabled.

When set, an 8-bit modulus timer interrupt occurs when the timer reaches $00. At this time the timer counter is loaded with the value stored in T8ADR and the 8-bit counter will continue to count down at the selected clock rate.

T8AF — 8-bit timer A underflow flag

Set when 8-bit modulus timer A reaches $00. An interrupt is generated if enabled by T8AI. This bit is cleared by a write to the T8ACR register with T8AF **set**.

Bits [5:3] — Not implemented; always read zero

CSA[2:0] — 8-bit timer A clock rate

These bits select the timer A clock, as shown in Table 8-6.

8.3.2.3 T8BDR — 8-bit modulus timer B data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
8-bit modulus timer B data (T8BDR)	$005A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	undefined

This 8-bit register contains the value that will be loaded into the timer B down-counter at the next underflow. At reset, timer B is stopped and the state of the modulus register is indeterminate.

8.3.2.4 T8BCR — 8-bit modulus timer B control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
8-bit modulus timer B control (T8BCR)	$005E	T8BI	T8BF	0	0	PRB	CSB2	CSB1	CSB0	0000 0000

T8BI — 8-bit timer B interrupt enable

1 (set) – Hardware interrupt requested when T8BF flag set.

0 (clear) – Interrupt disabled.

When set, an 8-bit modulus timer interrupt occurs when the timer reaches $00. At this time the timer counter is loaded with the value stored in T8BDR and the 8-bit counter will continue to count down at the selected clock rate.

T8BF — 8-bit timer B underflow flag

1 (set) – Underflow has occurred.

0 (clear) – No underflow has occurred.

Set when 8-bit modulus timer B reaches $00. An interrupt is generated if enabled by T8BI. This bit is cleared by a write to the T8BCR register with T8BF **set**.

Bits [5, 4] — Not implemented; always read zero

PRB — 8-bit timer B preset

A write to the T8BCR register with this bit set will preset the timer B counter to the modulus register value. The clock must be stopped before writing to the register. This bit always reads as 0.

CSB[2:0] — 8-bit timer B clock rate

These bits select the timer B clock, as shown in Table 8-6. At reset, timer B is not clocked.

8.3.2.5 T8CDR — 8-bit modulus timer C data register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
8-bit modulus timer C data (T8CDR)	$005B	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	undefined

This 8-bit register contains the value that will be loaded into the timer C down-counter at the next underflow. At reset, timer C is stopped and state of the modulus register is indeterminate.

8.3.2.6 T8CCR — 8-bit modulus timer C control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
8-bit modulus timer C control (T8CCR)	$005F	T8CI	T8CF	0	0	PRC	CSC2	CSC1	CSC0	0000 0000

T8CI8 — bit timer C interrupt enable

1 (set) – Hardware interrupt requested when T8CF flag set.

0 (clear) – Interrupt disabled.

When set, an 8-bit modulus timer interrupt occurs when the timer reaches $00. At this time the timer counter is loaded with the value stored in T8CDR and the 8-bit counter will continue to count down at the selected clock rate.

T8CF — 8-bit timer C underflow flag

1 (set) – Underflow has occurred.

0 (clear) – No underflow has occurred.

Set when 8-bit modulus timer C reaches $00. An interrupt is generated if enabled by T8CI. This bit is cleared by a write to the T8CCR register with T8CF **set**.

Bits [5, 4] — Not implemented; always read zero

PRC — 8-bit timer C preset

A write to the T8CCR register with this bit set will preset the timer C counter to the modulus register value. The clock must be stopped before writing to the register. This bit always reads as 0.

CSC[2:0] — 8-bit timer C clock rate

These bits select the timer C clock, as shown in Table 8-6. At reset, timer C is not clocked.

9
ANALOG-TO-DIGITAL CONVERTER

The analog-to-digital (A/D) system, a successive approximation converter, uses an all-capacitive charge redistribution technique to convert analog signals to digital values.

The A/D converter shares input pins with port E:

Pin	Alternate function
PE0	AD0
PE1	AD1
PE2	AD2
PE3	AD3
PE4	AD4
PE5	AD5
PE6	AD6
PE7	AD7

9.1 Overview

The A/D system is an 8-channel, 8-bit, multiplexed-input converter. The VDD AD and VSS AD pins are used to input supply voltage to the A/D converter. This allows the supply voltage to be bypassed independently. The converter does not require external sample and hold circuits because of the type of charge redistribution technique used. A/D converter timing can be synchronized to the system E clock, or to an internal resistor capacitor (RC) oscillator. The A/D converter system consists of four functional blocks: multiplexer, analog converter, digital control and result storage. Refer to Figure 9-1.

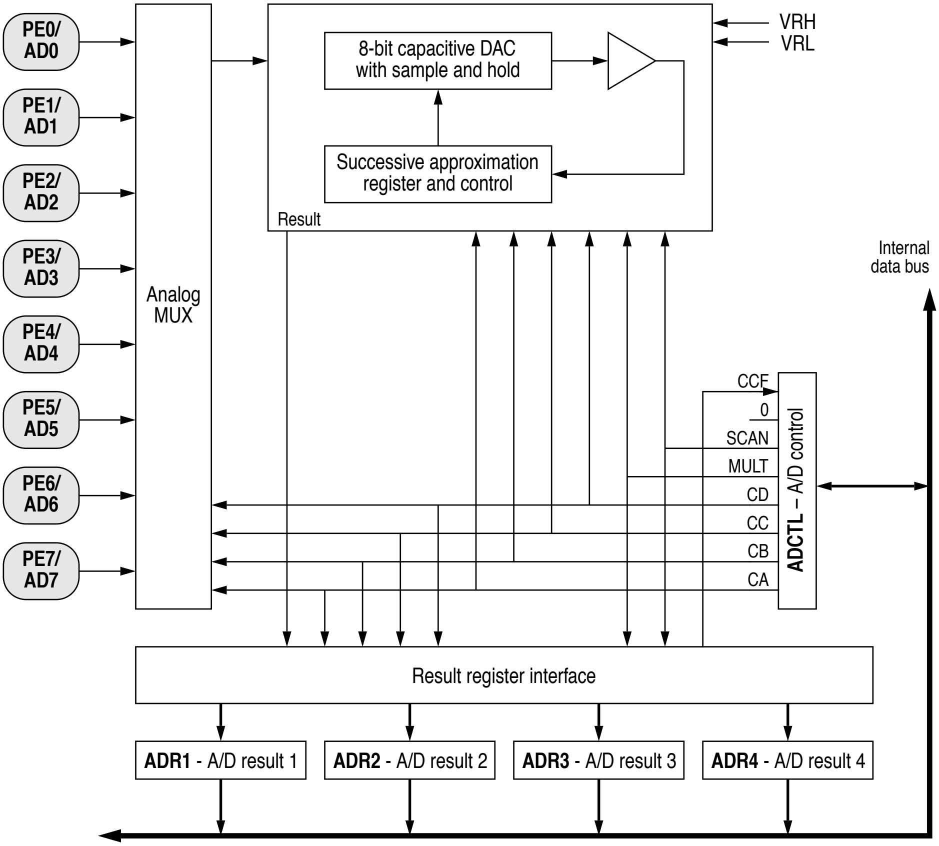
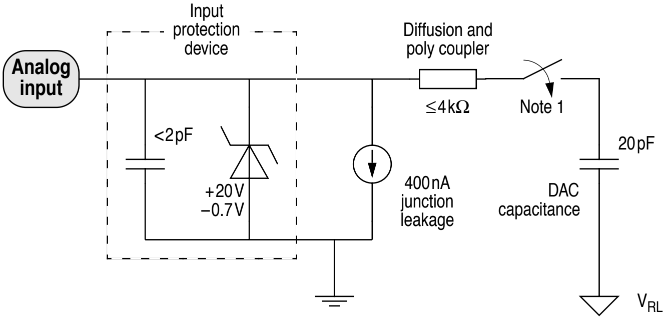


Figure 9-1 A/D converter block diagram

9.1.1 Multiplexer

The multiplexer selects one of 16 inputs for conversion. Input selection is controlled by the value of bits CD – CA in the ADCTL register. The eight port E pins are fixed-direction analog inputs to the multiplexer, and additional internal analog signal lines are routed to it.

Port E pins can also be used as digital inputs (see Section 4). Digital reads of port E pins should be avoided during the sample portion of an A/D conversion cycle, when the gate signal to the N-channel input gate is on. Because no P-channel devices are directly connected to either input pins or reference voltage pins, voltages above V_{DD} do not cause a latchup problem, although current and voltage should be limited according to maximum ratings. Refer to Figure 9-2, which is a functional diagram of an input pin.



Note 1: The analog switch is closed only during the 12 cycle sample time
Note 2: All component values are approximate

Figure 9-2 Electrical model of an A/D input pin (in sample mode)

9.1.2 Analog converter

Conversion of an analog input selected by the multiplexer occurs in this block. It contains a digital-to-analog capacitor (DAC) array, a comparator, and a successive approximation register (SAR). Each conversion is a sequence of eight comparison operations, beginning with the most significant bit (MSB). Each comparison determines the value of a bit in the SAR.

The DAC array performs two functions. It acts as a sample and hold circuit during the entire conversion sequence, and provides comparison voltage to the comparator during each successive comparison.

The result of each successive comparison is stored in the SAR. When a conversion sequence is complete, the contents of the SAR are transferred to the appropriate result register.

A charge pump provides switching voltage to the gates of analog switches in the multiplexer. Charge pump output must stabilize between 7 and 8 volts within up to 100 µs before the converter can be used. The charge pump is enabled by the ADPU bit in the OPTION register.

9.1.3 Digital control

All A/D converter operations are controlled by bits in register ADCTL. In addition to selecting the analog input to be converted, ADCTL bits indicate conversion status, and control whether single or continuous conversions are performed. Finally, the ADCTL bits determine whether conversions are performed on single or multiple channels.

9.1.4 Result registers

Four 8-bit registers (ADR1 – ADR4) store conversion results. Each of these registers can be accessed by the processor in the CPU. The conversion complete flag (CCF) indicates when valid data is present in the result registers. The result registers are written during a portion of the system clock cycle when reads do not occur, so there is no conflict.

9.1.5 A/D converter clocks

The CSEL bit in the OPTION register selects whether the A/D converter uses the system E clock or an internal RC oscillator for synchronization. When E clock frequency is below 750kHz, charge leakage in the capacitor array can cause errors, and the internal oscillator should be used. When the RC clock is used, additional errors can occur because the comparator is sensitive to the additional system clock noise.

9.1.6 Conversion sequence

A/D converter operations are performed in sequences of four conversions each. A conversion sequence can repeat continuously or stop after one iteration. The conversion complete flag (CCF) is set after the fourth conversion in a sequence to show the availability of data in the result registers. Figure 9-3 shows the timing of a typical sequence. Synchronization is referenced to the system E clock.

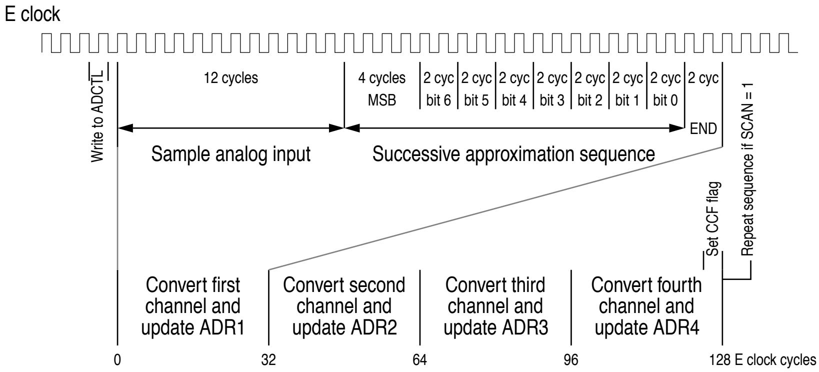

E clock
Write to ADCTL
12 cycles
4 cycles
MSB
2 cyc
bit 6
2 cyc
bit 5
2 cyc
bit 4
2 cyc
bit 3
2 cyc
bit 2
2 cyc
bit 1
2 cyc
bit 0
2 cyc
END
Sample analog input
Successive approximation sequence
Set CCF flag
Repeat sequence if SCAN = 1
Convert first channel and update ADR1
Convert second channel and update ADR2
Convert third channel and update ADR3
Convert fourth channel and update ADR4
0
32
64
96
128 E clock cycles

Figure 9-3 A/D conversion sequence

9.1.7 Conversion process

The A/D conversion sequence begins one E clock cycle after a write to the A/D control/status register, ADCTL. The bits in ADCTL select the channel and the mode of conversion.

An input voltage equal to V_{RL} converts to $00 and an input voltage equal to V_{RH} converts to $FF (full scale), with no overflow indication. For ratiometric conversions of this type, the source of each analog input should use V_{RH} as the supply voltage and be referenced to V_{RL}.

9.2 A/D converter power-up and clock select

ADPU (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. After the A/D converter is turned on, the analog bias voltages will take up to 100µs to stabilize.

When the A/D converter system is operating from the MCU E clock, all switching and comparator operations are synchronized to the MCU clocks. This allows the comparator results to be sampled at 'quiet' times, which minimizes noise errors. The internal RC oscillator is asynchronous with respect to the MCU clock, so noise can affect the A/D converter results. This results in a slightly lower typical accuracy when using the internal oscillator (CSEL = 1).

9.2.1 OPTION — System configuration options register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System config. options 1 (OPTION)	$0039	ADPU	CSEL	IRQE	DLY	CME	FCME	CR1	CR0	0001 0000

The 8-bit special-purpose OPTION register sets internal system configuration options during initialization. The time protected control bits, IRQE, DLY, FCME and CR[1:0] can be written to only once in the first 64 cycles after a reset and then they become read-only bits. This minimizes the possibility of any accidental changes to the system configuration. They may be written at any time in special modes.

ADPU — A/D power-up

1 (set) – A/D system power enabled.

0 (clear) – A/D system disabled, to reduce supply current.

After enabling the A/D power, at least 100µs should be allowed for system stabilization.

CSEL — Clock select

1 (set) – A/D, *EPROM* and EEPROM use internal RC clock source (about 1.5MHz).

0 (clear) – A/D, *EPROM* and EEPROM use system E clock (must be at least 1MHz).

Selects alternate clock source for on-chip *EPROM*, EEPROM and A/D charge pumps. The on-chip RC clock should be used when the E clock frequency falls below 1MHz.

IRQE — Configure $\overline{\textbf{IRQ}}$ for falling edge sensitive operation (refer to Section 3)

1 (set) – Falling edge sensitive operation.

0 (clear) – Low level sensitive operation.

DLY — Enable oscillator start-up delay (refer to Section 3)

1 (set) – A stabilization delay is imposed as the MCU is started up from STOP mode or from power-on reset.

0 (clear) – The oscillator start-up delay coming out of STOP is bypassed and the MCU resumes processing within about four bus cycles. A stable external oscillator is required if this option is selected.

CME — Clock monitor enable (refer to Section 10)

1 (set) – Clock monitor enabled.

0 (clear) – Clock monitor disabled.

FCME — Force clock monitor enable (refer to Section 10)

1 (set) – Clock monitor enabled, cannot be disabled until next reset.

0 (clear) – Clock monitor follows the state of the CME bit.

CR[1:0] — COP timer rate select bits (refer to Section 10)

9.3 Channel assignments

The multiplexer allows the A/D converter to select one of sixteen analog signals. Eight of these channels correspond to port E input lines to the MCU, four others are internal reference points or test functions; the remaining four channels are reserved. Refer to Table 9-1.

Table 9-1 A/D converter channel assignments

Channel number	Channel signal	Result in ADRx if MULT = 1
1	AD0	ADR1
2	AD1	ADR2
3	AD2	ADR3
4	AD3	ADR4
5	AD4	ADR1
6	AD5	ADR2
7	AD6	ADR3
8	AD7	ADR4
9–12	reserved	—
13	V_{RH}[1]	ADR1
14	V_{RL}[1]	ADR2
15	$V_{RH}/2$[1]	ADR3
16	reserved[1]	ADR4

[1] Used for factory testing.

9.3.1 Single-channel operation

There are two types of single-channel operation. In the first type (SCAN = 0), the single selected channel is converted four consecutive times. The first result is stored in A/D result register 1 (ADR1), and the fourth result is stored in ADR4. After the fourth conversion is complete, all conversion activity is halted until a new conversion command is written to the ADCTL register. In the second type of single-channel operation (SCAN = 1), conversions continue to be performed on the selected channel with the fifth conversion being stored in register ADR1 (overwriting the first conversion result), the sixth conversion overwriting ADR2, and so on.

9.3.2 Multiple-channel operation

There are two types of multiple-channel operation. In the first type (SCAN = 0), a selected group of four channels is converted once only. The first result is stored in A/D result register 1 (ADR1), and the fourth result is stored in ADR4. After the fourth conversion is complete, all conversion activity is halted until a new conversion command is written to the ADCTL register. In the second type of multiple-channel operation (SCAN = 1), conversions continue to be performed on the selected group of channels with the fifth conversion being stored in register ADR1 (replacing the earlier conversion result for the first channel in the group), the sixth conversion overwriting ADR2, and so on.

9.4 Control, status and results registers

9.4.1 ADCTL — A/D control and status register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
A/D control & status (ADCTL)	$0030	CCF	0	SCAN	MULT	CD	CC	CB	CA	u0uu uuuu

All bits in this register can be read or written, except bit 7, which is a read-only status indicator, and bit 6, which always reads as zero. Write to ADCTL to initiate a conversion. To quit a conversion in progress, write to this register and a new conversion sequence begins immediately.

CCF — Conversions complete flag

1 (set) – All four A/D result registers contain valid conversion data.

0 (clear) – At least one of the A/D result registers contains invalid data.

A read-only status indicator, this bit is set when all four A/D result registers contain valid conversion results. Each time the ADCTL register is overwritten, this bit is automatically cleared to zero and a conversion sequence is started. In the continuous mode, CCF is set at the end of the first conversion sequence.

Bit 6 — Not implemented; always reads zero.

SCAN — Continuous scan control

1 (set) – A/D conversions take place continuously.

0 (clear) – Each of the four conversions is performed only once.

When this control bit is clear, the four requested conversions are performed once to fill the four result registers. When this control bit is set, the four conversions are repeated continuously with the result registers updated as data becomes available.

MULT — Multiple-channel/single-channel control

1 (set) – Each A/D channel has a result register allocated to it.

0 (clear) – Four consecutive conversions from the same A/D channel are stored in the results registers.

When this bit is clear, the A/D converter system is configured to perform four consecutive conversions on the single channel specified by the four channel select bits CD–CA (bits 3–0 of the ADCTL register). When this bit is set, the A/D system is configured to perform a conversion on each of the four channels where each result register corresponds to one channel.

Note: When the multiple-channel continuous scan mode is used, extra care is needed in the design of circuitry driving the A/D inputs. The charge on the capacitive DAC array before the sample time is related to the voltage on the previously converted channel. A charge share situation exists between the internal DAC capacitance and the external circuit capacitance. Although the amount of charge involved is small, the rate at which it is repeated is every 64 µs for an E clock of 2 MHz. The RC charging rate of the external circuit must be balanced against this charge sharing effect to avoid errors in accuracy. Refer to the ***M68HC11 Reference Manual (M68HC11RM/AD)*** for further information.

CD–CA — Channel selects D–A

When a multiple channel mode is selected (MULT = 1), the two least significant channel select bits (CB and CA) have no meaning and the CD and CC bits specify which group of four channels is to be converted.

Channel select control bits	Channel signal	Result in ADRx if MULT = 1
CD:CC:CB:CA		
0 0 0 0	AD0	ADR1
0 0 0 1	AD1	ADR2
0 0 1 0	AD2	ADR3
0 0 1 1	AD3	ADR4
0 1 0 0	AD4	ADR1
0 1 0 1	AD5	ADR2
0 1 1 0	AD6	ADR3
0 1 1 1	AD7	ADR4
1 0 X X	reserved	—
1 1 0 0	V_{RH}(1)	ADR1
1 1 0 1	V_{RL}(1)	ADR2
1 1 1 0	$V_{RH}/2$(1)	ADR3
1 1 1 1	reserved(1)	ADR4

[1] Used for factory testing.

9.4.2 ADR1–ADR4 — A/D converter results registers

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
A/D result 1 (ADR1)	$0031	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
A/D result 2 (ADR2)	$0032	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
A/D result 3 (ADR3)	$0033	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined
A/D result 4 (ADR4)	$0034	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	undefined

These read-only registers hold an 8-bit conversion result. Writes to these registers have no effect. Data in the A/D converter result registers is valid when the CCF flag in the ADCTL register is set, indicating a conversion sequence is complete. If conversion results are needed sooner, refer to Figure 9-3, which shows the A/D conversion sequence diagram.

9.5 Operation in STOP and WAIT modes

If a conversion sequence is in progress when either the STOP or WAIT mode is entered, the conversion of the current channel is suspended. When the MCU resumes normal operation, that channel is resampled and the conversion sequence is resumed. As the MCU exits the WAIT mode, the A/D circuits are stable and valid results can be obtained on the first conversion. However, in STOP mode, all analog bias currents are disabled and it is necessary to allow a stabilization period when leaving the STOP mode. If the STOP mode is exited with a delay (DLY = 1), there is enough time for these circuits to stabilize before the first conversion. If the STOP mode is exited with no delay (DLY bit in OPTION register = 0), allow 10 ms for the A/D circuitry to stabilize to avoid invalid results.

10
RESETS AND INTERRUPTS

Resets and interrupt operations load the program counter with a vector that points to a new location from which instructions are to be fetched. A reset immediately stops execution of the current instruction and forces the program counter to a known starting address. Internal registers and control bits are initialized so that the MCU can resume executing instructions. An interrupt temporarily suspends normal program execution whilst an interrupt service routine is being executed. After an interrupt has been serviced, the main program resumes as if there had been no interruption.

10.1 Resets

There are four possible sources of reset. Power-on reset (POR) and external reset share the normal reset vector. The computer operating properly (COP) reset and the clock monitor reset each has its own vector.

10.1.1 Power-on reset

A positive transition on VDD generates a power-on reset (POR), which is used only for power-up conditions. POR cannot be used to detect drops in power supply voltages. A delay is imposed which allows the clock generator to stabilize after the oscillator becomes active. If $\overline{\text{RESET}}$ is at logical zero at the end of the delay time, the CPU remains in the reset condition until $\overline{\text{RESET}}$ goes to logical one. A mask option selects one of two delay times; either 128 or 4064 t_{CYC} (internal clock cycles).

Note: This mask option is not available on the MC68HC711PH8, where the delay time is 4064 t_{CYC}.

It is important to protect the MCU during power transitions. Most M68HC11 systems need an external circuit that holds the $\overline{\text{RESET}}$ pin low whenever V_{DD} is below the minimum operating level. This external voltage level detector, or other external reset circuits, are the usual source of reset in a system. The POR circuit only initializes internal circuitry during cold starts. Refer to Figure 2-3.

10.1.2 External reset ($\overline{\text{RESET}}$)

The CPU distinguishes between internal and external reset conditions by sensing whether the reset pin rises to a logic one in less than four E clock cycles after an internal device releases reset. When a reset condition is sensed, the $\overline{\text{RESET}}$ pin is driven low by an internal device for eight E clock cycles, then released. Four E clock cycles later it is sampled. If the pin is still held low, the CPU assumes that an external reset has occurred. If the pin is high, it indicates that the reset was initiated internally by either the COP system or the clock monitor. It is not advisable to connect an external resistor capacitor (RC) power-up delay circuit to the reset pin of M68HC11 devices because the circuit charge time constant can cause the device to misinterpret the type of reset that occurred. To guarantee recognition of an external reset, the $\overline{\text{RESET}}$ pin should be held low for at least 16 clock cycles.

10.1.3 COP reset

The MCU includes a COP system to help protect against software failures. When the COP is enabled, the software is responsible for keeping a free-running watchdog timer from timing out. When the software is no longer being executed in the intended sequence, a system reset is initiated.

The state of the NOCOP bit in the CONFIG register determines whether the COP system is enabled or disabled. To change the enable status of the COP system, change the contents of the CONFIG register and then perform a system reset. In the special test and bootstrap operating modes, the COP system is initially inhibited by the disable resets (DISR) control bit in the TEST1 register. The DISR bit can subsequently be written to zero to enable COP resets.

The COP function has two possible clock sources. When PLL clock generation is not used (VDDSYN = 0), the clocking chain for the COP function is tapped off from the main timer divider chain ($E/2^{15}$); refer to Figure 8-1. When the PLL clock generation is used (VDDSYN = 1), the COP function can be clocked by the underflow of the 8-bit modulus timer A (CLK64/4); see Figure 8-2. The COP timer rate control bits CR[1:0] in the OPTION register determine the COP timeout period. The COP clock source frequency is scaled by the factor shown in Table 10-1 or Table 10-2. After reset, bits CR[1:0] are zero, which selects the shortest timeout period. In normal operating modes, these bits can only be written once, within 64 bus cycles after reset.

Table 10-1 COP timer rate select (PLL disabled)

CR[1:0]	Divide $E/2^{15}$ by	EXTALi = 8 MHz: timeout[1]	EXTALi = 12MHz: timeout[1]	EXTALi = 16 MHz: timeout[1]
0 0	1	16.384 ms	10.923 ms	8.192 ms
0 1	4	65.536 ms	43.691 ms	32.768 ms
1 0	16	262.14 ms	174.76 ms	131.07 ms
1 1	64	1.049 sec	699.05 ms	524.29 ms
	E =	**2.0 MHz**	**3.0 MHz**	**4.0 MHz**

[1] The timeout period has a tolerance of –0/+one cycle of the $E/2^{15}$ clock due to the asynchronous implementation of the COP circuitry. For example, with EXTALi = 8 MHz, the uncertainty is –0/+16.384 ms. See also the ***M68HC11 Reference Manual, (M68HC11RM/AD)***.

Table 10-2 COP timer rate select (PLL enabled)

CR[1:0]	Divide CLK64 by	CLK64 = 4.096 kHz: timeout[1]	CLK64 =64 Hz: timeout[1]	CLK64 = 4 Hz: timeout[1]
0 0	4	1 ms	62.5 ms	1 s
0 1	16	3.9 ms	250 ms	4 s
1 0	64	15.6 ms	1 s	16 s
1 1	256	62.5 sec	4 s	64 s

[1] The timeout period has a tolerance of –0/+one cycle of the CLK64/4 clock due to the asynchronous implementation of the COP circuitry. For example, with CLK64 = 64 Hz, the uncertainty is –0/+62.5 ms. See also the ***M68HC11 Reference Manual, (M68HC11RM/AD)***.

10.1.3.1 COPRST — Arm/reset COP timer circuitry register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
COP timer arm/reset (COPRST)	$003A	(bit 7)	(6)	(5)	(4)	(3)	(2)	(1)	(bit 0)	not affected

Complete the following reset sequence to service the COP timer. Write $55 to COPRST to arm the COP timer clearing mechanism. Then write $AA to COPRST to clear the COP timer. Executing instructions between these two steps is possible as long as both steps are completed in the correct sequence before the timer times out.

10.1.4 Clock monitor reset

The clock monitor circuit is based on an internal RC time delay. If no MCU clock edges are detected within this RC time delay, the clock monitor can optionally generate a system reset. The clock monitor function is enabled or disabled by the CME control bit in the OPTION register. The presence of a timeout is determined by the RC delay, which allows the clock monitor to operate without any MCU clocks.

Clock monitor is used as a backup for the COP system. Because the COP needs a clock to function, it is disabled when the clocks stop. Therefore, the clock monitor system can detect clock failures not detected by the COP system.

Semiconductor wafer processing causes variations of the RC timeout values between individual devices. An E clock frequency below 10 kHz is detected as a clock monitor error. An E clock frequency of 200 kHz or more prevents clock monitor errors. Using the clock monitor function when the E clock is below 200 kHz is not recommended.

Special considerations are needed when a STOP instruction is executed and the clock monitor is enabled. Because the STOP function causes the clocks to be halted, the clock monitor function generates a reset sequence if it is enabled at the time the STOP mode was initiated. Before executing a STOP instruction, clear the CME bit in the OPTION register to zero to disable the clock monitor. After recovery from STOP, set the CME bit to logic one to enable the clock monitor.

10.1.5 OPTION — System configuration options register 1

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
System config. options 1 (OPTION)	$0039	ADPU	CSEL	IRQE	DLY	CME	FCME	CR1	CR0	0001 0000

The special-purpose OPTION register sets internal system configuration options during initialization. The time protected control bits (IRQE, DLY, FCME and CR[1:0]) can be written to only once in the first 64 cycles after a reset and then they become read-only bits. This minimizes the possibility of any accidental changes to the system configuration. They may be written at any time in special modes.

ADPU — A/D power-up (Refer to Section 9)

1 (set) – A/D system power enabled.

0 (clear) – A/D system disabled, to reduce supply current.

CSEL — Clock select (Refer to Section 9)

1 (set) – A/D, *EPROM* and EEPROM use internal RC clock (about 1.5MHz).

0 (clear) – A/D, *EPROM* and EEPROM use system E clock
(must be at least 1MHz).

IRQE — Configure $\overline{\text{IRQ}}$ for falling edge sensitive operation (Refer to Section 3)

1 (set) – Falling edge sensitive operation.

0 (clear) – Low level sensitive operation.

DLY — Enable oscillator start-up delay

1 (set) – A stabilization delay is imposed as the MCU is started up from STOP mode (or from power-on reset).

0 (clear) – The oscillator start-up delay is bypassed and the MCU resumes processing within about four bus cycles. A stable external oscillator is required if this option is selected.

Note: Because DLY is set on reset, a delay is always imposed as the MCU is started up from power-on reset.

A mask option on the MC68HC11PH8 allows the selection of either a short or long delay time for power-on reset and exit from STOP mode; either 128 or 4064 bus cycles. *This option is not available on the MC68HC711PH8 where the delay time is 4064 bus cycles.*

CME — Clock monitor enable

1 (set) – Clock monitor enabled.

0 (clear) – Clock monitor disabled.

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.

In order to use both STOP and clock monitor, the CME bit should be cleared before executing STOP, then set again after recovering from STOP.

FCME — Force clock monitor enable

1 (set) – Clock monitor enabled; cannot be disabled until next reset.

0 (clear) – Clock monitor follows the state of the CME bit.

When FCME is set, slow or stopped clocks will cause a clock failure reset sequence. To utilize STOP mode, FCME should always be cleared.

CR[1:0] — COP timer rate select bits

The COP function can be clocked either by the internal E clock divided by 2^{15}, or by the output of the 8-bit modulus timer A, CLK64/4. These control bits determine a scaling factor for the watchdog timer period. See Table 10-1 and Table 10-2.

10.1.6 CONFIG — Configuration control register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Configuration control (CONFIG)	$003F	ROMAD	FREEZ	CLK4X	PAREN	NOSEC	NOCOP	ROMON	EEON	xxxx xxxx

Among other things, CONFIG controls the presence and location of EEPROM in the memory map and enables the COP watchdog system. A security feature that protects data in EEPROM and RAM is available on mask programmed MCUs.

CONFIG is made up of EEPROM cells and static working latches. The operation of the MCU is controlled directly by these latches and not the EEPROM byte. When programming the CONFIG register, the EEPROM byte is accessed. When the CONFIG register is read, the static latches are accessed.

These bits can be read at any time. The value read is the one latched into the register from the EEPROM cells during the last reset sequence. A new value programmed into this register is not readable until after a subsequent reset sequence.

On the MC68HC711PH8, and on the MC68HC11PH8 if selected by a mask option, the ROMON bit can be written at any time if MDA = 1 (expanded mode or special test mode). It cannot be written in bootstrap mode, and is forced to a logic one in single chip mode.

Other bits in CONFIG can be written at any time if SMOD = 1 (bootstrap or special test mode). If SMOD = 0 (single chip or expanded mode), these bits can only be written using the EEPROM programming sequence, and none of the bits are readable or active until latched via the next reset.

FREEZ is only active in expanded user mode.

ROMAD — ROM/*EPROM* mapping control (refer to Section 3)

1 (set) – ROM addressed from $4000 to $FFFF.

0 (clear) – ROM addressed from $0000 to $BFFF (expanded mode only).

In single chip mode, reset sets this bit.

FREEZ — Expanded user mode address bus freeze (refer to Section 3)

1 (set) – The external bus is only active when externally mapped resources are accessed (expanded mode only)

0 (clear) – Normal operation.

CLK4X — 4X clock enable (refer to Section 3)

1 (set) – 4XCLK or EXTALi driven out on the 4XOUT pin.

0 (clear) – 4XOUT pin disabled.

PAREN — Pull-up assignment register enable (refer to Section 4)

1 (set)	–	PPAR register enabled; pull-ups can be enabled using PPAR.
0 (clear)	–	PPAR register disabled; all pull-ups disabled.

NOSEC — EEPROM security disabled (refer to Section 3)

1 (set)	–	Disable security.
0 (clear)	–	Enable security.

NOCOP — COP system disable

1 (set)	–	COP system disabled.
0 (clear)	–	COP system enabled (forces reset on timeout).

ROMON — ROM/*EPROM* enable (refer to Section 3)

1 (set)	–	ROM/*EPROM* included in the memory map.
0 (clear)	–	ROM/*EPROM* excluded from the memory map.

EEON — EEPROM enable (refer to Section 3)

1 (set)	–	EEPROM included in the memory map.
0 (clear)	–	EEPROM excluded from the memory map.

10.2 Effects of reset

When a reset condition is recognized, the internal registers and control bits are forced to an initial state. Depending on the cause of the reset and the operating mode, the reset vector can be fetched from any of six possible locations, as shown in Table 10-3.

Table 10-3 Reset cause, reset vector and operating mode

Cause of reset	Normal mode vector	Special test or bootstrap
POR or $\overline{\text{RESET}}$ pin	\$FFFE, \$FFFF	\$BFFE, \$BFFF
Clock monitor failure	\$FFFC, \$FFFD	\$BFFC, \$BFFD
COP watchdog timeout	\$FFFA, \$FFFB	\$BFFA, \$BFFB

These initial states then control on-chip peripheral systems to force them to known start-up states, as described in the following paragraphs.

10.2.1 Central processing unit

After reset, the CPU fetches the restart vector from the appropriate address during the first three cycles, and begins executing instructions. The stack pointer and other CPU registers are indeterminate immediately after reset; however, the X and I interrupt mask bits in the condition code register (CCR) are set to mask any interrupt requests. Also, the S-bit in the CCR is set to inhibit the STOP mode.

10.2.2 Memory map

After reset, the INIT register is initialized to $00, putting the 2K bytes of RAM at locations $0080–$087F, and the control registers at locations $0000–$007F. The INIT2 register puts EEPROM at locations $0D00–$0FFF.

10.2.3 Parallel I/O

When a reset occurs in expanded operating modes, port B, C, and F pins used for parallel I/O are dedicated to the expansion bus. If a reset occurs during a single chip operating mode, all ports are configured as general purpose high-impedance inputs.

Note: Do not confuse pin function with the electrical state of the pin at reset. All general-purpose I/O pins configured as inputs at reset are in a high-impedance state. Port data registers reflect the port’s functional state at reset. The pin function is mode dependent.

10.2.4 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.

10.2.5 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.

10.2.6 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.

10.2.7 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.

10.2.8 8-bit modulus timer system

On reset, the clock source for Timer A is set at EXTALi/8 and the associated modulus register is initialized to $FF. Timers B and C are stopped on reset and pins PH6 and PH7 default to being general purpose I/O pins.

10.2.9 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 is initialized to $0004. 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 wake-up functions are disabled. The TDRE and TC status bits in the SCI status register are both set, 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 are cleared.

Note: The foregoing paragraph also applies to SCI2. The MI BUS function is disabled, since MIE2 is cleared on reset.

10.2.10 Serial peripheral interface (SPI)

The SPI1 and SPI2 systems are disabled by reset. Their associated port pins default to being general purpose I/O lines.

10.2.11 Analog-to-digital converter

The A/D converter configuration is indeterminate after reset. The ADPU bit is cleared by reset, which disables the A/D system. The conversion complete flag is cleared by reset.

10.2.12 LCD module

The LCD module is disabled by reset. PB4-PB7 default to being general purpose I/O lines in single chip mode, or higher order address outputs in expanded mode.

10.2.13 System

The EEPROM programming controls are disabled, so the memory system is configured for normal read operation. PSEL[4:0] are initialized with the binary value %00110, causing the external $\overline{\text{IRQ}}$ pin to have the highest I-bit interrupt priority. The $\overline{\text{IRQ}}$ pin is configured for level-sensitive operation (for wired-OR systems). The RBOOT, SMOD, and MDA bits in the HPRIO register reflect the status of the MODB and MODA inputs at the rising edge of reset. The DLY control bit is set to specify that an oscillator start-up delay is imposed upon recovery from STOP mode or power-on reset. The clock monitor system is disabled because CME and FCME are cleared.

10.3 Reset and interrupt priority

Resets and interrupts have a hardware priority that determines which reset or interrupt is serviced first when simultaneous requests occur. Any maskable interrupt can be given priority over other maskable interrupts.

The first six interrupt sources are not maskable by the I-bit in the CCR. The priority arrangement for these sources is fixed and is as follows:

1) POR or $\overline{\text{RESET}}$ pin
2) Clock monitor reset
3) COP watchdog reset
4) $\overline{\text{XIRQ}}$ interrupt
– Illegal opcode interrupt — see Section 10.4.3 for details of handling
– Software interrupt (SWI) — see Section 10.4.4 for details of handling

The maskable interrupt sources have the following priority arrangement:

5) $\overline{\text{IRQ}}$
6) Real-time interrupt
7) Timer input capture 1
8) Timer input capture 2
9) Timer input capture 3
10) Timer output compare 1
11) Timer output compare 2
12) Timer output compare 3
13) Timer output compare 4
14) Timer input capture 4/output compare 5
15) SPI2 transfer complete
16) SCI2/MI BUS system
17) Timer overflow
18) 8-bit modulus timers
19) Pulse accumulator overflow
20) Pulse accumulator input edge
21) Wired-OR port H
22) SPI1 transfer complete
23) SCI1 system

Any one of these maskable interrupts can be assigned the highest maskable interrupt priority by writing the appropriate value to the PSEL bits in the HPRIO register. Otherwise, the priority arrangement remains the same. An interrupt that is assigned highest priority is still subject to global masking by the I-bit in the CCR, or by any associated local bits. Interrupt vectors are not affected by priority assignment. To avoid race conditions, HPRIO can only be written while I-bit interrupts are inhibited.

10.3.1 HPRIO — Highest priority I-bit interrupt and misc. register

	Address	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	State on reset
Highest priority interrupt (HPRIO)	$003C	RBOOT	SMOD	MDA	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0	xxx0 0110

RBOOT, SMOD, and MDA bits depend on power-up initialization mode and can only be written in special modes when SMOD = 1. Refer to Table 3-4.

RBOOT — Read bootstrap ROM (refer to Section 3)

1 (set) – Bootloader ROM enabled, at $BE40–$BFFF.

0 (clear) – Bootloader ROM disabled and not in map.

SMOD — Special mode select (refer to Section 3)

1 (set) – Special mode variation in effect.

0 (clear) – Normal mode variation in effect.

MDA — Mode select A (refer to Section 3)

1 (set) – Normal expanded or special test mode in effect.

0 (clear) – Normal single chip or special bootstrap mode in effect.

PSEL[4:0] — Priority select bits

These bits select one interrupt source to be elevated above all other I-bit-related sources and can be written to only while the I-bit in the CCR is set (interrupts disabled). See Table 10-4.

Table 10-4 Highest priority interrupt selection

PSELx					Interrupt source promoted
4	3	2	1	0	
0	0	0	X	X	Reserved (default to $\overline{\text{IRQ}}$)
0	0	1	0	0	Reserved (default to $\overline{\text{IRQ}}$)
0	0	1	0	1	Reserved (default to $\overline{\text{IRQ}}$)
0	0	1	1	0	$\overline{\text{IRQ}}$ (external pin)
0	0	1	1	1	Real-time interrupt
0	1	0	0	0	Timer input capture 1
0	1	0	0	1	Timer input capture 2
0	1	0	1	0	Timer input capture 3
0	1	0	1	1	Timer output compare 1
0	1	1	0	0	Timer output compare 2
0	1	1	0	1	Timer output compare 3
0	1	1	1	0	Timer output compare 4
0	1	1	1	1	Timer output compare 5/input capture 4
1	0	0	0	0	Timer overflow
1	0	0	0	1	Pulse accumulator overflow
1	0	0	1	0	Pulse accumulator input edge
1	0	0	1	1	SPI1 serial transfer complete
1	0	1	0	0	SCI1 serial system
1	0	1	0	1	SPI2 serial transfer complete
1	0	1	1	0	SCI2/MI BUS serial system
1	0	1	1	1	8-bit modulus timers
1	1	0	0	0	Wired-OR port H
1	1	0	0	1	Reserved (default to $\overline{\text{IRQ}}$)
1	1	0	1	X	Reserved (default to $\overline{\text{IRQ}}$)
1	1	1	X	X	Reserved (default to $\overline{\text{IRQ}}$)

Table 10-5 Interrupt and reset vector assignments

Vector address	Interrupt source	CCR mask bit	Local mask
FFC0, C1 – FFCC, CD	reserved	—	—
FFCE, CF	• Wired-OR port H	I	IEH[7:0]
FFD0, D1	• 8-bit modulus timer A underflow • 8-bit modulus timer B underflow • 8-bit modulus timer C underflow	I	T8AI T8BI T8CI
FFD2, D3	• SCI2/MI BUS receive data register full • SCI2/MI BUS receiver overrun • SCI2 transmit data register empty • SCI2 transmit complete • SCI2 idle line detect	I	RIE2 RIE2 TIE2 TCIE2 ILIE2
FFD4, D5	SPI2 serial transfer complete	I	SP2IE
FFD6, D7	• SCI1 receive data register full • SCI1 receiver overrun • SCI1 transmit data register empty • SCI1 transmit complete • SCI1 idle line detect	I	RIE RIE TIE TCIE ILIE
FFD8, D9	SPI1 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	$\overline{\text{IRQ}}$ 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

10.4 Interrupts

Excluding reset type interrupts, the MC68HC11PH8 has 22 interrupt vectors that support 32 interrupt sources. The 19 maskable interrupts are generated by on-chip peripheral systems. These interrupts are recognized when the global interrupt mask bit (I) in the condition code register (CCR) is clear. The three nonmaskable interrupt sources are illegal opcode trap, software interrupt, and $\overline{\text{XIRQ}}$ pin. Refer to Table 10-5, which shows the interrupt sources and vector assignments for each source.

For some interrupt sources, such as the SCI interrupts, the flags are automatically cleared during the normal course of responding to the interrupt requests. For example, the RDRF flag in the SCI system is cleared by the automatic clearing mechanism consisting of a read of the SCI status register while RDRF is set, followed by a read of the SCI data register. The normal response to an RDRF interrupt request would be to read the SCI status register to check for receive errors, then to read the received data from the SCI data register. These two steps satisfy the automatic clearing mechanism without requiring any special instructions.

10.4.1 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 10-6. 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 end of the interrupt service routine, the return from interrupt instruction is executed and the saved registers are pulled from the stack in reverse order so that normal program execution can resume. Refer to Section 11 for further information.

Table 10-6 Stacking order on entry to interrupts

Memory location	CPU registers
SP	PCL
SP – 1	PCH
SP – 2	IYL
SP – 3	IYH
SP – 4	IXL
SP – 5	IXH
SP – 6	ACCA
SP – 7	ACCB
SP – 8	CCR

10.4.2 Nonmaskable interrupt request ($\overline{\text{XIRQ}}$)

Nonmaskable interrupts are useful because they can always interrupt CPU operations. The most common use for such an interrupt is for serious system problems, such as program runaway or power failure. The $\overline{\text{XIRQ}}$ input is an updated version of the $\overline{\text{NMI}}$ (nonmaskable interrupt) input of earlier MCUs.

Upon reset, both the X-bit and I-bit of the CCR are set to inhibit all maskable interrupts and $\overline{\text{XIRQ}}$. After minimum system initialization, software can clear the X-bit by a TAP instruction, enabling $\overline{\text{XIRQ}}$ interrupts. Thereafter, software cannot set the X-bit. Thus, an $\overline{\text{XIRQ}}$ interrupt is a nonmaskable interrupt. Because the operation of the I-bit-related interrupt structure has no effect on the X-bit, the internal $\overline{\text{XIRQ}}$ pin remains unmasked. In the interrupt priority logic, the $\overline{\text{XIRQ}}$ interrupt has a higher priority than any source that is maskable by the I-bit. All I-bit-related interrupts operate normally with their own priority relationship.

When an I-bit-related interrupt occurs, the I-bit is automatically set by hardware after stacking the CCR byte. The X-bit is not affected. When an X-bit-related interrupt occurs, both the X and I bits are automatically set by hardware after stacking the CCR. A return from interrupt instruction restores the X and I bits to their pre-interrupt request state.

10.4.3 Illegal opcode trap

Because not all possible opcodes or opcode sequences are defined, the MCU includes an illegal opcode detection circuit, which generates an interrupt request. When an illegal opcode is detected and the interrupt is recognized, the current value of the program counter is stacked. After interrupt service is complete, the user should reinitialize the stack pointer to ensure that repeated execution of illegal opcodes does not cause stack underflow. Left uninitialized, the illegal opcode vector can point to a memory location that contains an illegal opcode. This condition causes an infinite loop that causes stack underflow. The stack grows until the system crashes.

The illegal opcode trap mechanism works for all unimplemented opcodes on all four opcode map pages. The address stacked as the return address for the illegal opcode interrupt is the address of the first byte of the illegal opcode. Otherwise, it would be almost impossible to determine whether the illegal opcode had been one or two bytes. The stacked return address can be used as a pointer to the illegal opcode, so that the illegal opcode service routine can evaluate the offending opcode.

10.4.4 Software interrupt

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.

10.4.5 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.

10.4.6 Reset and interrupt processing

The following flow diagrams illustrate the reset and interrupt process. Figure 10-1 and Figure 10-2 illustrate how the CPU begins from a reset and how interrupt detection relates to normal opcode fetches. Figure 10-3 to Figure 10-4 provide an expanded version of a block in Figure 10-1 and illustrate interrupt priorities. Figure 10-7 shows the resolution of interrupt sources within the SCI subsystem.

10.5 Low power operation

Both STOP and WAIT suspend CPU operation until a reset or interrupt occurs. The WAIT condition suspends processing and reduces power consumption to an intermediate level. The STOP condition turns off all on-chip clocks and reduces power consumption to an absolute minimum while retaining the contents of all bytes of the RAM.

10.5.1 WAIT

The WAI opcode places the MCU in the WAIT condition, 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 $\overline{\text{XIRQ}}$, or any of the internally generated interrupts, such as the timer or serial interrupts. The on-chip crystal oscillator remains active throughout the WAIT stand-by 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 PH2 clock to the free-running timer system is stopped if the I-bit is set and the COP system is disabled by NOCOP being set. In addition, further power can be saved if the clock to the 16-bit counter is stopped by clearing the T16EN bit in PLLCR, with the PLL active (see Section 8.1.1.1). Several other systems can also 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 zero and halting the RC clock (CSEL cleared). 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 (lowest power consumption is achieved when RE=TE=0). Power consumption is reduced if all the PWM enable bits (PWEN[4:1]) are cleared, thereby disabling every PWM channel. Setting the WEN bit in PLLCR will result in WAIT mode using a slower clock and hence less power (see Section 2.5). Therefore the power consumption in WAIT is dependent on the particular application.

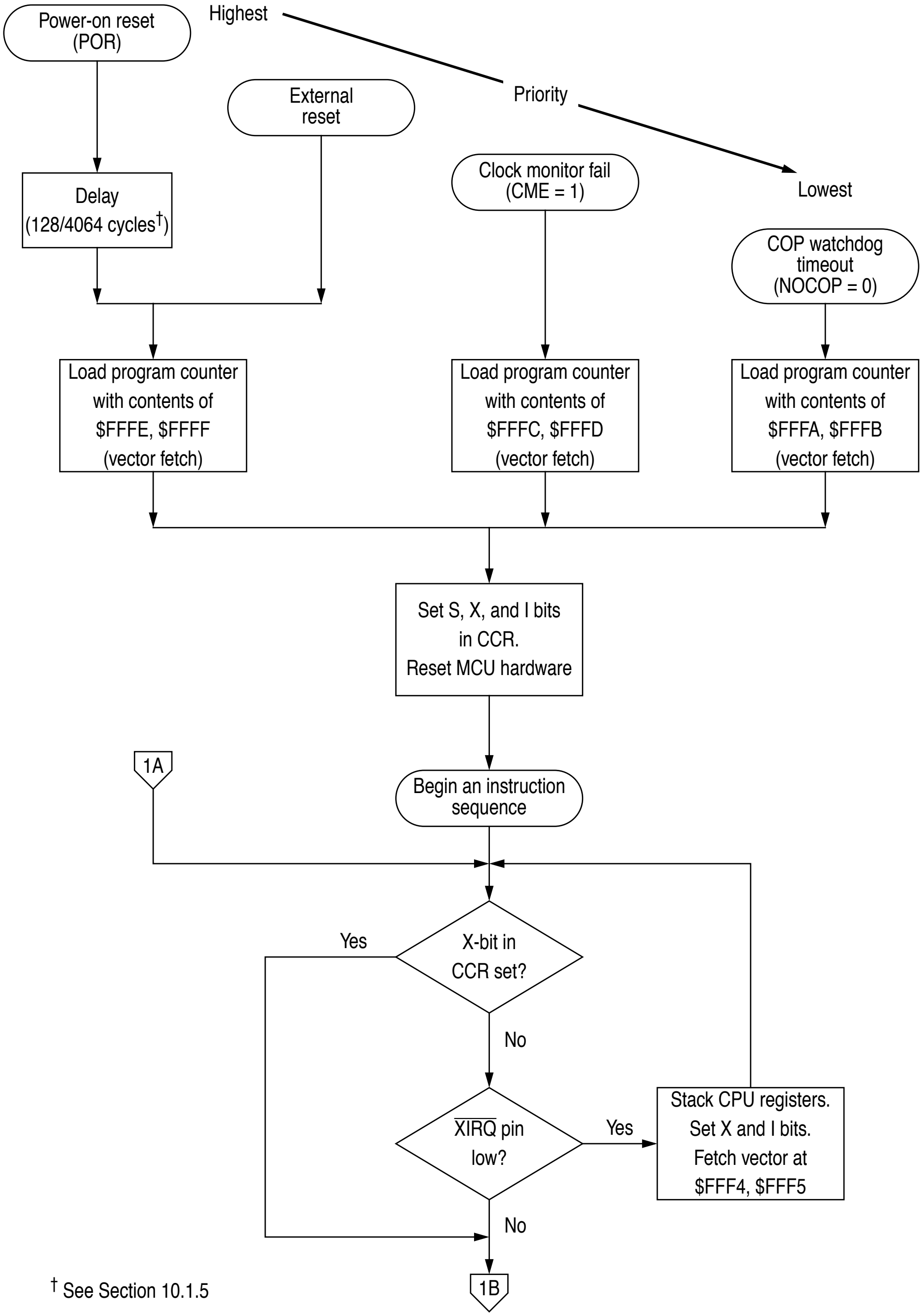
10.5.2 STOP

Executing the STOP instruction while the S-bit in the CCR is clear places the MCU in the STOP condition. If the S-bit is set, the STOP opcode is treated as a no-op (NOP). The STOP condition offers minimum power consumption because all clocks, including the crystal oscillator, are stopped while in this mode. To exit STOP and resume normal processing, a logic low level must be applied to one of the external interrupts ($\overline{\text{IRQ}}$ or $\overline{\text{XIRQ}}$) or to the $\overline{\text{RESET}}$ pin. A keyboard interrupt on port H or a pending edge-triggered $\overline{\text{IRQ}}$ can also bring the CPU out of STOP.

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

To use the $\overline{\text{IRQ}}$ pin as a means of recovering from STOP, the I-bit in the CCR must be clear ($\overline{\text{IRQ}}$ not masked). The $\overline{\text{XIRQ}}$ pin can be used to wake up the MCU from STOP regardless of the state of the X-bit in the CCR, although the recovery sequence depends on the state of the X-bit. If X is clear ($\overline{\text{XIRQ}}$ not masked), the MCU starts up, beginning with the stacking sequence leading to normal service of the $\overline{\text{XIRQ}}$ request. If X is set ($\overline{\text{XIRQ}}$ masked or inhibited), then processing continues with the instruction that immediately follows the STOP instruction, and no $\overline{\text{XIRQ}}$ interrupt service is requested or pending.

Because the oscillator is stopped in STOP mode, a restart delay may be imposed to allow oscillator stabilization upon leaving STOP. If the internal oscillator is being used, this delay is required; however, if a stable external oscillator is being used, the DLY control bit can be used to bypass this start-up delay. The DLY control bit is set by reset and can be optionally cleared during initialization. If the DLY equal to zero option is used to avoid start-up delay on recovery from STOP, then reset should not be used as the means of recovering from STOP, as this causes DLY to be set again by reset, imposing the restart delay. This same delay also applies to power-on-reset, regardless of the state of the DLY control bit, but does not apply to a reset while the clocks are running. See Section 3.3.2.4.


Power-on reset (POR)
Highest
Priority
Lowest
External reset
Clock monitor fail (CME = 1)
COP watchdog timeout (NOCOP = 0)
Delay (128/4064 cycles†)
Load program counter with contents of $FFFE, $FFFF (vector fetch)
Load program counter with contents of $FFFC, $FFFD (vector fetch)
Load program counter with contents of $FFFA, $FFFB (vector fetch)
Set S, X, and I bits in CCR. Reset MCU hardware
1A
Begin an instruction sequence
Yes
X-bit in CCR set?
No
XIRQ pin low?
Yes
Stack CPU registers. Set X and I bits. Fetch vector at $FFF4, $FFF5
No
1B

† See Section 10.1.5

Figure 10-1 Processing flow out of reset (1 of 2)

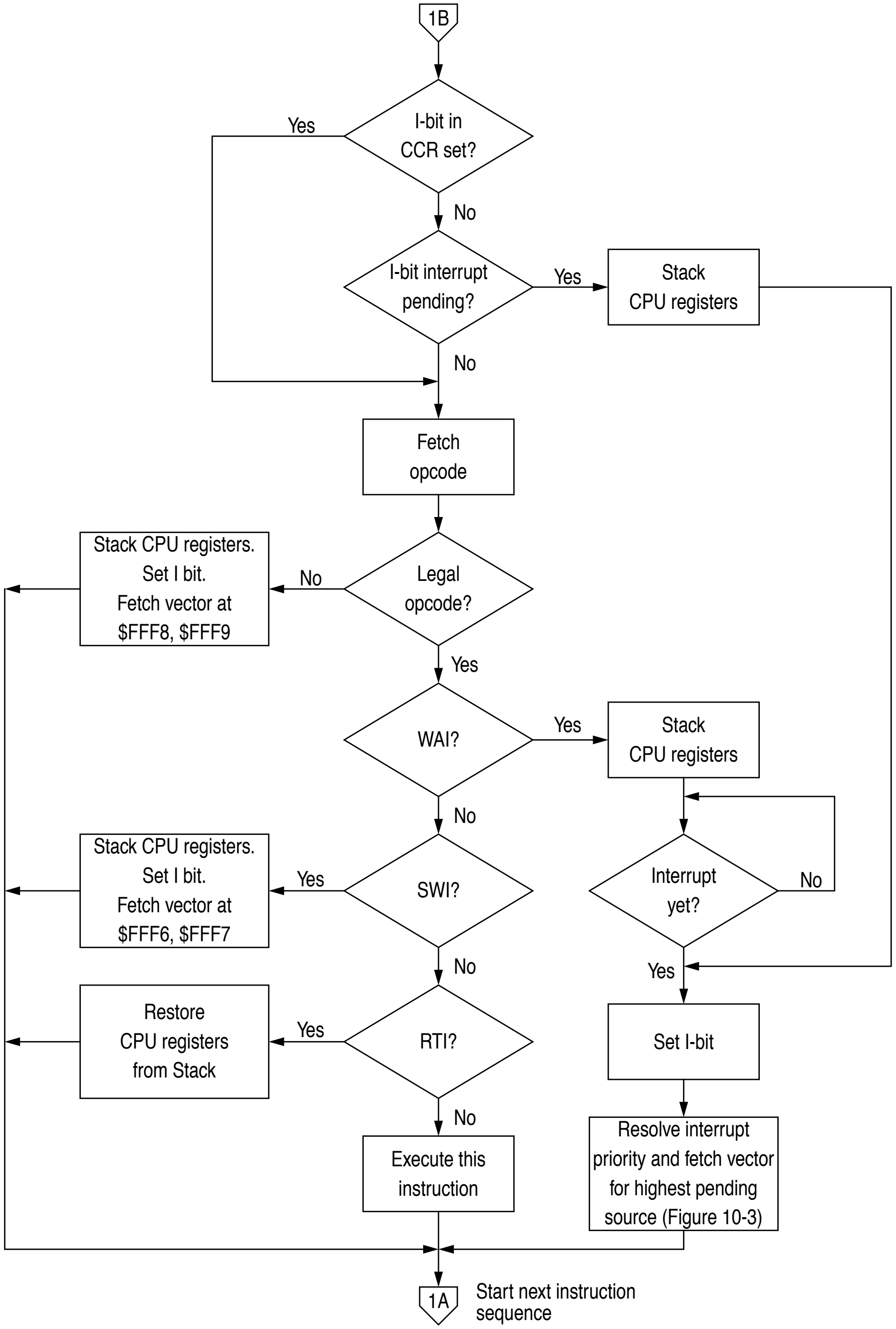

1B
I-bit in CCR set?
Yes
No
I-bit interrupt pending?
Yes
Stack CPU registers
No
Fetch opcode
Legal opcode?
No
Stack CPU registers. Set I bit. Fetch vector at $FFF8, $FFF9
Yes
WAI?
Yes
Stack CPU registers
No
SWI?
Yes
Stack CPU registers. Set I bit. Fetch vector at $FFF6, $FFF7
Interrupt yet?
No
Yes
No
RTI?
Yes
Restore CPU registers from Stack
Set I-bit
No
Execute this instruction
Resolve interrupt priority and fetch vector for highest pending source (Figure 10-3)
1A
Start next instruction sequence

Figure 10-2 Processing flow out of reset (2 of 2)

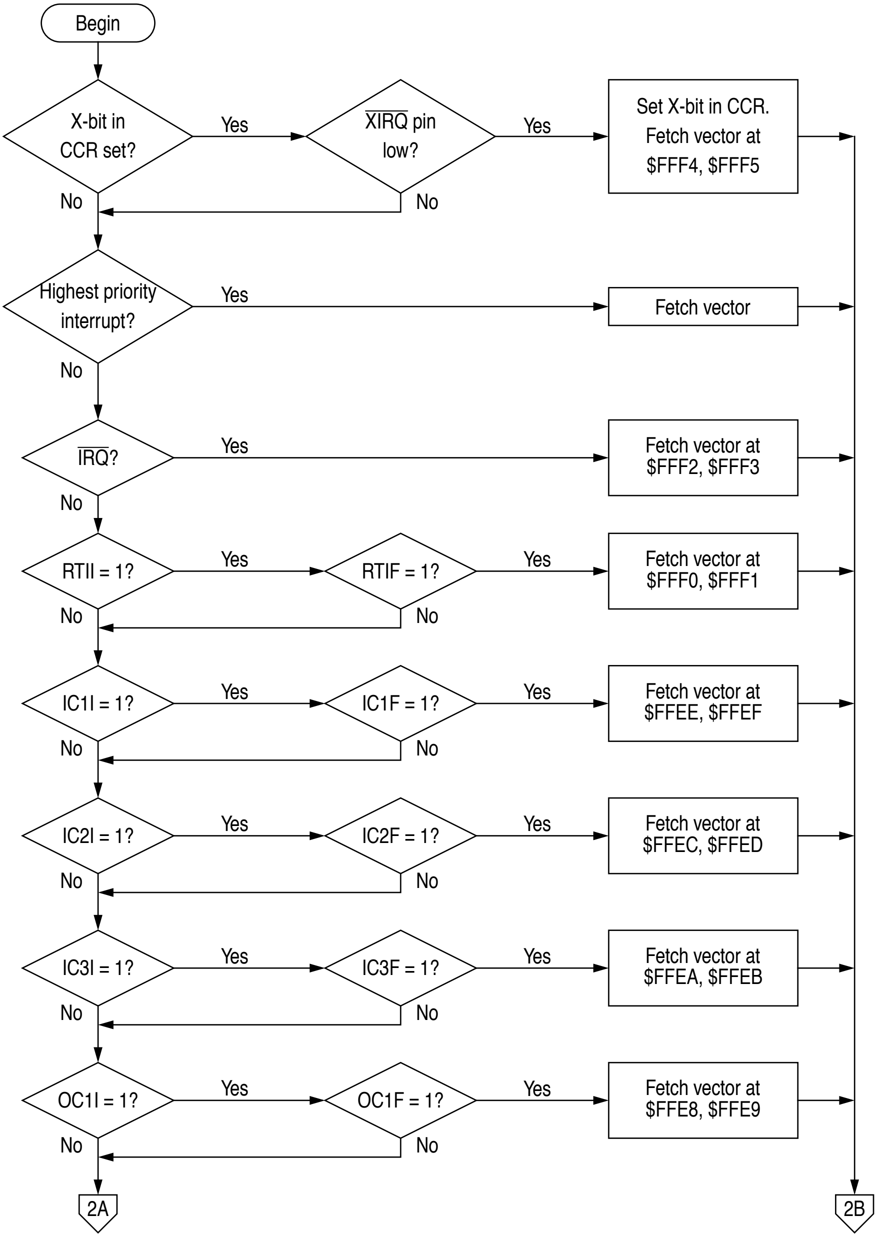
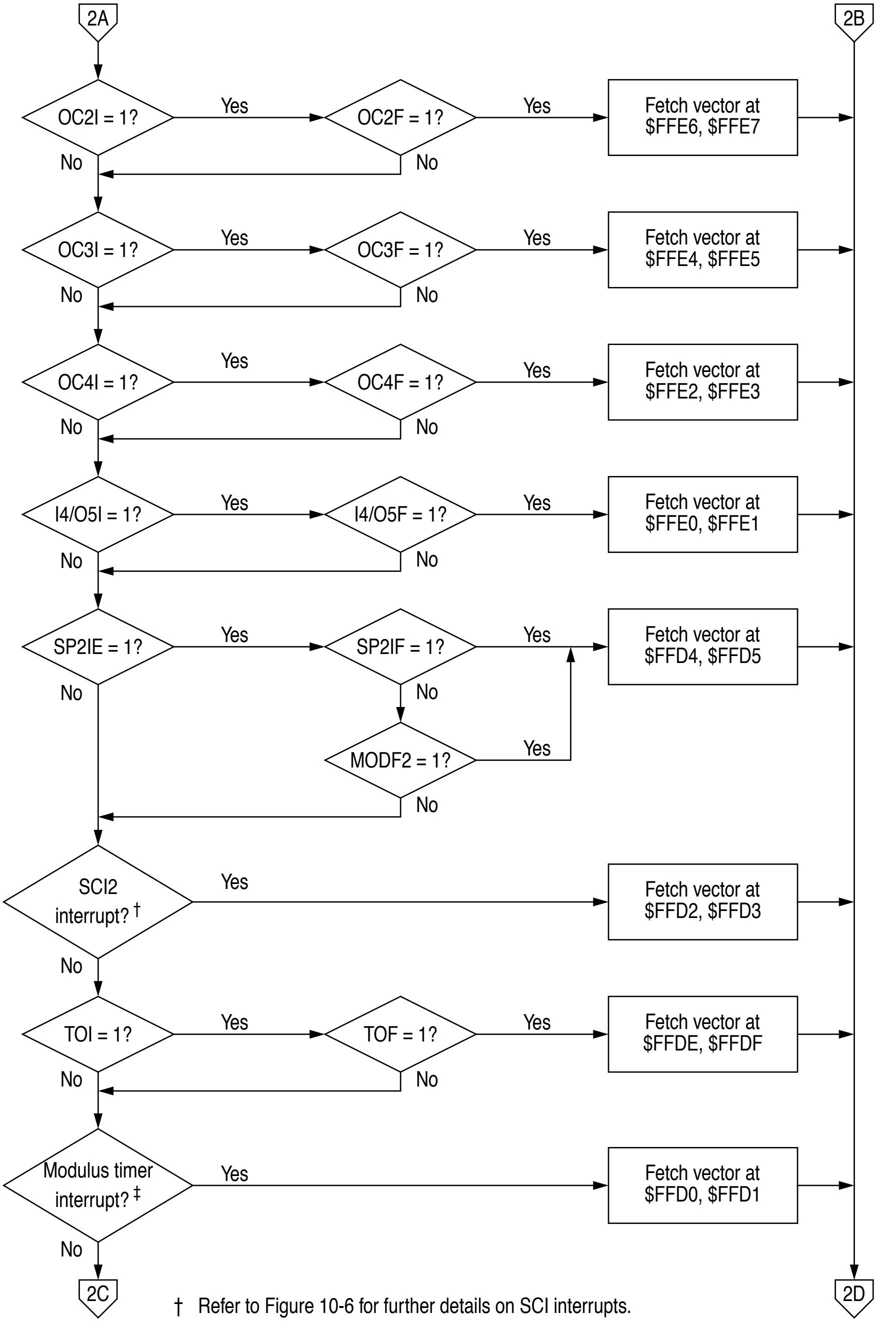

Begin
X-bit in CCR set?
Yes
No
$\overline{XIRQ}$ pin low?
Yes
No
Set X-bit in CCR. Fetch vector at $FFF4, $FFF5
Highest priority interrupt?
Yes
No
Fetch vector
$\overline{IRQ}$?
Yes
No
Fetch vector at $FFF2, $FFF3
RTII = 1?
Yes
No
RTIF = 1?
Yes
No
Fetch vector at $FFF0, $FFF1
IC1I = 1?
Yes
No
IC1F = 1?
Yes
No
Fetch vector at $FFEE, $FFEF
IC2I = 1?
Yes
No
IC2F = 1?
Yes
No
Fetch vector at $FFEC, $FFED
IC3I = 1?
Yes
No
IC3F = 1?
Yes
No
Fetch vector at $FFEA, $FFEB
OC1I = 1?
Yes
No
OC1F = 1?
Yes
No
Fetch vector at $FFE8, $FFE9
2A
2B

Figure 10-3 Interrupt priority resolution (1 of 3)


2A
2B
OC2I = 1?
Yes
OC2F = 1?
Yes
Fetch vector at $FFE6, $FFE7
No
No
OC3I = 1?
Yes
OC3F = 1?
Yes
Fetch vector at $FFE4, $FFE5
No
No
OC4I = 1?
Yes
OC4F = 1?
Yes
Fetch vector at $FFE2, $FFE3
No
No
I4/O5I = 1?
Yes
I4/O5F = 1?
Yes
Fetch vector at $FFE0, $FFE1
No
No
SP2IE = 1?
Yes
SP2IF = 1?
Yes
Fetch vector at $FFD4, $FFD5
No
No
MODF2 = 1?
Yes
No
SCI2 interrupt? †
Yes
Fetch vector at $FFD2, $FFD3
No
TOI = 1?
Yes
TOF = 1?
Yes
Fetch vector at $FFDE, $FFDF
No
No
Modulus timer interrupt? ‡
Yes
Fetch vector at $FFD0, $FFD1
No
2C
2D

† Refer to Figure 10-6 for further details on SCI interrupts.

‡ Refer to Figure 10-7 for further details on modulus timer interrupts.

Figure 10-4 Interrupt priority resolution (2 of 3)

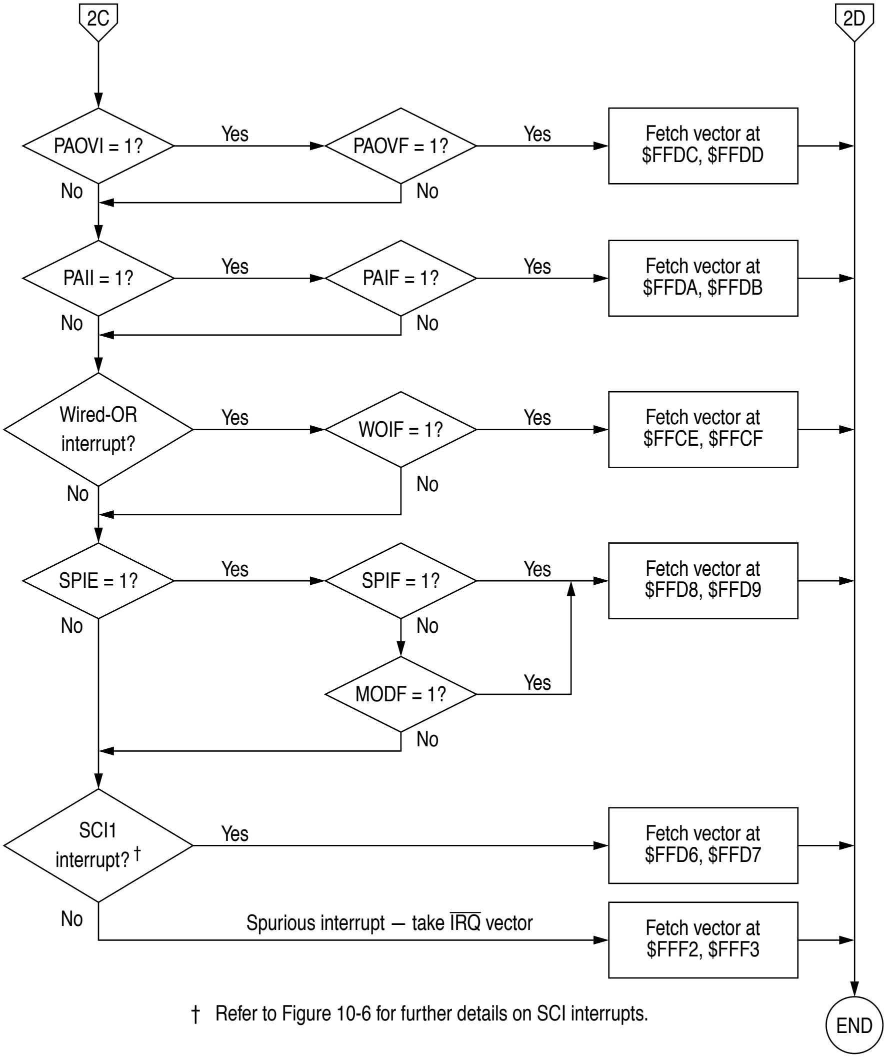

2C
2D
PAOVI = 1?
Yes
PAOVF = 1?
Yes
Fetch vector at $FFDC, $FFDD
No
No
PAII = 1?
Yes
PAIF = 1?
Yes
Fetch vector at $FFDA, $FFDB
No
No
Wired-OR interrupt?
Yes
WOIF = 1?
Yes
Fetch vector at $FFCE, $FFCF
No
No
SPIE = 1?
Yes
SPIF = 1?
Yes
Fetch vector at $FFD8, $FFD9
No
No
MODF = 1?
Yes
No
SCI1 interrupt? †
Yes
Fetch vector at $FFD6, $FFD7
No
Spurious interrupt — take IRQ vector
Fetch vector at $FFF2, $FFF3
END
† Refer to Figure 10-6 for further details on SCI interrupts.

Figure 10-5 Interrupt priority resolution (3 of 3)

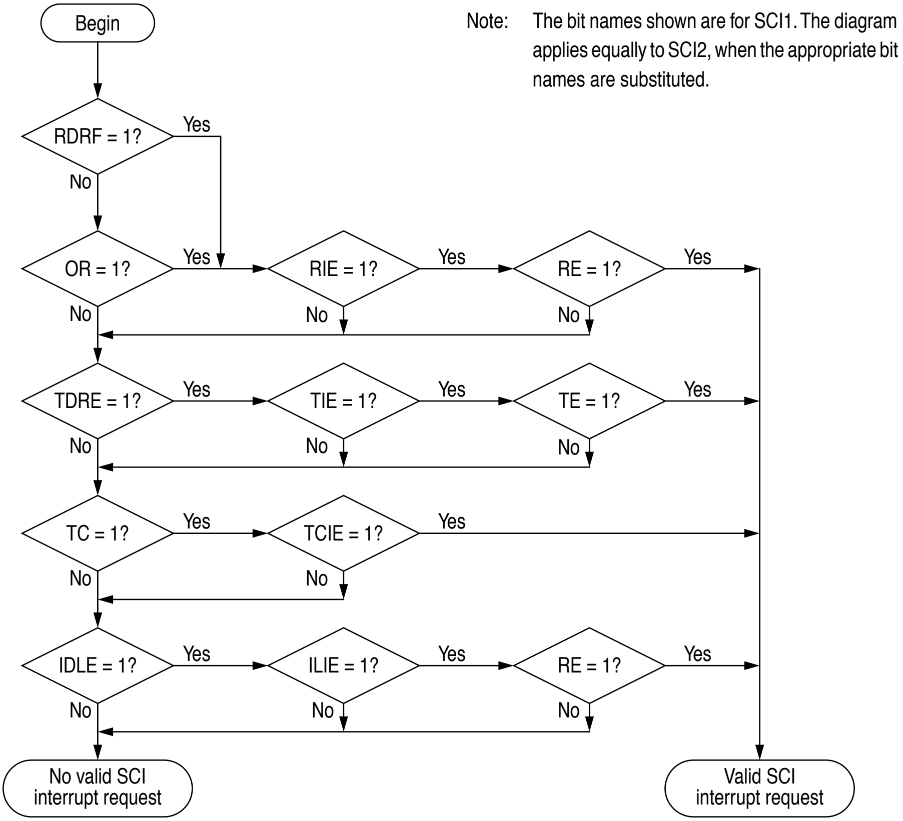

Begin
Note: The bit names shown are for SCI1. The diagram applies equally to SCI2, when the appropriate bit names are substituted.
RDRF = 1?
Yes
No
OR = 1?
RIE = 1?
RE = 1?
TDRE = 1?
TIE = 1?
TE = 1?
TC = 1?
TCIE = 1?
IDLE = 1?
ILIE = 1?
RE = 1?
No valid SCI interrupt request
Valid SCI interrupt request

Figure 10-6 Interrupt source resolution within the SCI subsystem

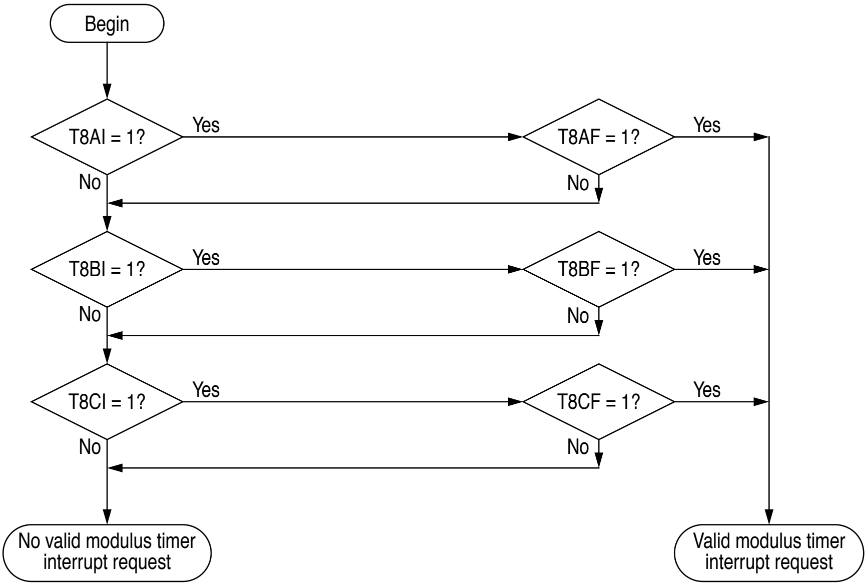


Figure 10-7 Interrupt source resolution within the 8-bit modulus timer subsystem

THIS PAGE INTENTIONALLY LEFT BLANK

11

CPU CORE AND INSTRUCTION SET

This section discusses the M68HC11 central processing unit (CPU) architecture, its addressing modes and the instruction set. For more detailed information on the instruction set, refer to the ***M68HC11 Reference Manual (M68HC11RM/AD)***.

The CPU is designed to treat all peripheral, I/O and memory locations identically, as addresses in the 64Kbyte memory map. This is referred to as memory-mapped I/O. There are no special instructions for I/O that are separate from those used for memory. This architecture also allows accessing an operand from an external memory location with no execution-time penalty.

11.1 Registers

M68HC11 CPU registers are an integral part of the CPU and are not addressed as if they were memory locations. The seven registers are shown in Figure 11-1 and are discussed in the following paragraphs.

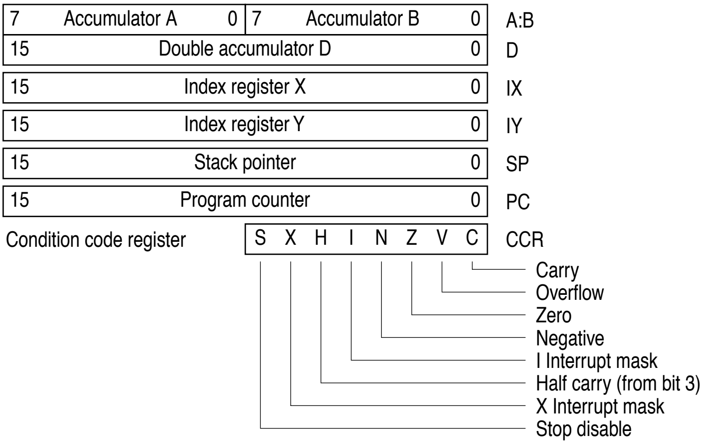


Figure 11-1 Programming model

11.1.1 Accumulators A, B and D

Accumulators A and B are general purpose 8-bit registers that hold operands and results of arithmetic calculations or data manipulations. For some instructions, these two accumulators are treated as a single double-byte (16-bit) accumulator called accumulator D. Although most operations can use accumulators A or B interchangeably, the following exceptions apply:

- The ABX and ABY instructions add the contents of 8-bit accumulator B to the contents of 16-bit register X or Y, but there are no equivalent instructions that use A instead of B.
- The TAP and TPA instructions transfer data from accumulator A to the condition code register, or from the condition code register to accumulator A, however, there are no equivalent instructions that use B rather than A.
- The decimal adjust accumulator A (DAA) instruction is used after binary-coded decimal (BCD) arithmetic operations, but there is no equivalent BCD instruction to adjust accumulator B.
- The add, subtract, and compare instructions associated with both A and B (ABA, SBA, and CBA) only operate in one direction, making it important to plan ahead to ensure the correct operand is in the correct accumulator.

11.1.2 Index register X (IX)

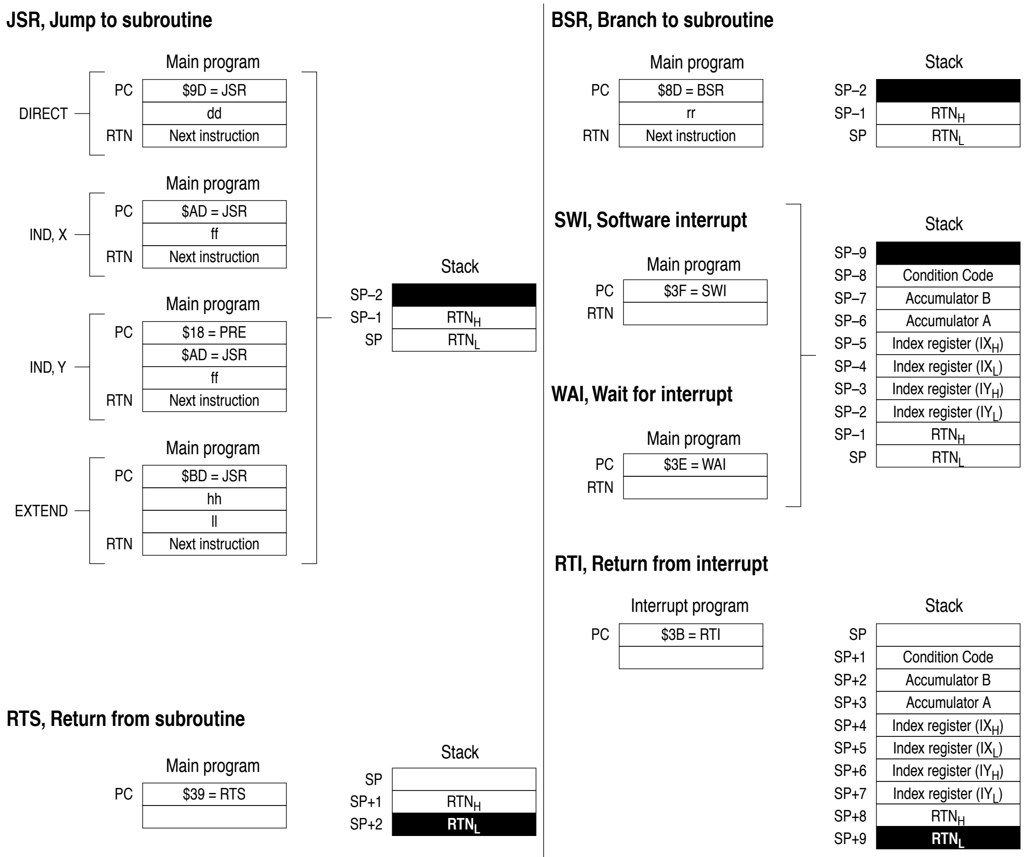
The IX register provides a 16-bit indexing value that can be added to the 8-bit offset provided in an instruction to create an effective address. The IX register can also be used as a counter or as a temporary storage register.

11.1.3 Index register Y (IY)

The 16-bit IY register performs an indexed mode function similar to that of the IX register. However, most instructions using the IY register require an extra byte of machine code and an extra cycle of execution time because of the way the opcode map is implemented. Refer to Section 11.3 for further information.

11.1.4 Stack pointer (SP)

The M68HC11 CPU has an automatic program stack. This stack can be located anywhere in the address space and can be any size up to the amount of memory available in the system. Normally the SP is initialized by one of the first instructions in an application program. The stack is configured as a data structure that grows downward from high memory to low memory. Each time a new byte is pushed onto the stack, the SP is decremented. Each time a byte is pulled from the stack, the SP is incremented. At any given time, the SP holds the 16-bit address of the next free location in the stack. Figure 11-2 is a summary of SP operations.



LEGEND

RTN	Address of the next instruction in the main program, to be executed on return from subroutine
RTN_H	More significant byte of return address
RTN_L	Less significant byte of return address
■	Shaded cells show stack pointer position after the operation is complete
dd	8-bit direct address ($0000–$00FF); the high byte is assumed to be $00
ff	8-bit positive offset ($00 to $FF (0 to 256)) is added to the index register contents
hh	High order byte of 16-bit extended address
ll	Low order byte of 16-bit extended address
rr	Signed relative offset ($80 to $7F (–128 to +127)); offset is relative to the address following the offset byte

Figure 11-2 Stacking operations

When a subroutine is called by a jump to subroutine (JSR) or branch to subroutine (BSR) instruction, the address of the instruction after the JSR or BSR is automatically pushed onto the stack, less significant byte first. When the subroutine is finished, a return from subroutine (RTS) instruction is executed. The RTS pulls the previously stacked return address from the stack, and loads it into the program counter. Execution then continues at this recovered return address.

When an interrupt is recognized, the current instruction finishes normally, the return address (the current value in the program counter) is pushed onto the stack, all of the CPU registers are pushed onto the stack, and execution continues at the address specified by the vector for the interrupt. At the end of the interrupt service routine, an 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.

There are instructions that push and pull the A and B accumulators and the X and Y index registers. These instructions 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.

11.1.5 Program counter (PC)

The program counter, a 16-bit register, contains the address of the next instruction to be executed. After reset, the program counter is initialized from one of six possible vectors, depending on operating mode and the cause of reset.

Table 11-1 Reset vector comparison

	POR or $\overline{\text{RESET}}$ pin	Clock monitor	COP watchdog
Normal	$FFFE, $FFFF	$FFFC, $FFFD	$FFFA, $FFFB
Test or Boot	$BFFE, $BFFF	$BFFE, $BFFF	$BFFE, $BFFF

11.1.6 Condition code register (CCR)

This 8-bit register contains five condition code indicators (C, V, Z, N, and H), two interrupt masking bits, (IRQ and XIRQ) and a stop disable bit (S). In the M68HC11 CPU, condition codes are automatically updated by most instructions. For example, load accumulator A (LDAA) and store accumulator A (STAA) instructions automatically set or clear the N, Z, and V condition code flags. Pushes, pulls, add B to X (ABX), add B to Y (ABY), and transfer/exchange instructions do not affect the condition codes. Refer to Table 11-2, which shows the condition codes that are affected by a particular instruction.

11.1.6.1 Carry/borrow (C)

The C-bit is set if the arithmetic logic unit (ALU) performs a carry or borrow during an arithmetic operation. The C-bit also acts as an error flag for multiply and divide operations. Shift and rotate instructions operate with and through the carry bit to facilitate multiple-word shift operations.

11.1.6.2 Overflow (V)

The overflow bit is set if an operation causes an arithmetic overflow. Otherwise, the V-bit is cleared.

11.1.6.3 Zero (Z)

The Z-bit is set if the result of an arithmetic, logic, or data manipulation operation is zero. Otherwise, the Z-bit is cleared. Compare instructions do an internal implied subtraction and the condition codes, including Z, reflect the results of that subtraction. A few operations (INX, DEX, INY, and DEY) affect the Z-bit and no other condition flags. For these operations, only = and ≠ conditions can be determined.

11.1.6.4 Negative (N)

The N-bit is set if the result of an arithmetic, logic, or data manipulation operation is negative; otherwise, the N-bit is cleared. A result is said to be negative if its most significant bit (MSB) is set (MSB = 1). A quick way to test whether the contents of a memory location has the MSB set is to load it into an accumulator and then check the status of the N-bit.

11.1.6.5 Interrupt mask (I)

The interrupt request (IRQ) mask (I-bit) is a global mask that disables all maskable interrupt sources. While the I-bit is set, interrupts can become pending, but the operation of the CPU continues uninterrupted until the I-bit is cleared. After any reset, the I-bit is set by default and can only be cleared by a software instruction. When an interrupt is recognized, the I-bit is set after the registers are stacked, but before the interrupt vector is fetched. After the interrupt has been serviced, a return from interrupt instruction is normally executed, restoring the registers to the values that were present before the interrupt occurred. Normally, the I-bit is zero after a return from interrupt is executed. Although the I-bit can be cleared within an interrupt service routine, ‘nesting’ interrupts in this way should only be done when there is a clear understanding of latency and of the arbitration mechanism. Refer to Section 10.

11.1.6.6 Half carry (H)

The H-bit is set when a carry occurs between bits 3 and 4 of the arithmetic logic unit during an ADD, ABA, or ADC instruction. Otherwise, the H-bit is cleared. Half carry is used during BCD operations.

11.1.6.7 X interrupt mask (X)

The XIRQ mask (X) bit disables interrupts from the $\overline{\text{XIRQ}}$ pin. After any reset, X is set by default and must be cleared by a software instruction. When an $\overline{\text{XIRQ}}$ interrupt is recognized, the X- and I-bits are set after the registers are stacked, but before the interrupt vector is fetched. After the interrupt has been serviced, an RTI instruction is normally executed, causing the registers to be restored to the values that were present before the interrupt occurred. The X interrupt mask bit is set only by hardware $\overline{\text{RESET}}$ or $\overline{\text{XIRQ}}$ acknowledge). X is cleared only by program instruction (TAP, where the associated bit of A is 0; or RTI, where bit 6 of the value loaded into the CCR from the stack has been cleared). There is no hardware action for clearing X.

11.1.6.8 Stop disable (S)

Setting the STOP disable (S) bit prevents the STOP instruction from putting the M68HC11 into a low-power stop condition. If the STOP instruction is encountered by the CPU while the S-bit is set, it is treated as a no-operation (NOP) instruction, and processing continues to the next instruction. S is set by reset — STOP disabled by default.

11.2 Data types

The M68HC11 CPU supports the following data types:

- Bit data
- 8-bit and 16-bit signed and unsigned integers
- 16-bit unsigned fractions
- 16-bit addresses

A byte is eight bits wide and can be accessed at any byte location. A word is composed of two consecutive bytes with the most significant byte at the lower value address. Because the M68HC11 is an 8-bit CPU, there are no special requirements for alignment of instructions or operands.

11.3 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 four-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.

11.4 Addressing modes

Six addressing modes; immediate, direct, extended, indexed, inherent, and relative, detailed in the following paragraphs, can be used to access memory. 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.

11.4.1 Immediate (IMM)

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 two, three, and four (if prebyte is required) byte immediate instructions. The effective address is the address of the byte following the instruction.

11.4.2 Direct (DIR)

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 two-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.

11.4.3 Extended (EXT)

In the extended addressing mode, the effective address of the argument is contained in two bytes following the opcode byte. These are three-byte instructions (or four-byte instructions if a prebyte is required). One or two bytes are needed for the opcode and two for the effective address.

11.4.4 Indexed (IND, X; IND, Y)

In the indexed addressing mode, an 8-bit unsigned offset contained in the instruction is added to the value contained in an index register (IX or IY) — the sum is the effective address. This addressing mode allows referencing any memory location in the 64Kbyte address space. These are two- to five-byte instructions, depending on whether or not a prebyte is required.

11.4.5 Inherent (INH)

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations that use only the index registers or accumulators, as well as control instructions with no arguments, are included in this addressing mode. These are one or two-byte instructions.

11.4.6 Relative (REL)

The relative addressing mode is used only for branch instructions. If the branch condition is true, an 8-bit signed offset included in the instruction is added to the contents of the program counter to form the effective branch address. Otherwise, control proceeds to the next instruction. These are usually two-byte instructions.

11.5 Instruction set

Refer to Table 11-2, which shows all the M68HC11 instructions in all possible addressing modes. For each instruction, the table shows the operand construction, the number of machine code bytes, and execution time in CPU E clock cycles.

Table 11-2 Instruction set (Sheet 1 of 6)

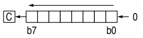
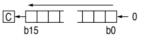
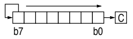
Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
ABA	Add accumulators	A + B ⇒ A	INH	1B	—	2	—	—	Δ	—	Δ	Δ	Δ	Δ
ABX	Add B to X	IX + (00:B) ⇒ IX	INH	3A	—	3	—	—	—	—	—	—	—	—
ABY	Add B to Y	IY + (00:B) ⇒ IY	INH	18 3A	—	4	—	—	—	—	—	—	—	—
ADCA (opr)	Add with carry to A	A + M + C ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	89 99 B9 A9 18 A9	ii dd hh ll ff ff	2 3 4 4 5	—	—	Δ	—	Δ	Δ	Δ	Δ
ADCB (opr)	Add with carry to B	B + M + C ⇒ B	B IMM B DIR B EXT B IND, X B IND, Y	C9 D9 F9 E9 18 E9	ii dd hh ll ff ff	2 3 4 4 5	—	—	Δ	—	Δ	Δ	Δ	Δ
ADDA (opr)	Add memory to A	A + M ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	8B 9B BB AB 18 AB	ii dd hh ll ff ff	2 3 4 4 5	—	—	Δ	—	Δ	Δ	Δ	Δ
ADDB (opr)	Add memory to B	B + M ⇒ B	B IMM B DIR B EXT B IND, X B IND, Y	CB DB FB EB 18 EB	ii dd hh ll ff ff	2 3 4 4 5	—	—	Δ	—	Δ	Δ	Δ	Δ
ADDD (opr)	Add 16-bit to D	D + (M:M+1) ⇒ D	IMM DIR EXT IND, X IND, Y	C3 D3 F3 E3 18 E3	jj kk dd hh ll ff ff	4 5 6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
ANDA (opr)	AND A with memory	A • M ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	84 94 B4 A4 18 A4	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
ANDB (opr)	AND B with memory	B • M ⇒ B	B IMM B DIR B EXT B IND, X B IND, Y	C4 D4 F4 E4 18 E4	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
ASL (opr)	Arithmetic shift left		EXT IND, X IND, Y	78 68 18 68	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
ASLA	Arithmetic shift left A		A INH	48	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLB	Arithmetic shift left B		B INH	58	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASLD	Arithmetic shift left D		INH	05	—	3	—	—	—	—	Δ	Δ	Δ	Δ
ASR	Arithmetic shift right		EXT IND, X IND, Y	77 67 18 67	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
ASRA	Arithmetic shift right A		A INH	47	—	2	—	—	—	—	Δ	Δ	Δ	Δ
ASRB	Arithmetic shift right B		B INH	57	—	2	—	—	—	—	Δ	Δ	Δ	Δ
BCC (rel)	Branch if carry clear	C = 0 ?	REL	24	rr	3	—	—	—	—	—	—	—	—
BCLR (opr) (msk)	Clear bit(s)	$M \cdot (\overline{mm}) \Rightarrow M$	DIR IND, X IND, Y	15 1D 18 1D	dd mm ff mm ff mm	6 7 8	—	—	—	—	Δ	Δ	0	—
BCS (rel)	Branch if carry set	C = 1 ?	REL	25	rr	3	—	—	—	—	—	—	—	—
BEQ (rel)	Branch if equal to zero	Z = 1 ?	REL	27	rr	3	—	—	—	—	—	—	—	—
BGE (rel)	Branch if ≥ zero	N ⊕ V = 0 ?	REL	2C	rr	3	—	—	—	—	—	—	—	—
BGT (rel)	Branch if > zero	Z + (N ⊕ V) = 0 ?	REL	2E	rr	3	—	—	—	—	—	—	—	—
BHI (rel)	Branch if higher	C + Z = 0 ?	REL	22	rr	3	—	—	—	—	—	—	—	—

Table 11-2 Instruction set (Sheet 2 of 6)

Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
BHS (rel)	Branch if higher or same	C = 0 ?	REL	24	rr	3	—	—	—	—	—	—	—	—
BITA (opr)	Bit(s) test A with memory	A • M	A IMM A DIR A EXT A IND, X A IND, Y	85 95 B5 A5 18 A5	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
BITB (opr)	Bit(s) test B with memory	B • M	B IMM B DIR B EXT B IND, X B IND, Y	C5 D5 F5 E5 18 E5	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
BLE (rel)	Branch if ≤ zero	Z + (N ⊕ V) = 1 ?	REL	2F	rr	3	—	—	—	—	—	—	—	—
BLO (rel)	Branch if lower	C = 1 ?	REL	25	rr	3	—	—	—	—	—	—	—	—
BLS (rel)	Branch if lower or same	C + Z = 1 ?	REL	23	rr	3	—	—	—	—	—	—	—	—
BLT (rel)	Branch if < zero	N ⊕ V = 1 ?	REL	2D	rr	3	—	—	—	—	—	—	—	—
BMI (rel)	Branch if minus	N = 1 ?	REL	2B	rr	3	—	—	—	—	—	—	—	—
BNE (rel)	Branch if ≠ zero	Z = 0 ?	REL	26	rr	3	—	—	—	—	—	—	—	—
BPL(rel)	Branch if plus	N = 0 ?	REL	2A	rr	3	—	—	—	—	—	—	—	—
BRA (rel)	Branch always	1 = 1 ?	REL	20	rr	3	—	—	—	—	—	—	—	—
BRCLR(opr) (msk) (rel)	Branch if bit(s) clear	M • mm = 0 ?	DIR IND, X IND, Y	13 1F 18 1F	dd mm rr ff mm rr ff mm rr	6 7 8	—	—	—	—	—	—	—	—
BRN (rel)	Branch never	1 = 0 ?	REL	21	rr	3	—	—	—	—	—	—	—	—
BRSET(opr) (msk) (rel)	Branch if bit(s) set	$\overline{M}$ • mm = 0 ?	DIR IND, X IND, Y	12 1E 18 1E	dd mm rr ff mm rr ff mm rr	6 7 8	—	—	—	—	—	—	—	—
BSET (opr) (msk)	Set bit(s)	M + mm ⇒ M	DIR IND, X IND, Y	14 1C 18 1C	dd mm ff mm ff mm	6 7 8	—	—	—	—	Δ	Δ	0	—
BSR (rel)	Branch to subroutine	see Figure 11-2	REL	8D	rr	6	—	—	—	—	—	—	—	—
BVC (rel)	Branch if overflow clear	V = 0 ?	REL	28	rr	3	—	—	—	—	—	—	—	—
BVS (rel)	Branch if overflow set	V = 1 ?	REL	29	rr	3	—	—	—	—	—	—	—	—
CBA	Compare A with B	A – B	INH	11	—	2	—	—	—	—	Δ	Δ	Δ	Δ
CLC	Clear carry bit	0 ⇒ C	INH	0C	—	2	—	—	—	—	—	—	—	0
CLI	Clear interrupt mask	0 ⇒ I	INH	0E	—	2	—	—	—	0	—	—	—	—
CLR (opr)	Clear memory byte	0 ⇒ M	DIR IND, X IND, Y	7F 6F 18 6F	hh ll ff ff	6 6 7	—	—	—	—	0	1	0	0
CLRA	Clear accumulator A	0 ⇒ A	A INH	4F	—	2	—	—	—	—	0	1	0	0
CLRB	Clear accumulator B	0 ⇒ B	B INH	5F	—	2	—	—	—	—	0	1	0	0
CLV	Clear overflow flag	0 ⇒ V	INH	0A	—	2	—	—	—	—	—	—	0	—
CMPA (opr)	Compare A with memory	A – M	A IMM A DIR A EXT A IND, X A IND, Y	81 91 B1 A1 18 A1	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
CMPB (opr)	Compare B with memory	B – M	B IMM B DIR B EXT B IND, X B IND, Y	C1 D1 F1 E1 18 E1	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
COM (opr)	Ones complement memory byte	$FF – M ⇒ M	EXT IND, X IND, Y	73 63 18 63	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	0	1
COMA	Ones complement A	$FF – A ⇒ A	A INH	43	—	2	—	—	—	—	Δ	Δ	0	1
COMB	Ones complement B	$FF – B ⇒ B	B INH	53	—	2	—	—	—	—	Δ	Δ	0	1

Table 11-2 Instruction set (Sheet 3 of 6)

Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
CPD (opr)	Compare D with memory (16-bit)	D – (M:M+1)	IMM DIR EXT IND, X IND, Y	1A 83 1A 93 1A B3 1A A3 CD A3	jj kk dd hh ll ff ff	5 6 7 7 7	–	–	–	–	Δ	Δ	Δ	Δ
CPX (opr)	Compare IX with memory (16-bit)	IX – (M:M+1)	IMM DIR EXT IND, X IND, Y	8C 9C BC AC CD AC	jj kk dd hh ll ff ff	4 5 6 6 7	–	–	–	–	Δ	Δ	Δ	Δ
CPY (opr)	Compare IY with memory (16-bit)	IY – (M:M+1)	IMM DIR EXT IND, X IND, Y	18 8C 18 9C 18 BC 1A AC 18 AC	jj kk dd hh ll ff ff	5 6 7 7 7	–	–	–	–	Δ	Δ	Δ	Δ
DAA	Decimal adjust A	adjust sum to BCD	INH	19	–	2	–	–	–	–	Δ	Δ	?	Δ
DEC (opr)	Decrement memory byte	M – 1 ⇒ M	EXT IND, X IND, Y	7A 6A 18 6A	hh ll ff ff	6 6 7	–	–	–	–	Δ	Δ	Δ	–
DECA	Decrement accumulator A	A – 1 ⇒ A	A INH	4A	–	2	–	–	–	–	Δ	Δ	Δ	–
DECB	Decrement accumulator B	B – 1 ⇒ B	B INH	5A	–	2	–	–	–	–	Δ	Δ	Δ	–
DES	Decrement stack pointer	SP – 1 ⇒ SP	INH	34	–	3	–	–	–	–	–	–	–	–
DEX	Decrement index register X	IX – 1 ⇒ IX	INH	09	–	3	–	–	–	–	–	Δ	–	–
DEY	Decrement index register Y	IY – 1 ⇒ IY	INH	18 09	–	4	–	–	–	–	–	Δ	–	–
EORA (opr)	Exclusive OR A with memory	A ⊕ M ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	88 98 B8 A8 18 A8	ii dd hh ll ff ff	2 3 4 4 5	–	–	–	–	Δ	Δ	0	–
EORB (opr)	Exclusive OR B with memory	B ⊕ M ⇒ A	B IMM B DIR B EXT B IND, X B IND, Y	C8 D8 F8 E8 18 E8	ii dd hh ll ff ff	2 3 4 4 5	–	–	–	–	Δ	Δ	0	–
FDIV	Fractional divide, 16 by 16	D / IX ⇒ IX; r ⇒ D	INH	03	–	41	–	–	–	–	–	Δ	Δ	Δ
IDIV	Integer divide, 16 by 16	D / IX ⇒ IX; r ⇒ D	INH	02	–	41	–	–	–	–	–	Δ	0	Δ
INC (opr)	Increment memory byte	M + 1 ⇒ M	EXT IND, X IND, Y	7C 6C 18 6C	hh ll ff ff	6 6 7	–	–	–	–	Δ	Δ	Δ	–
INCA	Increment accumulator A	A + 1 ⇒ A	A INH	4C	–	2	–	–	–	–	Δ	Δ	Δ	–
INCB	Increment accumulator B	B + 1 ⇒ B	B INH	5C	–	2	–	–	–	–	Δ	Δ	Δ	–
INS	Increment stack pointer	SP + 1 ⇒ SP	INH	31	–	3	–	–	–	–	–	–	–	–
INX	Increment index register X	IX + 1 ⇒ IX	INH	08	–	3	–	–	–	–	–	Δ	–	–
INY	Increment index register Y	IY + 1 ⇒ IY	INH	18 08	–	4	–	–	–	–	–	Δ	–	–
JMP (opr)	Jump	see Figure 11-2	EXT IND, X IND, Y	7E 6E 18 6E	hh ll ff ff	3 3 4	–	–	–	–	–	–	–	–
JSR (opr)	Jump to subroutine	see Figure 11-2	DIR EXT IND, X IND, Y	9D BD AD 18 AD	dd hh ll ff ff	5 6 6 7	–	–	–	–	–	–	–	–
LDAA (opr)	Load accumulator A	M ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	86 96 B6 A6 18 A6	ii dd hh ll ff ff	2 3 4 4 5	–	–	–	–	Δ	Δ	0	–

Table 11-2 Instruction set (Sheet 4 of 6)

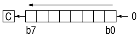
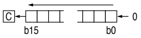
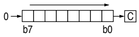
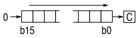
Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
LDAB (opr)	Load accumulator B	M ⇒ B	B IMM B DIR B EXT B IND, X B IND, Y	C6 D6 F6 E6 18 E6	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	0	—
LDD (opr)	Load double accumulator D	M ⇒ A; M+1 ⇒ B	IMM DIR EXT IND, X IND, Y	CC DC FC EC 18 EC	jj kk dd hh ll ff ff	3 4 5 5 6	—	—	—	—	Δ	Δ	0	—
LDS (opr)	Load stack pointer	M:M+1 ⇒ SP	IMM DIR EXT IND, X IND, Y	8E 9E BE AE 18 AE	jj kk dd hh ll ff ff	3 4 5 5 6	—	—	—	—	Δ	Δ	0	—
LDX (opr)	Load index register X	M:M+1 ⇒ IX	IMM DIR EXT IND, X IND, Y	CE DE FE EE CD EE	jj kk dd hh ll ff ff	3 4 5 5 6	—	—	—	—	Δ	Δ	0	—
LDY (opr)	Load index register Y	M:M+1 ⇒ IY	IMM DIR EXT IND, X IND, Y	18 CE 18 DE 18 FE 1A EE 18 EE	jj kk dd hh ll ff ff	4 5 6 6 6	—	—	—	—	Δ	Δ	0	—
LSL (opr)	Logical shift left		EXT IND, X IND, Y	78 68 18 68	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
LSLA	Logical shift left A		A INH	48	—	2	—	—	—	—	Δ	Δ	Δ	Δ
LSLB	Logical shift Left B		B INH	58	—	2	—	—	—	—	Δ	Δ	Δ	Δ
LSLD	Logical shift left D		INH	05	—	3	—	—	—	—	Δ	Δ	Δ	Δ
LSR (opr)	Logical shift right		EXT IND, X IND, Y	74 64 18 64	hh ll ff ff	6 6 7	—	—	—	—	0	Δ	Δ	Δ
LSRA	Logical shift right A		A INH	44	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRB	Logical shift right B		B INH	54	—	2	—	—	—	—	0	Δ	Δ	Δ
LSRD	Logical shift right D		INH	04	—	3	—	—	—	—	0	Δ	Δ	Δ
MUL	Multiply, 8 x 8	A * B ⇒ D	INH	3D	—	10	—	—	—	—	—	—	—	Δ
NEG (opr)	Twos 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	Twos complement A	0 – A ⇒ A	A INH	40	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NEGB	Twos complement B	0 – B ⇒ B	B INH	50	—	2	—	—	—	—	Δ	Δ	Δ	Δ
NOP	No operation	no operation	INH	01	—	2	—	—	—	—	—	—	—	—
ORAA	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	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 ⇒ Stack; SP = SP–1	A INH	36	—	3	—	—	—	—	—	—	—	—
PSHB	Push B onto stack	B ⇒ Stack; SP = SP–1	B INH	37	—	3	—	—	—	—	—	—	—	—
PSHX	Push IX onto stack (low first)	IX ⇒ Stack; SP = SP–2	INH	3C	—	4	—	—	—	—	—	—	—	—
PSHY	Push IY onto stack (low first)	IY ⇒ Stack; SP = SP–2	INH	18 3C	—	5	—	—	—	—	—	—	—	—

Table 11-2 Instruction set (Sheet 5 of 6)

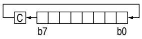
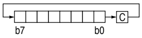
Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
PULA	Pull A from stack	SP = SP+1; Stack ⇒ A	A INH	32	–	4	–	–	–	–	–	–	–	–
PULB	Pull B from stack	SP = SP+1; Stack ⇒ B	B INH	33	–	4	–	–	–	–	–	–	–	–
PULX	Pull IX from stack (high first)	SP = SP+2; Stack ⇒ IX	INH	38	–	5	–	–	–	–	–	–	–	–
PULY	Pull IY from stack (high first)	SP = SP+2; Stack ⇒ IY	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 11-2	INH	3B	–	12	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ
RTS	Return from subroutine	see Figure 11-2	INH	39	–	5	–	–	–	–	–	–	–	–
SBA	Subtract B from A	A – B ⇒ A	INH	10	–	2	–	–	–	–	Δ	Δ	Δ	Δ
SBCA (opr)	Subtract with carry from A	A – M – C ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	82 92 B2 A2 18 A2	ii dd hh ll ff ff	2 3 4 4 5	–	–	–	–	Δ	Δ	Δ	Δ
SBCB (opr)	Subtract with carry from B	B – M – C ⇒ B	B IMM B DIR B EXT B IND, X B IND, Y	C2 D2 F2 E2 18 E2	ii dd hh ll ff ff	2 3 4 4 5	–	–	–	–	Δ	Δ	Δ	Δ
SEC	Set carry	1 ⇒ C	INH	0D	–	2	–	–	–	–	–	–	–	1
SEI	Set interrupt mask	1 ⇒ I	INH	0F	–	2	–	–	–	1	–	–	–	–
SEV	Set overflow flag	1 ⇒ V	INH	0B	–	2	–	–	–	–	–	–	1	–
STAA (opr)	Store accumulator A	A ⇒ M	A DIR A EXT A IND, X A IND, Y	97 B7 A7 18 A7	dd hh ll ff ff	3 4 4 5	–	–	–	–	Δ	Δ	0	–
STAB (opr)	Store accumulator B	B ⇒ M	B DIR B EXT B IND, X B IND, Y	D7 F7 E7 18 E7	dd hh ll ff ff	3 4 4 5	–	–	–	–	Δ	Δ	0	–
STD (opr)	Store accumulator D	A ⇒ M; B ⇒ M+1	DIR EXT IND, X IND, Y	DD FD ED 18 ED	dd hh ll ff ff	4 5 5 6	–	–	–	–	Δ	Δ	0	–
STOP	Stop internal clocks	–	INH	CF	–	2	–	–	–	–	–	–	–	–
STS (opr)	Store stack pointer	SP ⇒ M:M+1	DIR EXT IND, X IND, Y	9F BF AF 18 AF	dd hh ll ff ff	4 5 5 6	–	–	–	–	Δ	Δ	0	–
STX (opr)	Store index register X	IX ⇒ M:M+1	DIR EXT IND, X IND, Y	DF FF EF CD EF	dd hh ll ff ff	4 5 5 6	–	–	–	–	Δ	Δ	0	–
STY (opr)	Store index register Y	IY ⇒ M:M+1	DIR EXT IND, X IND, Y	18 DF 18 FF 1A EF 18 EF	dd hh ll ff ff	5 6 6 6	–	–	–	–	Δ	Δ	0	–

Table 11-2 Instruction set (Sheet 6 of 6)

Mnemonic	Operation	Description	Addressing mode	Instruction			Condition codes							
				Opcode	Operand	Cycles	S	X	H	I	N	Z	V	C
SUBA (opr)	Subtract memory from A	A – M ⇒ A	A IMM A DIR A EXT A IND, X A IND, Y	80 90 B0 A0 18 A0	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
SUBB (opr)	Subtract memory from B	B – M ⇒ B	B IMM B DIR B EXT B IND, X B IND, Y	C0 D0 F0 E0 18 E0	ii dd hh ll ff ff	2 3 4 4 5	—	—	—	—	Δ	Δ	Δ	Δ
SUBD (opr)	Subtract memory from D	D – M:M+1 ⇒ D	IMM DIR EXT IND, X IND, Y	83 93 B3 A3 18 A3	jj kk dd hh ll ff ff	4 5 6 6 7	—	—	—	—	Δ	Δ	Δ	Δ
SWI	Software interrupt	see Figure 11-2	INH	3F	—	14	—	—	—	1	—	—	—	—
TAB	Transfer A to B	A ⇒ B	INH	16	—	2	—	—	—	—	Δ	Δ	0	—
TAP	Transfer A to CC register	A ⇒ CCR	INH	06	—	2	Δ	↓	Δ	Δ	Δ	Δ	Δ	Δ
TBA	Transfer B to A	B ⇒ A	INH	17	—	2	—	—	—	—	Δ	Δ	0	—
TEST	Test (only in test modes)	address bus increments	INH	00	—	†	—	—	—	—	—	—	—	—
TPA	Transfer CC register to A	CCR ⇒ A	INH	07	—	2	—	—	—	—	—	—	—	—
TST (opr)	Test for zero or minus	M – 0	EXT IND, X IND, Y	7D 6D 18 6D	hh ll ff ff	6 6 7	—	—	—	—	Δ	Δ	0	0
TSTA	Test A for zero or minus	A – 0	A INH	4D	—	2	—	—	—	—	Δ	Δ	0	0
TSTB	Test B for zero or minus	B – 0	B INH	5D	—	2	—	—	—	—	Δ	Δ	0	0
TSX	Transfer stack pointer to X	SP + 1 ⇒ IX	INH	30	—	3	—	—	—	—	—	—	—	—
TSY	Transfer stack pointer to Y	SP + 1 ⇒ IY	INH	18 30	—	4	—	—	—	—	—	—	—	—
TXS	Transfer X to stack pointer	IX – 1 ⇒ SP	INH	35	—	3	—	—	—	—	—	—	—	—
TYS	Transfer Y to stack pointer	IY – 1 ⇒ SP	INH	18 35	—	4	—	—	—	—	—	—	—	—
WAI	Wait for interrupt	stack registers & WAIT	INH	3E	—	‡	—	—	—	—	—	—	—	—
XGDX	Exchange D with X	IX ⇒ D; D ⇒ IX	INH	8F	—	3	—	—	—	—	—	—	—	—
XGDY	Exchange D with Y	IY ⇒ D; D ⇒ IY	INH	18 8F	—	4	—	—	—	—	—	—	—	—

Operators

⇒	Is transferred to
•	Boolean AND
+	Arithmetic addition, except where used as an inclusive-OR symbol in Boolean formulae
⊕	Exclusive-OR
*	Multiply
:	Concatenation
–	Arithmetic subtraction, or negation symbol (Twos complement)

Operands

dd	8-bit direct address ($0000–$00FF); the high byte is assumed to be zero
ff	8-bit positive offset ($00 to $FF (0 to 256)) is added to the contents of the index register
hh	High order byte of 16-bit extended address
ii	One byte of immediate data
jj	High order byte of 16-bit immediate data
kk	Low order byte of 16-bit immediate data
ll	Low order byte of 16-bit extended address
mm	8-bit mask (set bits to be affected)
rr	Signed relative offset ($80 to $7F (–128 to +127)); offset is relative to the address following the offset byte

Cycles

†	Infinite, or until reset occurs
‡	12 cycles are used, beginning with the opcode fetch. A wait state is entered, which remains in effect for an integer number of MPU E clock cycles (n) until an interrupt is recognised. Finally, two additional cycles are used to fetch the appropriate interrupt vector. (14 + n, total).

Condition Codes

—	Bit not changed
0	Bit always cleared
1	Bit always set
Δ	Bit set or cleared, depending on the operation
↓	Bit can be cleared, but cannot become set
?	Not defined

A

ELECTRICAL SPECIFICATIONS (STANDARD)

This section contains the electrical specifications and associated timing information for the standard supply voltage (V_{DD} = 5V ± 10%) MC68HC11PH8 variants.

1.1 Maximum ratings

Rating	Symbol	Value	Unit
Supply voltage [1]	V_{DD}	– 0.3 to +7.0	V
Input voltage [1]	V_{in}	– 0.3 to +7.0	V
Operating temperature range – MC68HC11PH8, *MC68HC711PH8*	T_A	T_L to T_H −40 to +85	°C
Storage temperature range	T_{stg}	– 55 to +150	°C
Current drain per pin [2] – not VDD, VSS, VDD AD, VSS AD, VRH or VRL	I_D	25	mA

[1] All voltages are with respect to V_{SS}.

[2] Maximum current drain per pin is for one pin at a time, observing maximum power dissipation limits.

Note: This device contains circuitry designed to protect against damage due to high electrostatic voltages or electric fields. However, it is recommended that normal precautions be taken to avoid the application of any voltages higher than those given in the maximum ratings table to this high impedance circuit. For maximum reliability all unused inputs should be tied to either V_{SS} or V_{DD}.

1.2 Thermal characteristics and power considerations

The average chip junction temperature, T_J, in degrees Celsius can be obtained from the following equation:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \qquad [1]$$

where:

T_A = Ambient temperature (°C)

θ_{JA} = Package thermal resistance, junction-to-ambient (°C/W)

P_D = Total power dissipation = $P_{INT} + P_{I/O}$ (W)

P_{INT} = Internal chip power = $I_{DD} \bullet V_{DD}$ (W)

$P_{I/O}$ = Power dissipation on input and output pins (user determined)

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is:

$$P_D = \frac{K}{T_J + 273} \qquad [2]$$

Solving equations [1] and [2] for K gives:

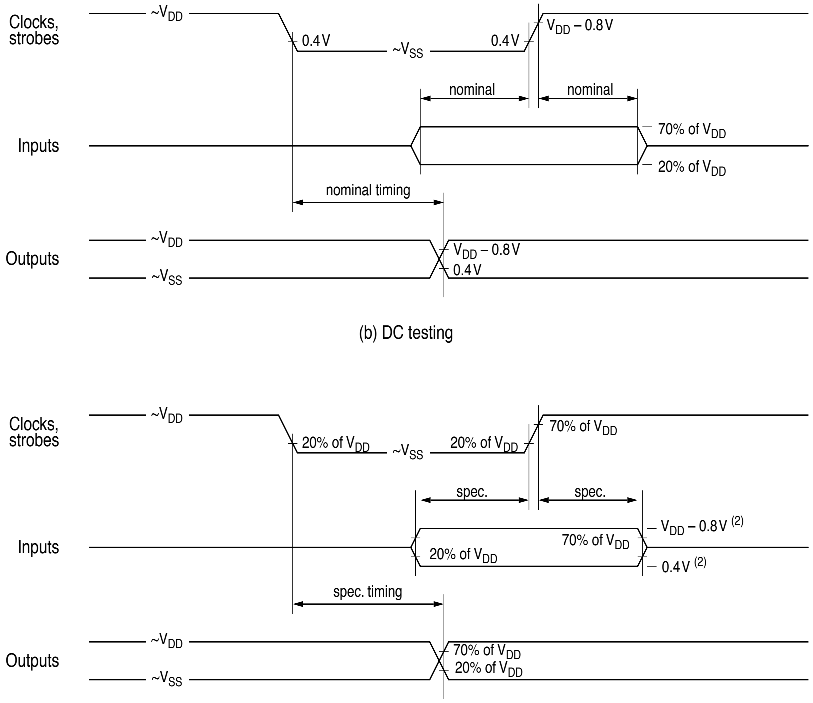
$$K = P_D \bullet (T_A + 273) + \theta_{JA} \bullet {P_D}^2 \qquad [3]$$

where K is a constant for a particular part. K can be determined by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained for any value of T_A, by solving the above equations. The package thermal characteristics are shown below:

Characteristics	Symbol	Value	Unit
Thermal resistance – 84-pin PLCC package *– 84-pin CERQUAD package (EPROM)* – 112-pin QFP package	θ_{JA}	 50 *50* TBD	°C/W

12

1.3 Test methods



(b) DC testing

(c) AC testing

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 V and V_{DD} – 0.8 V; timing measurements are taken at the 20% and 70% of V_{DD} points.

Figure A-1 Test methods

1.4 DC electrical characteristics

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H, unless otherwise noted)

Characteristic	Symbol	Min.	Max.	Unit
Output voltage[1] (I_{LOAD} = ± 10 µA): All outputs except XTAL All outputs except XTAL, $\overline{\text{RESET}}$ & MODA	 V_{OL} V_{OH}	 – V_{DD} – 0.1	 0.1 –	 V V
Output high voltage[1] (I_{LOAD} = –0.8 mA, V_{DD} = 4.5 V): All outputs except XTAL, $\overline{\text{RESET}}$ & MODA	 V_{OH}	 V_{DD} – 0.8	 –	 V
Output low voltage (I_{LOAD} = +1.6 mA): All outputs except XTAL	 V_{OL}	 –	 0.4	 V
Input high voltage: All inputs except $\overline{\text{RESET}}$ $\overline{\text{RESET}}$	V_{IH}	 0.7 V_{DD} 0.8 V_{DD}	 V_{DD} + 0.3 V_{DD} + 0.3	V
Input low voltage – all inputs	V_{IL}	V_{SS} – 0.3	0.2 V_{DD}	V
I/O ports three-state leakage (V_{IN} = V_{IH} or V_{IL})[2]: Ports A, B, C, D, F, G, H, MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$	 I_{OZ}	 –	 ±10	 µA
Input leakage[2] (V_{IN} = V_{DD} or V_{SS}): MODB/VSTBY $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$ (ROM parts) *$\overline{\text{XIRQ}}$ (EPROM parts)*	I_{IN}	 – – –	 ±10 ±1 ±10	µA
Input current with pull-up resistors (V_{IN} = V_{IL}): Ports B, C, F, G, H	 I_{IPR}	 20	 100	 µA
RAM stand-by voltage (power down)	V_{SB}	2.0	V_{DD}	V
RAM stand-by current (power down)	I_{SB}	–	10	µA
Input capacitance: Port E, $\overline{\text{IRQ}}$, $\overline{\text{XIRQ}}$, EXTAL Ports A, B, C, D, F, G, H, MODA/$\overline{\text{LIR}}$, $\overline{\text{RESET}}$	C_{IN}	 – –	 8 12	pF
Output load capacitance: All outputs except PD[4:1], PG[4:1], XTAL, MODA/$\overline{\text{LIR}}$ PD[4:1], PG[4:1]	C_L	 – –	 90 200	pF

[1] V_{OH} specification for $\overline{\text{RESET}}$ and MODA is not applicable as they are open-drain pins.
V_{OH} specification is not applicable to port C, port D and port G[5:0] in wired-OR mode.

[2] Refer to A/D specification for the leakage current value for port E.

A.4.1 DC electrical characteristics — modes of operation

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic	Symbol	6kHz	2MHz	3MHz	4MHz	Unit
Maximum total supply current (including PLL)(1):	I_{DD}					
RUN: Single chip mode		TBD	27	32	40	mA
RUN: Expanded mode		TBD	35	42	50	mA
WAIT: Single chip mode(2)		500	500	500	500	µA
STOP: Single chip mode		50	50	50	50	µA
Maximum power dissipation: Single chip mode	P_D	TBD	149	176	220	mW
Maximum power dissipation: Expanded mode		TBD	193	231	275	mW

(1) All current measurements taken with suitable decoupling capacitors across the power supply to suppress the transient switching currents inherent in CMOS designs.
EXTAL is driven with a square wave, with t_{CYC} = 167ms for 6kHz devices; 500/333/250ns for 2/3/4MHz devices.
VIL ≤ 0.2V; VIH ≥ VDD – 0.2V; no DC loads
WAIT: all peripheral functions shut down STOP: all clocks stopped

(2) WAIT values in the 6 kHz column obtained by using an external clock of 32 kHz and the PLL low-power WAIT mode (WEN = 1).

12

1.5 Control timing

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H)

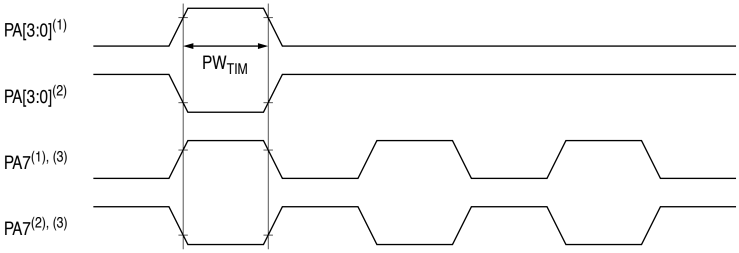
Characteristic [1]	Symbol	2.0 MHz		3.0 MHz		4.0 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
Frequency of operation	f_{OP}	0	2.0	0	3.0	0	4.0	MHz
E clock period	t_{CYC}	500	—	333	—	250	—	ns
Crystal frequency	f_{XTAL}	—	8.0	—	12.0	—	16.0	MHz
External oscillator frequency	$4f_{OP}$	0	8.0	0	12.0	0	16.0	MHz
Processor control set-up time (t_{PCSU} = t_{CYC}/4 + 50 ns)	t_{PCSU}	175	—	133	—	112	—	ns
Reset input pulse width [2]	PW_{RSTL} [3] PW_{RSTL} [4]	16 1	— —	16 1	— —	16 1	— —	t_{CYC}
Mode programming set-up time	t_{MPS}	2	—	2	—	2	—	t_{CYC}
Mode programming hold time	t_{MPH}	10	—	10	—	10	—	ns
Interrupt pulse width ($\overline{IRQ}$ edge sensitive mode)	PW_{IRQ}	t_{CYC} +20	—	t_{CYC} +20	—	t_{CYC} +20	—	ns
Timer pulse width (Input capture and pulse accumulator inputs)	PW_{TIM}	t_{CYC} +20	—	t_{CYC} +20	—	t_{CYC} +20	—	ns
WAIT recovery start-up time	t_{WRS}	—	4	—	4	—	4	t_{CYC}
Clock monitor reset	f_{CMON}	10	200	10	200	10	200	kHz

[1] All timing is given with respect to 20% and 70% of V_{DD}, unless otherwise noted.

[2] Reset is recognized during the first clock cycle it is held low. Internal circuitry then drives the pin low for eight clock cycles, releases the pin and samples the pin level four cycles later to determine the source of the interrupt. (See Section 10.)

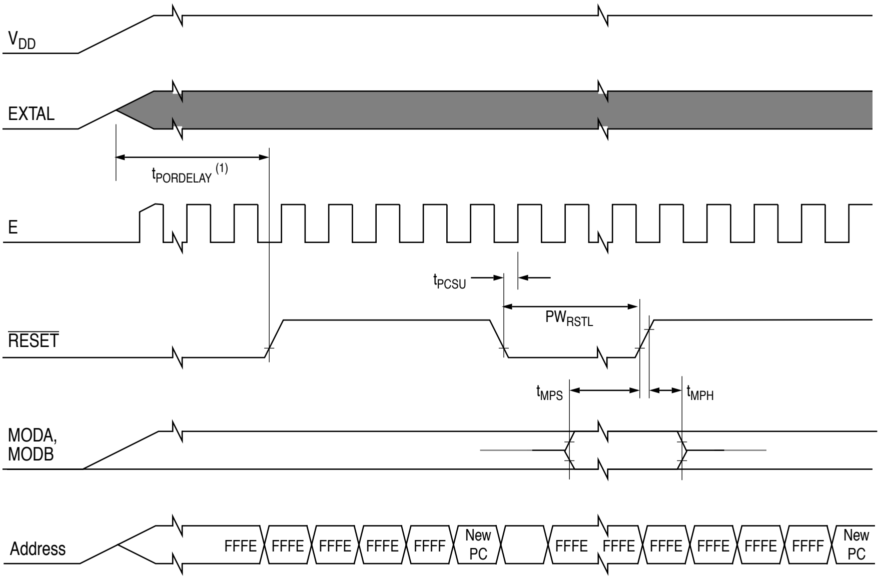
[3] To guarantee an external reset vector.

[4] This is the minimum input time; it can be pre-empted by an internal reset.



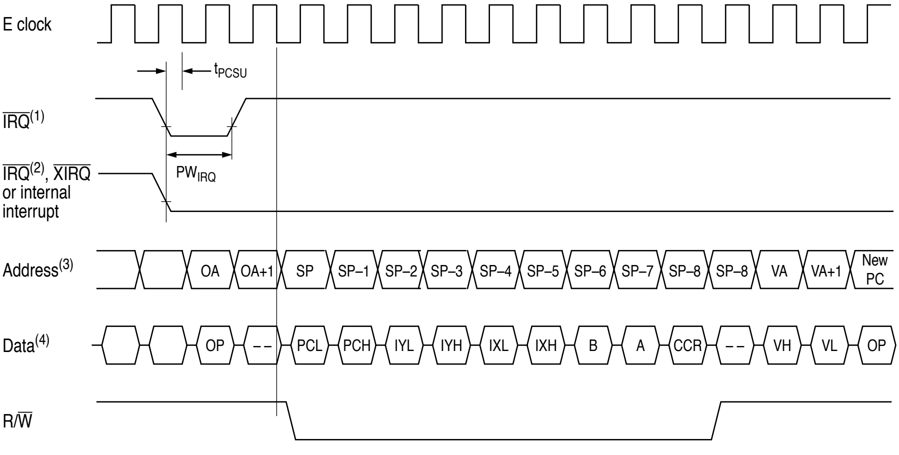
Notes
(1) Rising edge sensitive input.
(2) Falling edge sensitive input.
(3) Maximum pulse accumulator clocking rate is E clock frequency divided by two (E/2).

Figure A-2 Timer inputs



(1) $t_{PORDELAY}$ = 4064 t_{CYC} (or 128 t_{CYC} depending on mask option - MC68HC11PH8 only)

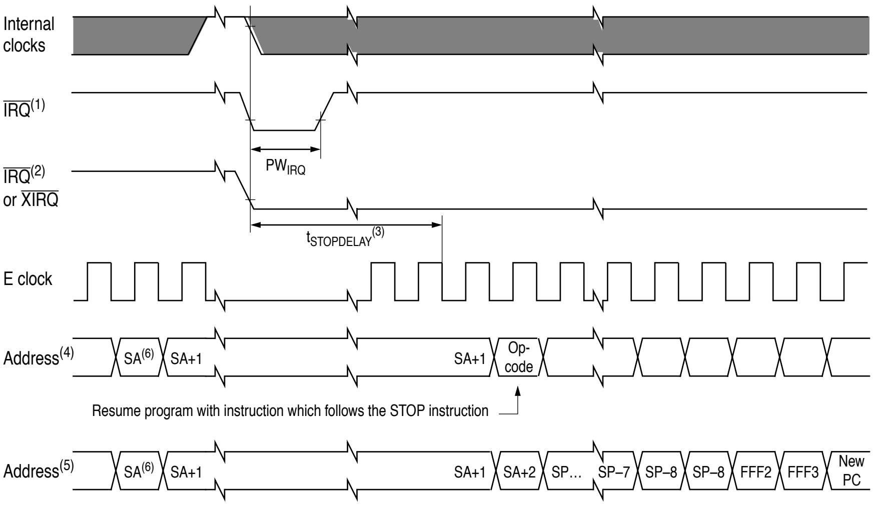
Figure A-3 Reset timing



Notes:
(1) Edge sensitive $\overline{IRQ}$ pin (IRQE = 1).
(2) Level sensitive $\overline{IRQ}$ pin (IRQE = 0).
(3) Where OA = Opcode address and VA = Vector address.
(4) Where OP = Opcode, VH = Vector (MSB) and VL = Vector (LSB).

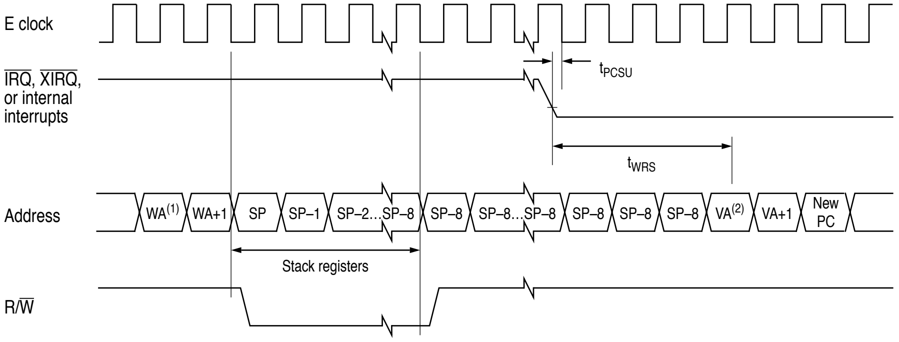
Figure A-4 Interrupt timing

12



Notes:
(1) Edge sensitive $\overline{\text{IRQ}}$ pin (IRQE = 1).
(2) Level sensitive $\overline{\text{IRQ}}$ pin (IRQE = 0).
(3) If DLY = 1: $t_{STOPDELAY}$ = 4064 t_{CYC} (or 128 t_{CYC} depending on mask option - MC68HC11PH8 only)
If DLY = 0: $t_{STOPDELAY}$ = 4 t_{CYC}
(4) $\overline{\text{XIRQ}}$ with X-bit in CCR = 1.
(5) $\overline{\text{IRQ}}$ (or $\overline{\text{XIRQ}}$, with X-bit = 0; in this case vector fetch will be $FFF4/5).
(6) SA = STOP address.

Figure A-5 STOP recovery timing



Notes:

$\overline{\text{RESET}}$ also causes recovery from WAIT. (1) WA = WAIT address. (2) VA = Vector address.

Figure A-6 WAIT recovery timing

A.5.1 Peripheral port timing

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H)

Characteristic [1]	Symbol	2.0 MHz		3.0 MHz		4.0 MHz		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	
Frequency of operation (E clock frequency)	f_{OP}	0	2.0	0	3.0	0	4.0	MHz
E clock period	t_{CYC}	500	—	333	—	250	—	ns
Peripheral data set-up time, all ports [2]	t_{PDSU}	100	—	100	—	100	—	ns
Peripheral data hold time, all ports [2]	t_{PDH}	50	—	50	—	50	—	ns
Delay time, peripheral data write	t_{PWD}							ns
MCU write to port A, B, G or H		—	200	—	200	—	200	
MCU write to port C, D or F ($t_{PWD} = t_{CYC}/4 + 100$ ns)		—	225	—	183	—	162	

(1) All timing is given with respect to 20% and 70% of V_{DD}, unless otherwise noted.

(2) Port C, D and G timing is valid for active drive (CWOM, DWOM, GWOM, WOMS and WOMS2 bits clear).

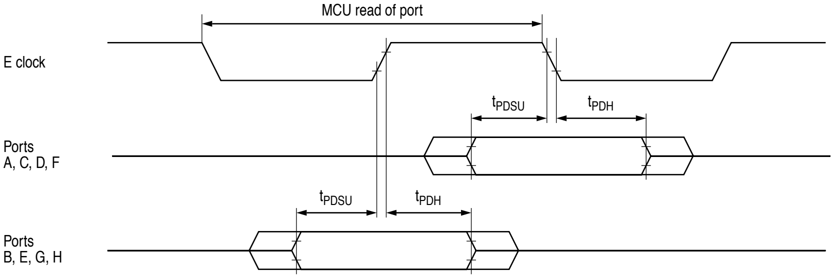


Figure A-7 Port read timing diagram

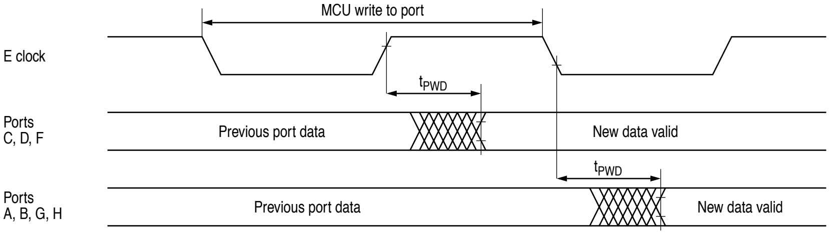


Figure A-8 Port write timing diagram

A.5.2 PLL control timing

(VDD = 5.0 Vdc ±10%, VSS = 0 Vdc, $T_A = T_L$ to T_H unless otherwise noted)

Characteristic	Symbol	Mask option 1			Mask option 2[1]			Units
		Min	Typical	Maximum	Min	Typical	Maximum	
PLL reference frequency	f_{REF}	25	32	50	50	614	2000	kHz
System frequency	f_{SYS}	dc	—	4	dc	—	4	MHz
PLL output frequency	f_{VCOOUT}	0.05	—	16	0.1	—	16	
External clock operation	f_{XTAL}	dc		16	dc		16	
Capacitor on pin XFC	C_{XFC}	—	47	—	—	47	—	nF
PLL stabilization time[2]	t_{PLLS}	—	20	TBD	—	10	TBD	ms
4XCLK stability[3][4]								
Short term	C_{STAB}	TBD	—	TBD	TBD	—	TBD	%
Long term		TBD	—	TBD	TBD	—	TBD	

(1) This mask option does not exist on the MC68HC711PH8, on which the PLL is optimized for use at 32 kHz.

(2) Assumes that stable VDDSYN is applied, that an external filter capacitor with a value of 47 nF is attached to the XFC pin, and that the crystal oscillator is stable. Stabilization time is measured from power-up to $\overline{\text{RESET}}$ release. This specification also applies to the period required for PLL stabilization after changing the X and Y frequency control bits in the synthesizer control register (SYNR) while PLL is running, and to the period required for the clock to stabilize after WAIT with WEN = 1.

(3) Short term stability is the average deviation from programmed frequency measured over a 2 µs interval at maximum f_{SYS}, Long term 4XCLK stability is the average deviation from programmed frequency measured over a 1 ms interval at maximum f_{SYS}. Stability is measured with a stable external clock applied — variation in crystal oscillator frequency is additive to this figure.

(4) This parameter is periodically sampled rather than 100% tested.

A.5.3 Analog-to-digital converter characteristics

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, 750 kHz ≤ E ≤ 4 MHz, unless otherwise noted)

Characteristic	Parameter	Min.	Absolute	2 MHz(1)	3 MHz(1)	4 MHz(1)	Unit
				Max.	Max.	Max.	
Resolution	Number of bits resolved by ADC	—	8	—	—	—	bits
Non-linearity	Maximum deviation from the ideal ADC transfer characteristics	—	—	±0.5	±1	±1	LSB
Zero error	Difference from the output of an ideal ADC for zero input voltage	—	—	±0.5	±1	±1	LSB
Full-scale error	Difference from the output of an ideal ADC for full-scale input voltage	—	—	±0.5	±1	±1	LSB
Total unadjusted error	Maximum sum of non-linearity, zero and full-scale errors	—	—	±0.5	±1.5	±1.5	LSB
Quantization error	Uncertainty due to converter resolution	—	—	±0.5	±0.5	±0.5	LSB
Absolute accuracy	Difference between the actual input voltage and the full-scale weighted equivalent of the binary output code, including all error sources	—	—	±1	±2	±2	LSB
Conversion range	Analog input voltage range	V_{RL}	—	V_{RH}	V_{RH}	V_{RH}	V
V_{RH}	Analog reference voltage (high) (2)	V_{RL}	—	V_{DD}+0.1	V_{DD}+0.1	V_{DD}+0.1	V
V_{RL}	Analog reference voltage (low) (2)	V_{SS}–0.1	—	V_{RH}	V_{RH}	V_{RH}	V
ΔV_R	Minimum difference between V_{RH} and V_{RL} (2)	3	—	—	—	—	V
Conversion time	Total time to perform a single A/D conversion: E clock Internal RC oscillator	— —	32 —	— t_{CYC}+32	— t_{CYC}+32	— t_{CYC}+32	t_{CYC} μs
Monotonicity	Conversion result never decreases with an increase in input voltage and has no missing codes	Guaranteed					
Zero input reading	Conversion result when $V_{IN} = V_{RL}$	$00	—	—	—	—	Hex
Full-scale reading	Conversion result when $V_{IN} = V_{RH}$	—	—	$FF	$FF	$FF	Hex
Sample acquisition time	Analog input acquisition sampling time: E clock Internal RC oscillator	— —	12 —	— 12	— 12	— 12	t_{CYC} μs
Sample/hold capacitance	Input capacitance (PE[0:7]) during sample	—	20 (typ)	—	—	—	pF
Input leakage	Input leakage on A/D pins: PE[0:7] V_{RL}, V_{RH}	— —	— —	400 1.0	400 1.0	400 1.0	nA μA

(1) For f_{OP} < 2 MHz, source impedances should be approximately 10 kΩ. For f_{OP} ≥ 2 MHz, source impedances should be in the range 5–10 kΩ. Source impedances greater than 10 kΩ have an adverse affect on A/D accuracy, because of input leakage.

(2) Performance verified down to ΔV_R = 2.5 V, however accuracy is tested and guaranteed at ΔV_R = 5 V ± 10%.

A.5.4 Serial peripheral interface timing

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic (1)		Symbol	2.0 MHz		3.0 MHz		4.0 MHz		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
	Operating frequency	Master	$f_{OP(M)}$	0	0.5	0	0.5	0	0.5	f_{OP}
		Slave	$f_{OP(S)}$	0	2.0	0	3.0	0	4.0	MHz
1	Cycle time	Master	$t_{CYC(M)}$	2.0	—	2.0	—	2.0	—	t_{CYC}
		Slave	$t_{CYC(S)}$	500	—	333	—	250	—	ns
2	Enable lead time (2)	Master	$t_{LEAD(M)}$	—	—	—	—	—	—	ns
		Slave	$t_{LEAD(S)}$	250	—	240	—	200	—	
3	Enable lag time (2)	Master	$t_{LAG(M)}$	—	—	—	—	—	—	ns
		Slave	$t_{LAG(S)}$	250	—	240	—	200	—	
4	Clock (SCK) high time	Master	$t_{W(SCKH)M}$	340	—	227	—	130	—	ns
		Slave	$t_{W(SCKH)S}$	190	—	127	—	85	—	
5	Clock (SCK) low time	Master	$t_{W(SCKL)M}$	340	—	227	—	130	—	ns
		Slave	$t_{W(SCKL)S}$	190	—	127	—	85	—	
6	Input data set-up time	Master	$t_{SU(M)}$	100	—	100	—	100	—	ns
		Slave	$t_{SU(S)}$	100	—	100	—	100	—	
7	Input data hold time	Master	$t_{H(M)}$	100	—	100	—	100	—	ns
		Slave	$t_{H(S)}$	100	—	100	—	100	—	
8	Access time (from high-z to data active)	Slave	t_A	0	120	0	120	0	120	ns
9	Disable time (hold time to high-z state)	Slave	t_{DIS}	—	300	—	300	—	300	ns
10	Data valid (after enable edge) (3)		$t_{V(S)}$	—	240	—	167	—	125	ns
11	Output data hold time (after enable edge)		t_{HO}	0	—	0	—	0	—	ns
12	Rise time (3)									
	SPI outputs (SCK, MOSI and MISO)		t_{RM}	—	100	—	100	—	100	ns
	SPI inputs (SCK, MOSI, MISO and $\overline{SS}$)		t_{RS}	—	2.0	—	2.0	—	2.0	µs
13	Fall time (3)									
	SPI outputs (SCK, MOSI and MISO)		t_{FM}	—	100	—	100	—	100	ns
	SPI inputs (SCK, MOSI, MISO and $\overline{SS}$)		t_{FS}	—	2.0	—	2.0	—	2.0	µs

(1) All timing is given with respect to 20% and 70% of V_{DD}, unless otherwise noted.

(2) Signal production depends on software.

(3) Assumes 200 pF load on all SPI pins.

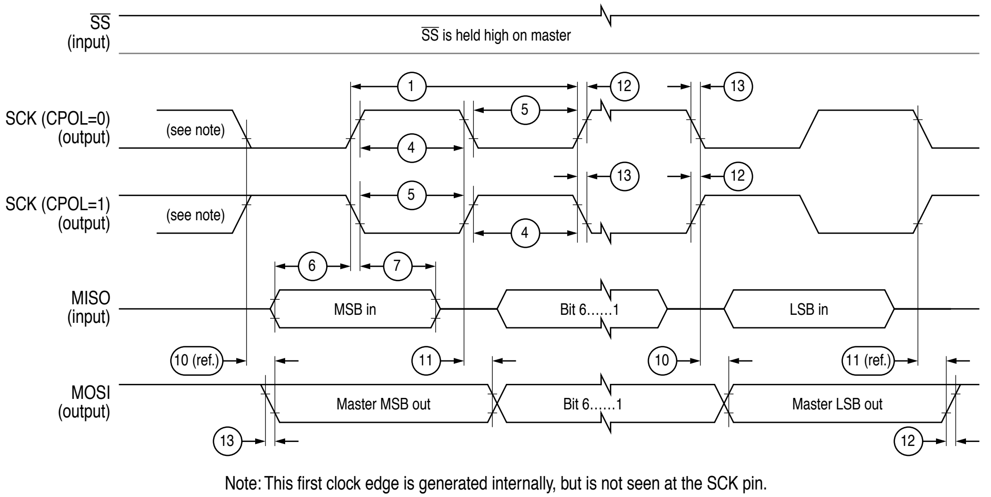


Figure A-9 SPI master timing (CPHA = 0)

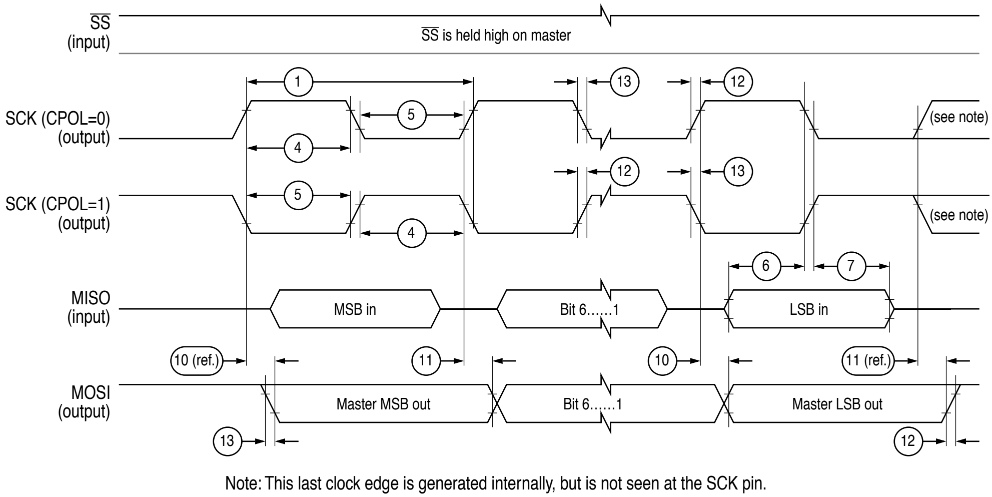
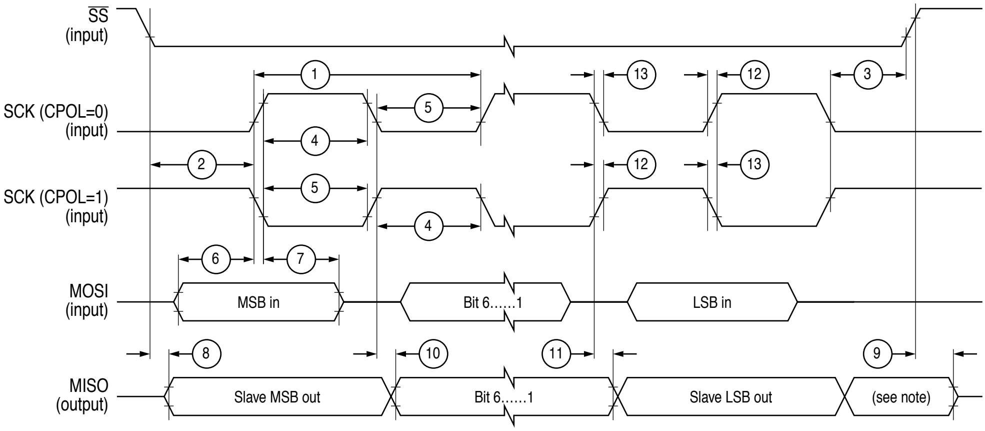
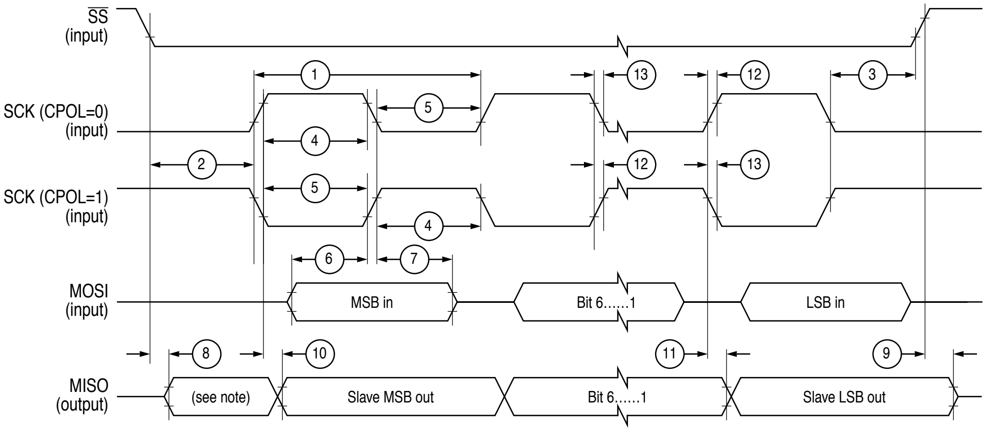


Figure A-10 SPI master timing (CPHA = 1)



Note: Not defined, but normally the MSB of character just received.

Figure A-11 SPI slave timing (CPHA = 0)



Note: Not defined, but normally the LSB of character last transmitted.

Figure A-12 SPI slave timing (CPHA = 1)

A.5.5 Non-multiplexed expansion bus timing

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Num	Characteristic [1]	Symbol	2.0 MHz		3.0 MHz		4.0 MHz		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
	Frequency of operation (E clock frequency)	f_{OP}	0	2.0	0	3.0	0	4.0	MHz
1	E clock period	t_{CYC}	500	—	333	—	250	—	ns
2	Pulse width, E low [2], [3]	PW_{EL}	230	—	147	—	105	—	ns
3	Pulse width, E high [2], [3]	PW_{EH}	225	—	142	—	100	—	ns
4A	E clock rise time	t_r	—	20	—	20	—	20	ns
4B	fall time	t_f	—	20	—	18	—	15	
9	Address hold time [3]	t_{AH}	53	—	32	—	21	—	ns
11	Address delay time [3]	t_{AD}	—	103	—	82	—	71	ns
12	Address valid to E rise time [3]	t_{AV}	127	—	65	—	34	—	ns
17	Read data set-up time	t_{DSR}	30	—	30	—	20	—	ns
18	Read data hold time	t_{DHR}	0	—	0	—	0	—	ns
19	Write data delay time	t_{DDW}	—	40	—	40	—	40	ns
21	Write data hold time [3]	t_{DHW}	63	—	42	—	31	—	ns
29	MPU address access time [3]	t_{ACCA}	348	—	203	—	144	—	ns
39	Write data set-up time [3]	t_{DSW}	185	—	102	—	60	—	ns
57	Address valid to data three-state time	t_{AVDZ}	—	10	—	10	—	10	ns

[1] All timing is given with respect to 20% and 70% of V_{DD}, unless otherwise noted.

[2] Input clock duty cycles other than 50% will affect the bus performance.

[3] For $f_{OP} \leq 2$ MHz the following formulae may be used to calculate parameter values:

$PW_{EL} = t_{CYC}/2 - 20\,\text{ns}$ $PW_{EH} = t_{CYC}/2 - 25\,\text{ns}$

$t_{AH} = t_{CYC}/8 - 10\,\text{ns}$ $t_{AD} = t_{CYC}/8 + 40\,\text{ns}$

$t_{AV} = PW_{EL} - t_{AD}$ $t_{DHW} = t_{CYC}/8$

$t_{ACCA} = t_{CYC} - t_f - t_{DSR} - t_{AD}$ $t_{DSW} = PW_{EH} - t_{DDW}$

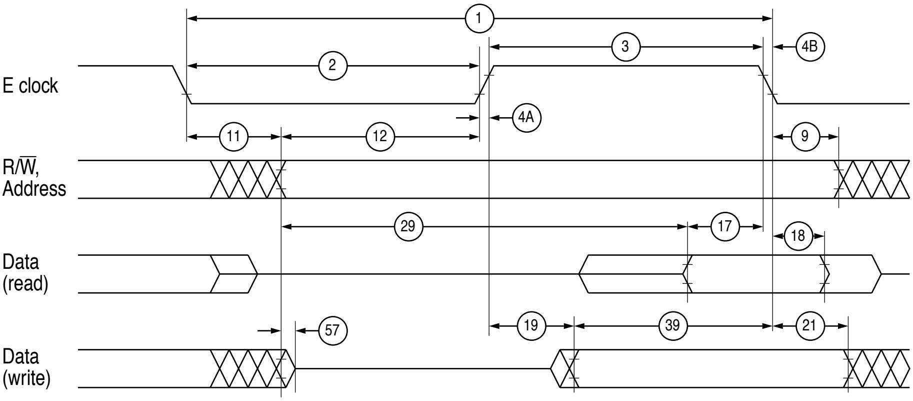


Figure A-13 Expansion bus timing

A.5.6 EEPROM characteristics

Characteristic	Temperature range –40 to +85°C	Unit
Programming time, t_{EEPROG} (1) <1 MHz, RCO enabled 1–2 MHz, RCO disabled ≥2 MHz & whenever RCO enabled	 10 20 10	ms
Erase time: byte, row and bulk (1)	10	ms
Write/erase endurance (2)	10000	cycles
Data retention (2)	10	years

(1) The RC oscillator (RCO) must be enabled (by setting the CSEL bit in the OPTION register) for EEPROM programming and erasure when the E clock frequency is less than 1.0 MHz.

(2) Refer to the current issue of Motorola's quarterly ***Reliability Monitor Report*** for the latest failure rate information.

12

A.5.7 EPROM characteristics

(V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, T_A = T_L to T_H, unless otherwise noted)

Characteristic	*Symbol*	Min	*Max*	*Unit*
Programming voltage	V_{PPE}	*12*	*12.75*	*V*
Programming voltage detect level	V_{PPH}	*TBD*	*TBD*	V
Programming time	t_{EPROG}	—	*5*	*ms*

B

MECHANICAL DATA AND ORDERING INFORMATION

B.1 Pin assignments

The MC68HC11PH8 is available in 84-pin PLCC or 112-pin TQFP packages; *in addition to those two packages, the MC68HC711PH8 is available in a windowed 84-pin CERQUAD package, to allow full use of the EPROM.*

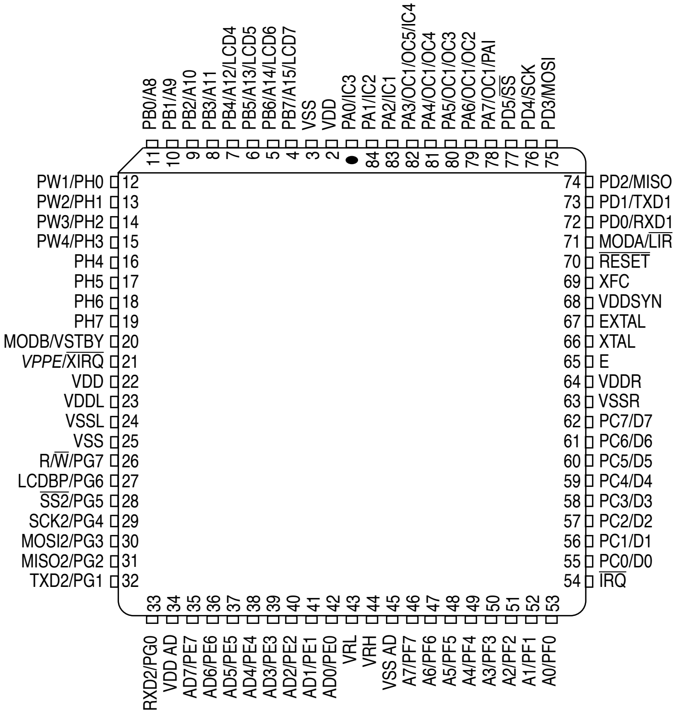


Figure B-1 84-pin PLCC/*CERQUAD* pinout

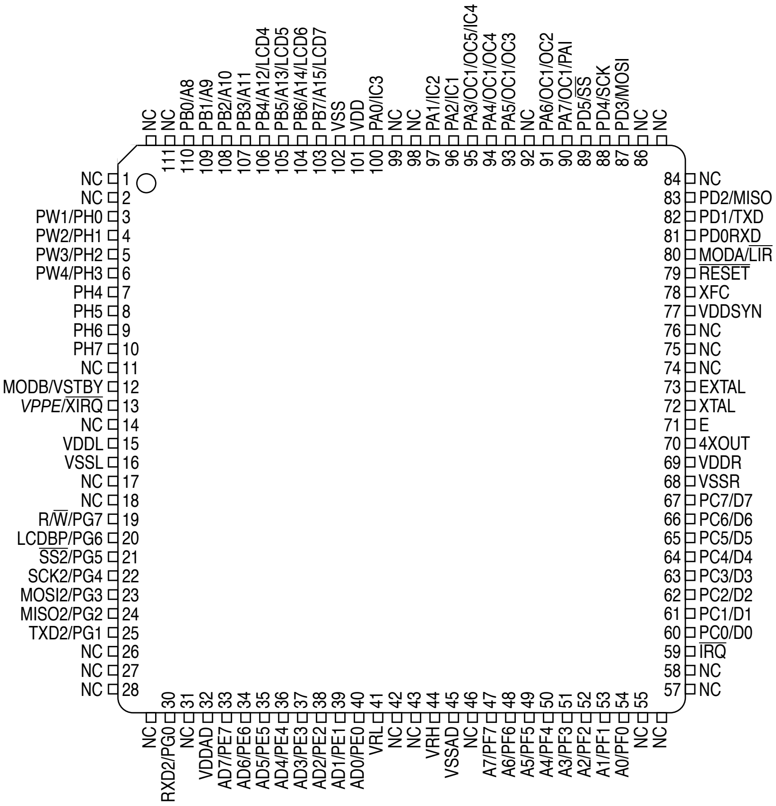
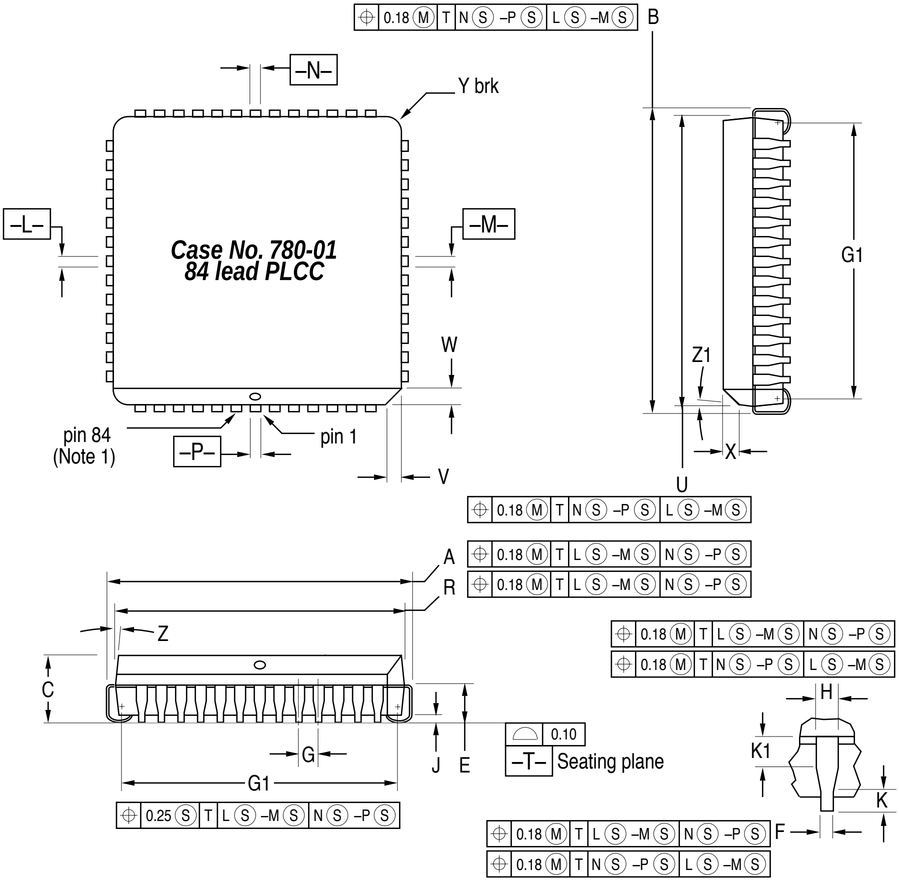


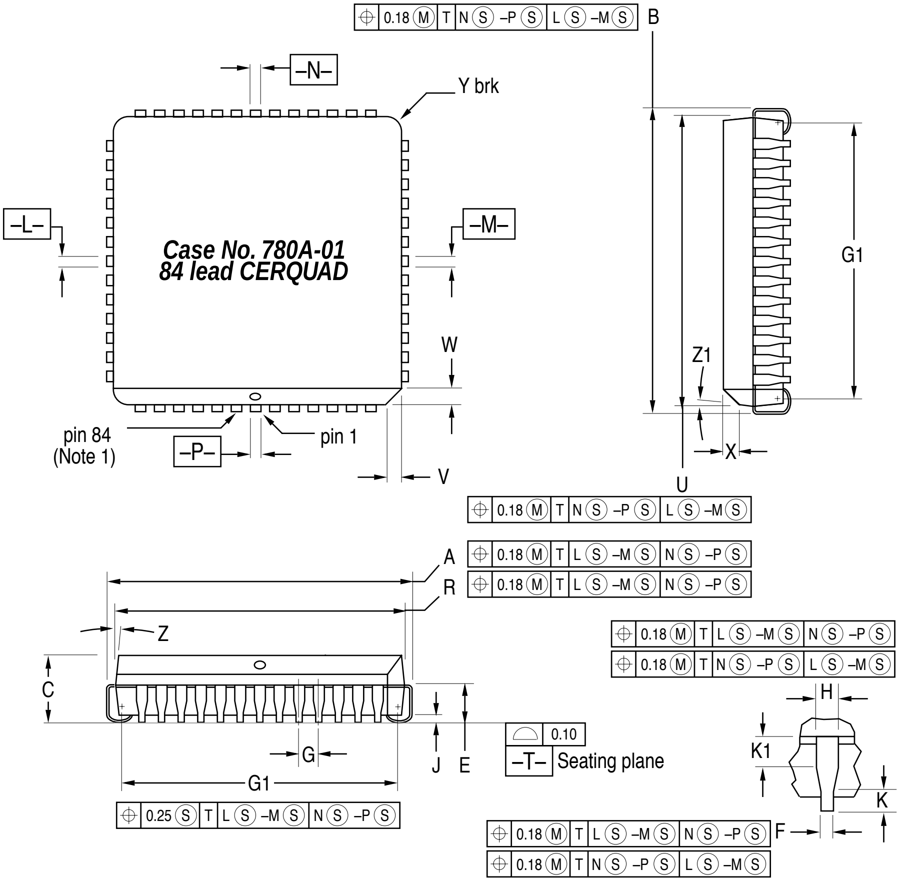
Figure B-2 112-pin TQFP pinout

B.2 Package dimensions


0.18 M T N S –P S L S –M S
B
–N–
Y brk
–L–
–M–
Case No. 780-01
84 lead PLCC
W
pin 84
(Note 1)
–P–
pin 1
V
G1
Z1
X
U
0.18 M T N S –P S L S –M S
A
R
0.18 M T L S –M S N S –P S
0.18 M T L S –M S N S –P S
Z
C
0.18 M T L S –M S N S –P S
0.18 M T N S –P S L S –M S
H
G
J
E
0.10
–T– Seating plane
K1
K
G1
0.25 S T L S –M S N S –P S
0.18 M T L S –M S N S –P S
F
0.18 M T N S –P S L S –M S

Dim.	Min.	Max.	Notes	Dim.	Min.	Max.
A	30.10	30.35	1. **Due to space limitations, this case shall be represented by a general case outline, rather than one showing all the leads.** 2. Datums –L–, –M–, –N– and –P– are determined where top of lead shoulder exits plastic body at mould parting line. 3. Dimension G1, true position to be measured at datum –T– (seating plane). 4. Dimensions R and U do not include mould protrusion. Allowable mould protrusion is 0.25 mm per side. 5. Dimensions and tolerancing per ANSI Y 14.5M, 1982. 6. All dimensions in mm.	U	29.21	29.36
B	30.10	30.35		V	1.07	1.21
C	4.20	4.57		W	1.07	1.21
E	2.29	2.79		X	1.07	1.42
F	0.33	0.48		Y	–	0.50
G	1.27 BSC			Z	2°	10°
H	0.66	0.81		G1	28.20	28.70
J	0.51	–		K1	1.02	–
K	0.64	–		Z1	2°	10°
R	29.21	29.36				

Figure B-3 84-pin PLCC mechanical dimensions


0.18 M T N S –P S L S –M S B
–N–
Y brk
–L–
–M–
Case No. 780A-01
84 lead CERQUAD
G1
W
Z1
pin 84
(Note 1)
–P–
pin 1
X
V
U
0.18 M T N S –P S L S –M S
A
0.18 M T L S –M S N S –P S
R
0.18 M T L S –M S N S –P S
Z
0.18 M T L S –M S N S –P S
0.18 M T N S –P S L S –M S
C
H
K1
0.10
–T– Seating plane
G
J
E
K
G1
0.25 S T L S –M S N S –P S
0.18 M T L S –M S N S –P S
F
0.18 M T N S –P S L S –M S

Dim.	Min.	Max.	Notes	Dim.	Min.	Max.
A	30.10	30.35	1. **Due to space limitations, this case shall be represented by a general case outline, rather than one showing all the leads.** 2. Datums –L–, –M–, –N– and –P– are determined where top of lead shoulder exits package body at glass parting line. 3. Dimension G1, true position to be measured at datum –T– (seating plane). 4. Dimensions R and U do not include glass protrusion. Allowable glass protrusion is 0.25 mm per side. 5. Dimensions and tolerancing per ANSI Y 14.5M, 1982. 6. All dimensions in mm.	U	29.21	29.36
B	30.10	30.35		V	1.07	1.21
C	4.20	4.57		W	1.07	1.21
E	2.29	2.79		X	1.07	1.42
F	0.33	0.53		Y	–	0.50
G	1.27 BSC			Z	2°	10°
H	0.66	0.81		G1	28.20	28.70
J	0.51	–		K1	1.02	–
K	0.64	–		Z1	2°	10°
R	29.21	29.36				

Figure B-4 84-pin CERQUAD mechanical dimensions

Please contact your Motorola Sales Office
for up-to-date information on the mechanical
dimensions for this package type.

Figure B-5 112-pin TQFP mechanical dimensions

B.3 Ordering Information

Use the information in Table B-1 to specify the appropriate device type when placing an order.

Table B-1 Ordering information

Package	Temperature	Description	Frequency	MC order number
84-pin PLCC	–40 to +85°C	Custom ROM	3 MHz 4 MHz	MC68HC11PH8CFN3 MC68HC11PH8CFN4
		Custom ROM, with security feature	3 MHz 4 MHz	MC68S11PH8CFN3 MC68S11PH8CFN4
		OTPROM (with security feature)	3 MHz 4 MHz	MC68S711PH8CFN3 MC68S711PH8CFN4
112-pin TQFP	–40 to +85°C	Custom ROM	3 MHz 4 MHz	MC68HC11PH8CPV3 MC68HC11PH8CPV4
		Custom ROM, with security feature	3 MHz 4 MHz	MC68S11PH8CPV3 MC68S11PH8CPV4
		OTPROM (with security feature)	3 MHz 4 MHz	MC68S711PH8CPV3 MC68S711PH8CPV4
84-pin CERQUAD	–40 to +85°C	EPROM (with security feature)	3 MHz 4 MHz	MC68S711PH8CFS3 MC68S711PH8CFS4

C
DEVELOPMENT SUPPORT

The following information provides a reference to development tools for the M68HC11 family of microcontrollers. For more detailed information please refer to the appropriate system manual.

Table C-1 M68HC11 development tools

Devices	Evaluation boards	Evaluation modules	Evaluation systems/kits	Programmer boards
MC68HC11PH8, *MC68HC711PH8*	–	M68EM11PH8	–	M68SPGMR11

Note: Target cables for the evaluation module should be ordered separately.

C.1 EVS — Evaluation system

The EVS is an economical tool for designing, debugging and evaluating target systems based on the MC68HC11PH8 and MC68HC711PH8 device types. The two printed circuit boards that comprise the EVS are the M68EM11PH8 emulator module and the M68PFB11KIT platform board. The main features of the EVS are as follows:

- Monitor/debugger firmware
- Single-line assembler/disassembler
- Host computer download capability
- Dual memory maps:
  - 64Kbyte monitor map that includes 16Kbytes of monitor EPROM
  - *MC68HC711PH8 user map that includes 64Kbytes of emulation RAM*
- MCU extension I/O port for single chip, expanded and special test operating modes
- RS-232C terminal and host I/O ports
- Logic analyser connector

C.2 MMDS11 — Motorola modular development system

The MMDS11 is an emulator system that provides a bus state analyser and real-time memory windows. The unit's integrated design environment includes an editor, an assembler, user interface and source-level debug. A complete MMDS11 consists of:

- A station module — the metal MMDS11 enclosure, containing the control board and the internal power supply. Most system cables connect to the MMDS11 station module. (The cable to an optional target system, however, runs through an aperture in the station module enclosure to connect directly to the emulator module).
- An emulator module (EM) — such as the EM11PH8: a printed circuit board that enables system functionality for a specific set of MCUs. The EM fits into the station module through a sliding panel in the enclosure top. The EM has a connector for the target cable.
- Two logic clip cable assemblies — twisted pair cables that connect the station module to your target system, a test fixture, a clock or any other circuitry useful for evaluation or analysis. One end of each cable assembly has a moulded connector, which fits into station module pod A or pod B. Leads at the other end of the cable terminate in female probe tips. Ball clips come with the cables.
- A 9-lead RS-232 serial cable — the cable that connects the station module to the host computer's RS-232 port.

C.3 SPGMR11 — Serial programmer system

The SPGMR11 is an economical tool for programming M68HC11 MCUs. The system consists of the M68SPGMR11 unit and a programming module which adapts the SPGMR11 to the appropriate MCU and package type. The programming module can be ordered as M68PA11PH8FN84 (for the 84-pin package) and M68PA11PH8PV112 (for the 112-pin package).

GLOSSARY

This section contains abbreviations and specialist words used in this data sheet and throughout the industry. Further information on many of the terms may be gleaned from Motorola's ***M68HC11 Reference Manual, M68HC11RM/AD***, or from a variety of standard electronics text books.

$xxxx The digits following the '$' are in hexadecimal format.

%xxxx The digits following the '%' are in binary format.

A/D, **ADC** Analog-to-digital (converter).

Bootstrap mode In this mode the device automatically loads its internal memory from an external source on reset and then allows this program to be executed.

Byte Eight bits.

CCR Condition codes register; an integral part of the CPU.

CERQUAD A ceramic package type, principally used for EPROM and high temperature devices.

Clear '0' — the logic zero state; the opposite of 'set'.

CMOS Complementary metal oxide semiconductor. A semiconductor technology chosen for its low power consumption and good noise immunity.

COP Computer operating properly. *aka* 'watchdog'. This circuit is used to detect device runaway and provide a means for restoring correct operation.

CPU Central processing unit.

D/A, DAC Digital-to-analog (converter).

EEPROM Electrically erasable programmable read only memory. *aka* 'EEROM'.

EPROM Erasable programmable read only memory. This type of memory requires exposure to ultra-violet wavelengths in order to erase previous data. *aka* 'PROM'.

ESD Electrostatic discharge.

Expanded mode In this mode the internal address and data bus lines are connected to external pins. This enables the device to be used in much more complex systems, where there is a need for external memory for example.

EVS Evaluation system. One of the range of platforms provided by Motorola for evaluation and emulation of their devices.

HCMOS High-density complementary metal oxide semiconductor. A semiconductor technology chosen for its low power consumption and good noise immunity.

I/O Input/output; used to describe a bidirectional pin or function.

Input capture (IC) This is a function provided by the timing system, whereby an external event is ‘captured’ by storing the value of a counter at the instant the event is detected.

Interrupt This refers to an asynchronous external event and the handling of it by the MCU. The external event is detected by the MCU and causes a predetermined action to occur.

$\overline{\textbf{IRQ}}$ Interrupt request. The overline indicates that this is an active-low signal format.

K byte A kilo-byte (of memory); 1024 bytes.

LCD Liquid crystal display.

LSB Least significant byte.

M68HC11 Motorola’s family of advanced 8-bit MCUs.

MCU Microcontroller unit.

MI BUS Motorola interconnect bus. A single wire, medium speed serial communications protocol.

MSB Most significant byte.

Nibble Half a byte; four bits.

NRZ Non-return to zero.

Opcode The opcode is a byte which identifies the particular instruction and operating mode to the CPU. See also: prebyte, operand.

Operand The operand is a byte containing information the CPU needs to execute a particular instruction. There may be from 0 to 3 operands associated with an opcode. See also: opcode, prebyte.

Output compare (OC) This is a function provided by the timing system, whereby an external event is generated when an internal counter value matches a predefined value.

PLCC Plastic leaded chip carrier package.

PLL Phase-locked loop circuit. This provides a method of frequency multiplication, to enable the use of a low frequency crystal in a high frequency circuit.

Prebyte This byte is sometimes required to qualify an opcode, in order to fully specify a particular instruction. See also: opcode, operand.

Pull-down, pull-up These terms refer to resistors, sometimes internal to the device, which are permanently connected to either ground or V_{DD}.

PWM Pulse width modulation. This term is used to describe a technique where the width of the high and low periods of a waveform is varied, usually to enable a representation of an analog value.

QFP Quad flat pack package.

RAM Random access memory. Fast read and write, but contents are lost when the power is removed.

RFI Radio frequency interference.

RTI Real-time interrupt.

ROM Read-only memory. This type of memory is programmed during device manufacture and cannot subsequently be altered.

RS-232C A standard serial communications protocol.

SAR Successive approximation register.

SCI Serial communications interface.

Set ‘1’ — the logic one state; the opposite of ‘clear’.

Silicon glen An area in the central belt of Scotland, so called because of the concentration of semiconductor manufacturers and users found there.

Single chip mode In this mode the device functions as a self contained unit, requiring only I/O devices to complete a system.

SPI Serial peripheral interface.

Test mode This mode is intended for factory testing.

TTL Transistor-transistor logic.

UART Universal asynchronous receiver transmitter.

VCO Voltage controlled oscillator.

Watchdog *see* ‘COP’.

Wired-OR A means of connecting outputs together such that the resulting composite output state is the logical OR of the state of the individual outputs.

Word Two bytes; 16 bits.

$\overline{\textbf{XIRQ}}$ Non-maskable interrupt request. The overline indicates that this has an active-low signal format.

THIS PAGE INTENTIONALLY LEFT BLANK

INDEX

In this index numeric entries are placed first; page references in *italics* indicate that the reference is to a figure.

How to reach us:
Mfax™: RMFAX0@email.sps.mot.com – TOUCHTONE (602) 244-6609
INTERNET: http://www.mot.com/SPS/
USA/EUROPE: Motorola Literature Distribution; P.O. Box 5405; Denver, Colorado 80217. 303-675-2140
JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC, Toshikatsu Otsuki, 6F Seibu-Butsuryu-Center,
3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 81-3-3521-8315
HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park, 51 Ting Kok Road,
Tai Po, N.T., Hong Kong. 852-26629298

MOTOROLA