MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

Order this document
by MC68HC11KTS/D

M68HC11 K Series

Technical Summary
8-Bit Microcontroller

The M68HC11 K-series microcontroller units (MCUs) are high-performance derivatives of the MC68HC11F1 and have several additional features. The MC68HC11K0, MC68HC11K1, MC68HC11K3, MC68HC11K4 and MC68HC711K4 comprise the series. These MCUs, with a nonmultiplexed expanded bus, are characterized by high speed and low power consumption. Their fully static design allows operation at frequencies from 4 MHz to dc.

This document contains information concerning standard, custom-ROM, and extended-voltage devices. Standard devices include those with disabled ROM (MC68HC11K1), disabled EEPROM (MC68HC11K3), disabled ROM and EEPROM (MC68HC11K0), or EPROM replacing ROM (MC68HC711K4). Custom-ROM devices have a ROM array that is programmed at the factory to customer specifications. Extended-voltage devices are guaranteed to operate over a much greater voltage range (3.0 Vdc to 5.5 Vdc) at lower frequencies than the standard devices. Refer to the device ordering information tables for details concerning these differences.

1 Features

- M68HC11 CPU
- Power Saving STOP and WAIT Modes
- 768 Bytes RAM (All Saved During Standby)
- 24 Kbytes ROM or EPROM
- 640 Bytes Electrically Erasable Programmable Read Only Memory (EEPROM)
- Optional Security Feature Protects Memory Contents
- On-Chip Memory Mapping Logic Allows Expansion to Over 1 Mbyte of Address Space
- PROG Mode Allows Use of Standard EPROM Programmer (27C256 Footprint)
- Nonmultiplexed Address and Data Buses
- Four Programmable Chip Selects with Clock Stretching (Expanded Modes)
- Enhanced 16-Bit Timer with Four-Stage Programmable Prescaler
  - — Three Input Capture (IC) Channels
  - — Four Output Compare (OC) Channels
  - — One Additional Channel, Selectable as Fourth IC or Fifth OC
- 8-Bit Pulse Accumulator
- Four 8-Bit or Two 16-Bit Pulse Width Modulation (PWM) Timer Channels
- Real-Time Interrupt Circuit
- Computer Operating Properly (COP) Watchdog
- Clock Monitor
- Enhanced Asynchronous Nonreturn to Zero (NRZ) Serial Communications Interface (SCI)
- Enhanced Synchronous Serial Peripheral Interface (SPI)
- Eight-Channel 8-Bit Analog-to-Digital (A/D) Converter
- Seven Bidirectional Input/Output (I/O) Ports (54 Pins)
- One Fixed Input-Only Port (8 Pins)
- Available in 84-Pin Plastic Leaded Chip Carrier (PLCC), 84-Pin Windowed Ceramic Leaded Chip Carrier (CLCC), and 80-Pin Quad Flat Pack (QFP)

This document contains information on a new product. Specifications and information herein are subject to change without notice.

© MOTOROLA INC., 1997

Table 1 Standard Device Ordering Information

Package	Temperature	CONFIG	Description	Frequency	MC Order Number
84-Pin PLCC	–40°to + 85°C	$DF	BUFFALO ROM	4 MHz	MC68HC11K4BCFN4
	–40°to + 85°C	$DD	No ROM	2 MHz	MC68HC11K1CFN2
				3 MHz	MC68HC11K1CFN3
				4 MHz	MC68HC11K1CFN4
	–40°to + 105°C	$DD	No ROM	2 MHz	MC68HC11K1VFN2
				3 MHz	MC68HC11K1VFN3
				4 MHz	MC68HC11K1VFN4
	–40°to + 125°C	$DD	No ROM	2 MHz	MC68HC11K1MFN2
				3 MHz	MC68HC11K1MFN3
				4 MHz	MC68HC11K1MFN4
	–40°to + 85°C	$DC	No ROM, No EEPROM	2 MHz	MC68HC11K0CFN2
				3 MHz	MC68HC11K0CFN3
				4 MHz	MC68HC11K0CFN4
	–40°to + 105°C	$DC	No ROM, No EEPROM	2 MHz	MC68HC11K0VFN2
				3 MHz	MC68HC11K0VFN3
				4 MHz	MC68HC11K0VFN4
	–40°to + 125°C	$DC	No ROM, No EEPROM	2 MHz	MC68HC11K0MFN2
				3 MHz	MC68HC11K0MFN3
				4 MHz	MC68HC11K0MFN4
	–40°to + 85°C	$DF	OTPROM	2 MHz	MC68HC711K4CFN2
				3 MHz	MC68HC711K4CFN3
				4 MHz	MC68HC711K4CFN4
	–40°to + 105°C	$DF	OTPROM	2 MHz	MC68HC711K4VFN2
				3 MHz	MC68HC711K4VFN3
				4 MHz	MC68HC711K4VFN4
	–40°to + 125°C	$DF	OTPROM	2 MHz	MC68HC711K4MFN2
				3 MHz	MC68HC711K4MFN3
				4 MHz	MC68HC711K4MFN4
80-Pin QFP (14 mm X 14 mm)	–40°to + 85°C	$DF	BUFFALO ROM	4 MHz	MC68HC11K4BCFU4
	–40°to + 85°C	$DD	No ROM	2 MHz	MC68HC11K1CFU2
				3 MHz	MC68HC11K1CFU3
				4 MHz	MC68HC11K1CFU4
	–40°to + 105°C	$DD	No ROM	2 MHz	MC68HC11K1VFU2
				3 MHz	MC68HC11K1VFU3
				4 MHz	MC68HC11K1VFU4
	–40°to + 85°C	$DC	No ROM, No EEPROM	2 MHz	MC68HC11K0CFU2
				3 MHz	MC68HC11K0CFU3
				4 MHz	MC68HC11K0CFU4
	–40°to + 105°C	$DC	No ROM, No EEPROM	2 MHz	MC68HC11K0VFU2
				3 MHz	MC68HC11K0VFU3
				4 MHz	MC68HC11K0VFU4

Table 1 Standard Device Ordering Information (Continued)

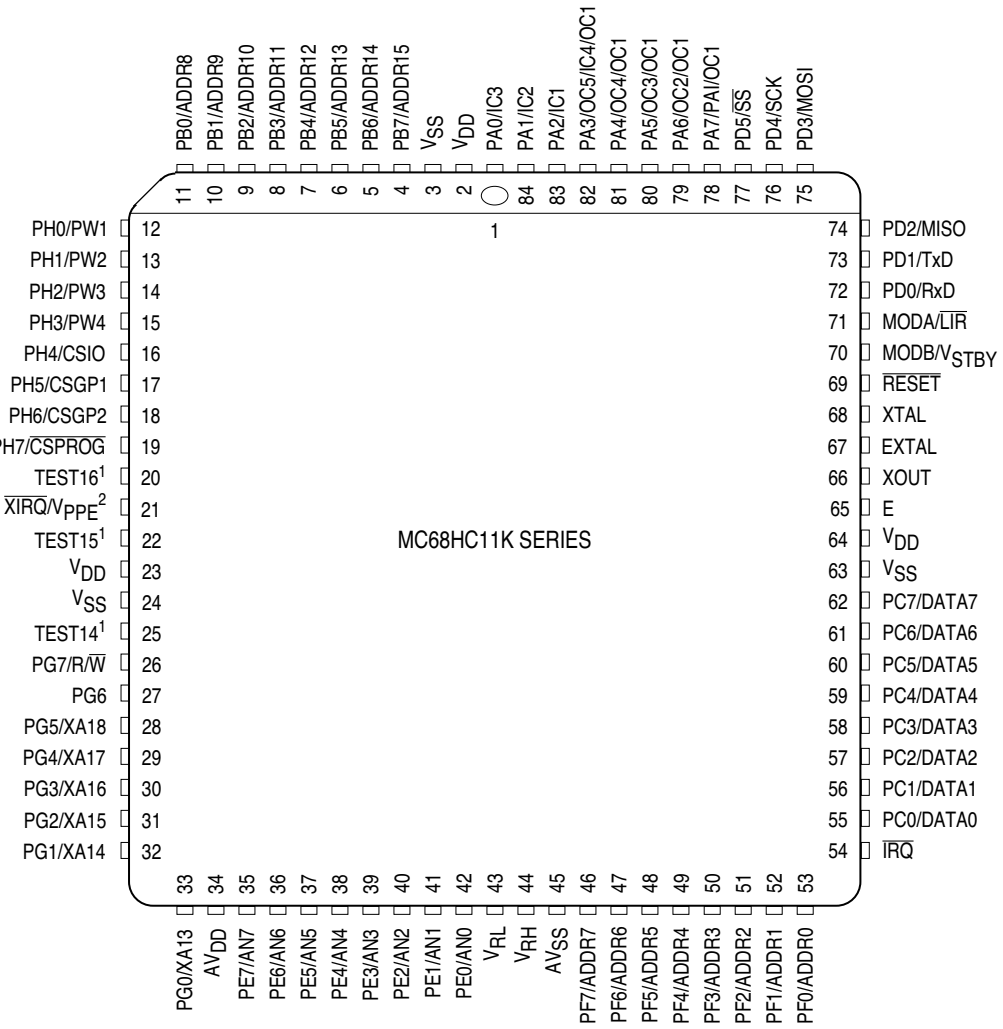
Package	Temperature	CONFIG	Description	Frequency	MC Order Number
84-Pin CLCC (Windowed)	–40°to + 85°C	$DF	EPROM	2 MHz	MC68HC711K4CFS2
				3 MHz	MC68HC711K4CFS3
				4 MHz	MC68HC711K4CFS4
	–40°to + 105°C	$DF	EPROM	2 MHz	MC68HC711K4VFS2
				3 MHz	MC68HC711K4VFS3
				4 MHz	MC68HC711K4VFS4
	–40°to + 125°C	$DF	EPROM	2 MHz	MC68HC711K4MFS2
				3 MHz	MC68HC711K4MFS3
				4 MHz	MC68HC711K4MFS4

Table 2 Extended Voltage (3.0 Vdc to 5.5 Vdc) Device Ordering Information

Package	Temperature	Description	Frequency	MC Order Number
84-Pin PLCC	–20°to + 70°C	Custom ROM	1 MHz	MC68L11K4FN1
			3 MHz	MC68L11K4FN3
		No ROM	1 MHz	MC68L11K1FN1
			3 MHz	MC68L11K1FN3
		No ROM, No EEPROM	1 MHz	MC68L11K0FN1
			3 MHz	MC68L11K0FN3
		Custom ROM, No EEPROM	1 MHz	MC68L11K3FN1
			3 MHz	MC68L11K3FN3
80-Pin QFP	–20°to + 70°C	Custom ROM	1 MHz	MC68L11K4FU1
			3 MHz	MC68L11K4FU3
		No ROM	1 MHz	MC68L11K1FU1
			3 MHz	MC68L11K1FU3
		No ROM, No EEPROM	1 MHz	MC68L11K0FU1
			3 MHz	MC68L11K0FU3
		Custom ROM, No EEPROM	1 MHz	MC68L11K3FU1
			3 MHz	MC68L11K3FU3

Table 3 Custom ROM Device Ordering Information

Package	Temperature	Description	Frequency	MC Order Number
84-Pin PLCC	–40°to + 85°C	Custom ROM	2 MHz	MC68HC11K4CFN2
			3 MHz	MC68HC11K4CFN3
			4 MHz	MC68HC11K4CFN4
	–40°to + 105°C	Custom ROM	2 MHz	MC68HC11K4VFN2
			3 MHz	MC68HC11K4VFN3
			4 MHz	MC68HC11K4VFN4
	–40°to + 125°C	Custom ROM	2 MHz	MC68HC11K4MFN2
			3 MHz	MC68HC11K4MFN3
			4 MHz	MC68HC11K4MFN4
	–40°to + 85°C	Custom ROM, No EEPROM	2 MHz	MC68HC11K3CFN2
			3 MHz	MC68HC11K3CFN3
			4 MHz	MC68HC11K3CFN4
	–40°to + 105°C	Custom ROM, No EEPROM	2 MHz	MC68HC11K3VFN2
			3 MHz	MC68HC11K3VFN3
			4 MHz	MC68HC11K3VFN4
	–40°to + 125°C	Custom ROM, No EEPROM	2 MHz	MC68HC11K3MFN2
			3 MHz	MC68HC11K3MFN3
			4 MHz	MC68HC11K3MFN4
80-Pin QFP	–40°to + 85°C	Custom ROM	2 MHz	MC68HC11K4CFU2
			3 MHz	MC68HC11K4CFU3
			4 MHz	MC68HC11K4CFU4
	–40°to + 105°C	Custom ROM	2 MHz	MC68HC11K4VFU2
			3 MHz	MC68HC11K4VFU3
			4 MHz	MC68HC11K4VFU4
	–40°to + 85°C	Custom ROM, No EEPROM	2 MHz	MC68HC11K3CFU2
			3 MHz	MC68HC11K3CFU3
			4 MHz	MC68HC11K3CFU4
	–40°to + 105°C	Custom ROM, No EEPROM	2 MHz	MC68HC11K3VFU2
			3 MHz	MC68HC11K3VFU3
			4 MHz	MC68HC11K3VFU4



1. Pins 20, 22, and 25 are used only during factory testing and should not be connected to external circuitry.
2. V_{PPE} applies only to devices with EPROM.

Figure 1 Pin Assignments for 84-Pin PLCC/CLCC

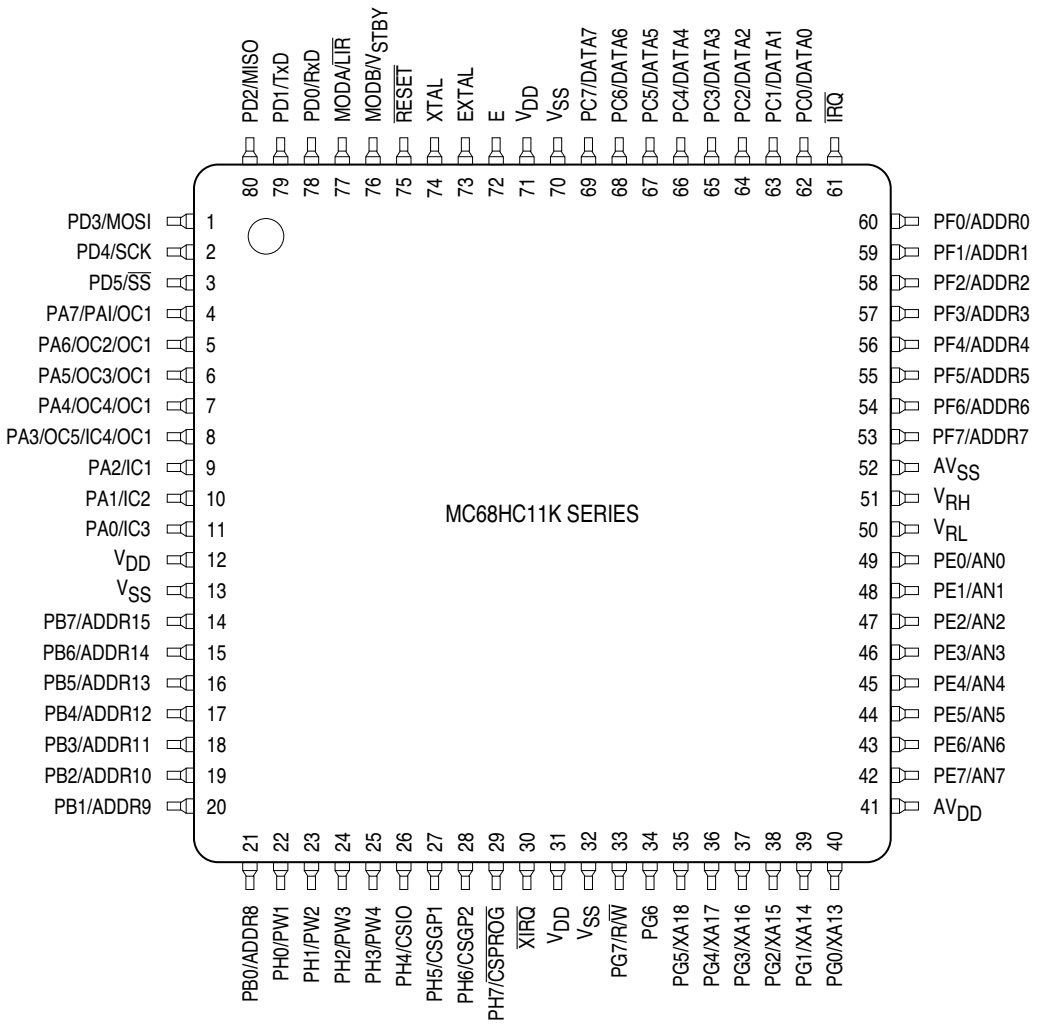
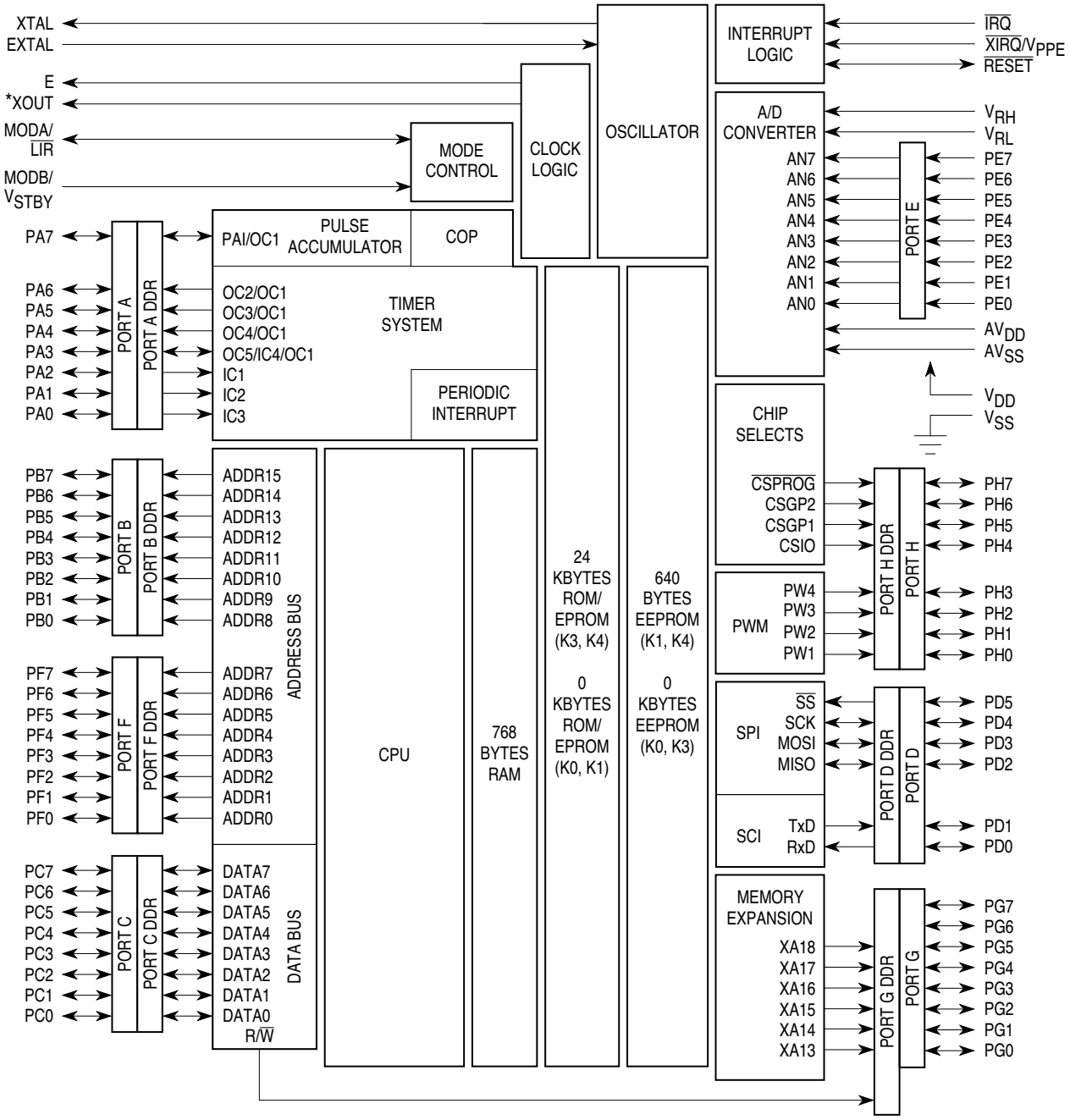


Figure 2 Pin Assignments for 80-Pin 14 mm X 14 mm TQFP



*XOUT pin omitted on 80-pin QFP.

Figure 3 M68HC11 K-Series Block Diagram

TABLE OF CONTENTS

Section			Page
1		**Features**	**1**
2		**Operating Modes**	**11**
	2.1	Single-Chip Operating Mode	11
	2.2	Expanded Operating Mode	11
	2.3	Bootstrap Mode	11
	2.4	Special Test Mode	11
	2.5	Mode Selection	11
3		**On-Chip Memory**	**14**
	3.1	Memory Map and Register Block	14
	3.2	RAM	17
	3.3	ROM/EPROM	18
	3.4	EEPROM	22
	3.5	Configuration Control Register (CONFIG)	24
	3.6	Security Feature	25
4		**Memory Expansion and Chip Selects**	**27**
	4.1	Memory Expansion	27
	4.2	Overlap Guidelines	30
	4.3	Chip Selects	30
	4.3.1	Program Chip Select (CSPROG)	31
	4.3.2	I/O Chip Select (CSIO)	31
	4.3.3	General-Purpose Chip Selects (CSGP1, CSGP2)	32
	4.3.4	Chip Select Priorities	32
	4.3.5	Chip Select Control Registers	32
	4.3.6	Examples of Memory Expansion Using Chip Selects	35
5		**Resets and Interrupts**	**38**
6		**Parallel Input/Output**	**42**
7		**Serial Communications Interface**	**49**
8		**Serial Peripheral Interface**	**56**
9		**Analog-to-Digital Converter**	**60**
10		**Main Timer**	**64**
	10.1	Real-Time Interrupt	70
11		**Pulse Accumulator**	**71**
12		**Pulse-Width Modulation Timer**	**74**
	12.1	PWM Boundary Cases	78

REGISTER INDEX

	C		
CFORC	Timer Compare Force	$000B	66
CONFIG	System Configuration Register	$003F	25
COPRST	Arm/Reset COP Timer Circuitry	$003A	40
CSCSTR	Chip Select Clock Stretch	$005A	33
CSCTL	Chip Select Control	$005B	32
	D		
DDRA	Data Direction Register for Port A	$0001	42
DDRB	Data Direction Register for Port B	$0002	43
DDRF	Data Direction Register for Port F	$0003	46
DDRG	Data Direction Register for Port G	$007F	47
DDRH	Data Direction Register for Port H	$007D	46
	E		
EPROG	EPROM Programming Control	$002B	19
	G		
GPCS1A	General-Purpose Chip Select 1 Address	$005C	33
GPCS1C	General-Purpose Chip Select 1 Control	$005D	34
GPCS2A	General-Purpose Chip Select 2 Address	$005E	34
GPCS2C	General-Purpose Chip Select 2 Control	$005F	34
	H		
HPRIO	Highest Priority I-Bit Interrupt and Miscellaneous	$003C	11, 40
	I		
INIT	RAM and Register Mapping	$003D	18
INIT2	EEPROM Mapping	$0037	24
	M		
MMSIZ	Memory Mapping Size	$0056	28
MMWBR	Memory Mapping Window Base	$0057	29
	O		
OC1D	Output Compare 1 Data	$000D	66
OC1M	Output Compare 1 Mask	$000C	66
OPT2	System Configuration Options 2	$0038	12, 44, 59
OPTION	System Configuration Options	$0039	39
	P		
PACNT	Pulse Accumulator Counter	$0027	73
PACTL	Pulse Accumulator Control	$0026	73
PGAR	Port G Assignment	$002D	28, 47
PORTA	Port A Data	$0000	42
PORTB	Port B Data	$0004	43
PORTC	Port C Data	$0006	43
PORTE	Port E Data	$000A	46
PORTF	Port F Data	$0005	46
PORTG	Port G Data	$007E	47
PORTH	Port H Data	$007C	46
PPAR	Port Pull-Up Assignment	$002C	48
PPROG	EEPROM Programming Control	$003B	22
PWCLK	Pulse-Width Modulation Clock Select	$0060	62, 76

PWCNT[4:1]	Pulse-Width Modulation Timer Counter 1 to 4	$0064–$0067	77
PWDTY[4:1]	Pulse-Width Modulation Timer Duty Cycle 1 to 4	$006C–$006F	78
PWEN	Pulse-Width Modulation Timer Enable	$0063	77
PWPER[4:1]	Pulse-Width Modulation Timer Period 1 to 4	$0068–$006B	78
PWPOL	Pulse-Width Modulation Timer Polarity	$0061	62, 76
PWSCAL	Pulse-Width Modulation Timer Prescaler	$0062	63, 77

S

SCBDH/L	SCI Baud Rate Control High/Low	$0070, $0071	52
SCCR1	SCI Control 1	$0072	45, 52
SCCR2	SCI Control 2	$0073	53
SCSR1	SCI Status Register 1	$0074	54
SCSR2	SCI Status Register 2	$0075	55
SPCR	Serial Peripheral Control	$0028	45
SPCR	Serial Peripheral Control Register	$0028	57
SPDR	SPI Data	$002A	58
SPSR	Serial Peripheral Status Register	$0029	58

T

TCNT	Timer Count	$000E, $000F	66
TCTL2	Timer Control 2	$0021	67
TFLG2	Timer Interrupt Flag 2	$0025	69, 72
TI4/O5	Timer Input Capture 4/Output Compare 5	$001E–$001F	67
TMSK1	Timer Interrupt Mask 1	$0022	68
TMSK2	Timer Interrupt Mask 2	$0024	68, 72
TOC1–TOC4	Timer Output Compare	$0016–$001D	67

2 Operating Modes

The M68HC11 K-series MCUs have four modes of operation that directly affect the address space. These modes are described as follows.

2.1 Single-Chip Operating Mode

In single-chip operating mode, the M68HC11 K-series MCUs are stand-alone microcontrollers with no external address or data bus. Addressing range is 64 Kbytes and is limited to on-chip resources. Refer to the memory map diagram.

2.2 Expanded Operating Mode

In expanded operating mode, the MCU has a 64 Kbyte address range and, using the expansion bus, can access external resources within the 64 Kbyte space. This space includes the same on-chip memory addresses used for single-chip mode, in addition to addressing capabilities for external peripheral and memory devices. Addressing beyond 64 Kbytes is available only in expanded mode using the on-chip, register-based memory mapping logic. The additional address lines for memory expansion (XA[18:13]) are implemented as alternate functions of port G. The expansion bus (external address and data buses) is made up of ports B, C, and F, and the R/$\overline{W}$ signal. In expanded operating 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. Refer to the memory map diagram.

2.3 Bootstrap Mode

Bootstrap mode allows special-purpose programs to be loaded into internal RAM. The MCU contains 448 bytes of bootstrap ROM which is enabled and present in the memory map only when the device is in bootstrap mode. The bootstrap ROM contains a program which initializes the SCI and allows the user to download up to 768 bytes of code into on-chip RAM. After a four-character delay, or after receiving the character for address $037F, control passes to the loaded program at $0080. Refer to the memory map diagram. Refer also to Application Note *M68HC11 Bootstrap Mode* (AN1060/D).

2.4 Special Test Mode

Special test mode is used primarily for factory testing. In this operating mode, ROM/EPROM is removed from the address space and interrupt vectors are accessed externally at $BFC0–$BFFF.

2.5 Mode Selection

Operating modes are selected by a combination of logic levels applied to two input pins (MODA and MODB) during reset. The logic level present (at the rising edge of reset) on these inputs is reflected in bits in the HPRIO register. After reset, the operating mode may be changed according to the table contained in the description of the HPRIO register.

The functions of two features that are enabled by bits in OPT2 register are dependent upon the operating mode. LIR driven is enabled with the LIRDV bit. Internal read visibility/not E is enabled with the IRVNE bit. Refer to the OPT2 register description that follows HPRIO.

HPRIO —Highest Priority I-Bit Interrupt and Miscellaneous **$003C**

	Bit 7	6	5	4	3	2	1	Bit 0	
	RBOOT*	SMOD*	MDA*	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0	
RESET:	0	0	0	0	0	1	1	0	Single Chip
	0	0	1	0	0	1	1	0	Expanded
	1	1	0	0	0	1	1	0	Bootstrap
	0	1	1	0	0	1	1	0	Special Test

*The reset values of RBOOT, SMOD, and MDA depend on the mode selected at power up.

RBOOT — Read Bootstrap ROM/EPROM

Valid only when SMOD is set (bootstrap or special test mode). Can only be written in special modes.

0 = Bootstrap ROM disabled and not in map

1 = Bootstrap ROM enabled and in map at $BE00–$BFFF

SMOD and MDA —Special Mode Select and Mode Select A

These two bits can be read at any time. They can be written anytime in special modes. MDA can only be written once in normal modes. SMOD cannot be set once it has been cleared.

Inputs		Mode	Latched at Reset	
MODB	**MODA**		**SMOD**	**MDA**
1	0	Single Chip	0	0
1	1	Expanded	0	1
0	0	Bootstrap	1	0
0	1	Special Test	1	1

PSEL[4:0] —Priority Select Bits [4:0]

Refer to **5 Resets and Interrupts**.

OPT2 — System Configuration Options 2 **$0038**

	Bit 7	6	5	4	3	2	1	Bit 0
	LIRDV	CWOM	—	IRVNE*	LSBF	SPR2	XDV1	XDV0
RESET:	0	0	0	—	0	0	0	0

*Can be written only once in normal modes. Can be written anytime in special modes.

LIRDV —LIR Driven

In single-chip and bootstrap modes, this bit has no meaning or effect. The LIR pin is normally configured for wired-OR operation (only pulls low). In order to detect consecutive instructions in a high-speed application, this signal can be made to drive high for a short time to prevent false triggering.

0 = LIR not driven high out of reset

1 = LIR driven high for one quarter cycle to reduce transition time

CWOM —Port C Wired-OR Mode

Refer to **6 Parallel Input/Output**.

Bit 5 —Not implemented

Always read zero

IRVNE —Internal Read Visibility/Not E

IRVNE can be written only once in normal modes (SMOD = 0). In special modes IRVNE can be written any time. In special test mode, IRVNE is reset to one. In all other modes, IRVNE is reset to zero.

In expanded modes this bit determines whether IRV is on or off.

0 = No internal read visibility on external bus

1 = Data from internal reads is driven out the external data bus.

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

0 = E is driven out from the chip.

1 = E pin is driven low. Refer to the following table.

Mode	IRVNE Out of Reset	E Clock Out of Reset	IRV Out of 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	Anytime
Special Test	1	On	On	IRV	Anytime

LSBF —LSB First Enable

Refer to **8 Serial Peripheral Interface**.

SPR2 —SPI Clock Rate Select

Refer to **8 Serial Peripheral Interface**.

XDV[1:0] —XOUT Clock Divide Select

Controls the frequency of the clock driven out of the XOUT pin

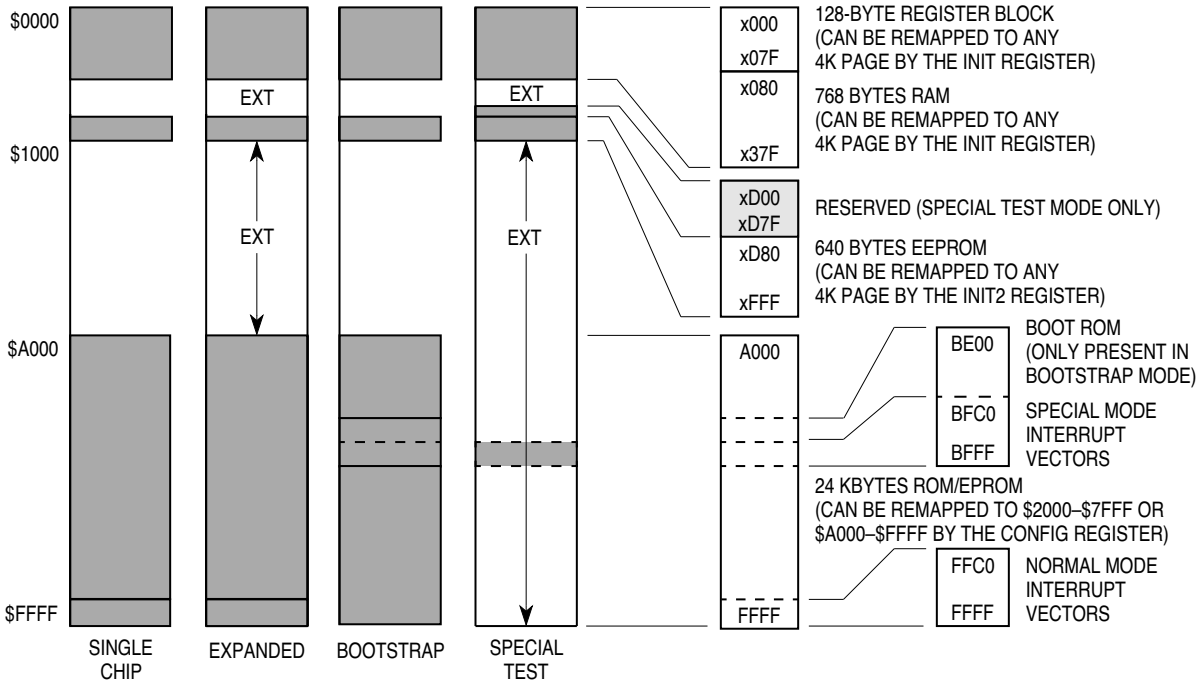
XDV [1:0]	XOUT = EXTAL Divided By	Frequency at EXTAL = 8 MHz	Frequency at EXTAL = 12 MHz	Frequency at EXTAL = 16 MHz
0 0	1	8 MHz	12 MHz	16 MHz
0 1	4	2 MHz	3 MHz	4 MHz
1 0	6	1.3 MHz	2 MHz	2.7 MHz
1 1	8	1 MHz	1.5 MHz	2 MHz

3 On-Chip Memory

In general, K-series MCUs have 768 bytes RAM, 640 bytes EEPROM, and 24 Kbytes ROM/EPROM. Some devices in the series have portions of their memory resources disabled. Some have ROM and some have EPROM replacing ROM. The following paragraphs describe the memory systems of devices in the series.

3.1 Memory Map and Register Block

The INIT, INIT2, and CONFIG registers control the presence and location of the registers, RAM, EEPROM, and ROM/EPROM in the 64 Kbyte CPU address space. The 128-byte register block originates at $0000 after reset and can be placed at any 4 Kbyte boundary ($x000) after reset by writing an appropriate value to the INIT register. Refer to **Figure 4**.



NOTE: ROM/EPROM can be enabled in special test mode by setting ROMON bit in the config register after reset.

Figure 4 Memory Map

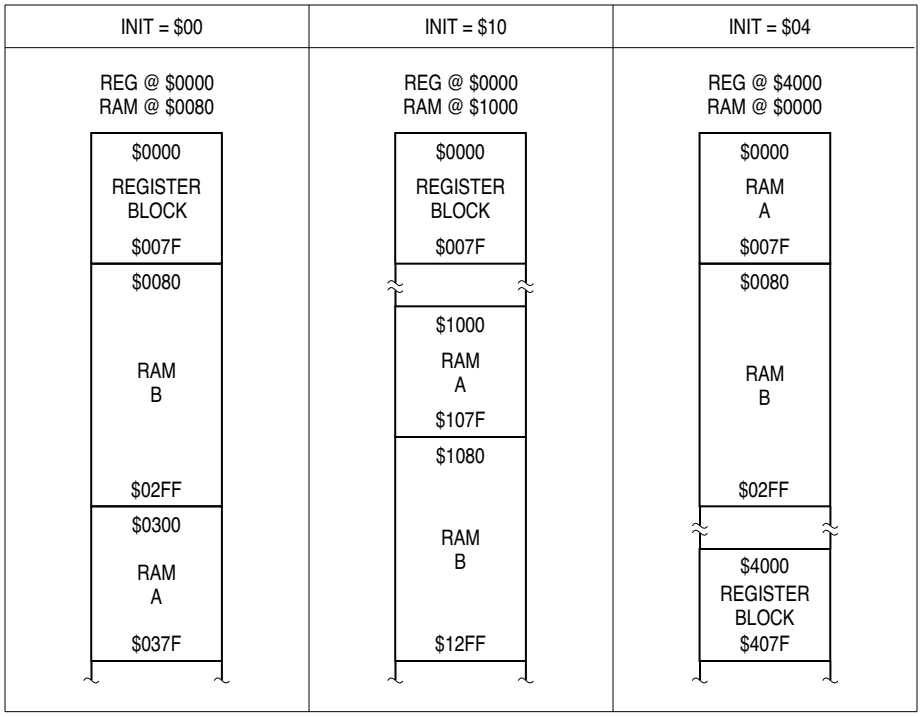


Figure 5 RAM and Register Mapping

Table 4 M68HC11 K Series Register and Control Bit Assignments

(Can be remapped to any 4-Kbyte boundary)

	Bit 7	6	5	4	3	2	1	Bit 0	
$0000	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PORTA
$0001	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
$0002	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
$0003	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	DDRF
$0004	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
$0005	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0	PORTF
$0006	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	PORTC
$0007	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
$0008	0	0	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
$0009	0	0	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
$000A	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0	PORTE
$000B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	CFORC
$000C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	OC1M
$000D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	OC1D
$000E	Bit 15	14	13	12	11	10	9	Bit 8	TCNT (High)
$000F	Bit 7	6	5	4	3	2	1	Bit 0	TCNT (Low)
$0010	Bit 15	14	13	12	11	10	9	Bit 8	TIC1 (High)
$0011	Bit 7	6	5	4	3	2	1	Bit 0	TIC1 (Low)
$0012	Bit 15	14	13	12	11	10	9	Bit 8	TIC2 (High)
$0013	Bit 7	6	5	4	3	2	1	Bit 0	TIC2 (Low)
$0014	Bit 15	14	13	12	11	10	9	Bit 8	TIC3 (High)

Table 4 M68HC11 K Series Register and Control Bit Assignments (Continued)

(Can be remapped to any 4-Kbyte boundary)

	Bit 7	6	5	4	3	2	1	Bit 0	
$0015	Bit 7	6	5	4	3	2	1	Bit 0	TIC3 (Low)
$0016	Bit 15	14	13	12	11	10	9	Bit 8	TOC1(High)
$0017	Bit 7	6	5	4	3	2	1	Bit 0	TOC1 (Low)
$0018	Bit 15	14	13	12	11	10	9	Bit 8	TOC2 (High)
$0019	Bit 7	6	5	4	3	2	1	Bit 0	TOC2 (Low)
$001A	Bit 15	14	13	12	11	10	9	Bit 8	TOC3 (High)
$001B	Bit 7	6	5	4	3	2	1	Bit 0	TOC3 (Low)
$001C	Bit 15	14	13	12	11	10	9	Bit 8	TOC4 (High)
$001D	Bit 7	6	5	4	3	2	1	Bit 0	TOC4 (Low)
$001E	Bit 15	14	13	12	11	10	9	Bit 8	TI4/O5 (High)
$001F	Bit 7	6	5	4	3	2	1	Bit 0	TI4/O5 (Low)
$0020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1
$0021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2
$0022	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I	TMSK1
$0023	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F	TFLG1
$0024	TOI	RTII	PAOVI	PAII	0	0	PR1	PR0	TMSK2
$0025	TOF	RTIF	PAOVF	PAIF	0	0	0	0	TFLG2
$0026	0	PAEN	PAMOD	PEDGE	0	I4/O5	RTR1	RTR0	PACTL
$0027	Bit 7	6	5	4	3	2	1	Bit 0	PACNT
$0028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
$0029	SPIF	WCOL	0	MODF	0	0	0	Bit 0	SPSR
$002A	Bit 7	6	5	4	3	2	1	Bit 0	SPDR
$002B	MBE	0	ELAT	EXCOL	EXROW	T1	T0	EPGM	EPROG*
$002C	0	0	0	0	HPPUE	GPPUE	FPPUE	BPPUE	PPAR
$002D	0	0	PGAR5	PGAR4	PGAR3	PGAR2	PGAR1	PGAR0	PGAR
$002E									Reserved
$002F									Reserved
$0030	CCF	0	SCAN	MULT	CD	CC	CB	CA	ADCTL
$0031	Bit 7	6	5	4	3	2	1	Bit 0	ADR1
$0032	Bit 7	6	5	4	3	2	1	Bit 0	ADR2
$0033	Bit 7	6	5	4	3	2	1	Bit 0	ADR3
$0034	Bit 7	6	5	4	3	2	1	Bit 0	ADR4
$0035	BULKP	LVPEN	BPRT4	PTCON	BPRT3	BPRT2	BPRT1	BPRT0	BPROT
$0036									Reserved
$0037	EE3	EE2	EE1	EE0	0	0	0	0	INIT2
$0038	LIRDV	CWOM	0	IRVNE	LSBF	SPR2	XDV1	XDV0	OPT2
$0039	ADPU	CSEL	IRQE	DLY	CME	FCME	CR1	CR0	OPTION
$003A	Bit 7	6	5	4	3	2	1	Bit 0	COPRST
$003B	ODD	EVEN	LVPI	BYTE	ROW	ERASE	EELAT	EEPGM	PPROG
$003C	RBOOT	SMOD	MDA	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO
$003D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT
$003E	TILOP	0	OCCR	CBYP	DISR	FCM	FCOP	0	TEST1
$003F	ROMAD	1	CLKX	PAREN	NOSEC	NOCOP	ROMON	EEON	CONFIG
$0040									Reserved
to									
$0055									Reserved
$0056	MXGS2	MXGS1	W2SZ1	W2SZ0	0	0	W1SZ1	W1SZ0	MMSIZ
$0057	W2A15	W2A14	W2A13	0	W1A15	W1A14	W1A13	0	MMWBR

Table 4 M68HC11 K Series Register and Control Bit Assignments (Continued)

(Can be remapped to any 4-Kbyte boundary)

	Bit 7	6	5	4	3	2	1	Bit 0	
$0058	0	X1A18	X1A17	X1A16	X1A15	X1A14	X1A13	0	MM1CR
$0059	0	X2A18	X2A17	X2A16	X2A15	X2A14	X2A13	0	MM2CR
$005A	IOSA	IOSB	GP1SA	GP1SB	GP2SA	GP2SB	PCSA	PCSB	CSCSTR
$005B	IOEN	IOPL	IOCSA	IOSZ	GCSPR	PCSEN	PCSZA	PCSZB	CSCTL
$005C	G1A18	G1A17	G1A16	G1A15	G1A14	G1A13	G1A12	G1A11	GPCS1A
$005D	G1DG2	G1DPC	G1POL	G1AV	G1SZA	G1SZB	G1SZC	G1SZD	GPCS1C
$005E	G2A18	G2A17	G2A16	G2A15	G2A14	G2A13	G2A12	G2A11	GPCS2A
$005F	0	G2DPC	G2POL	G2AV	G2SZA	G2SZB	G2SZC	G2SZD	GPCS2C
$0060	CON34	CON12	PCKA2	PCKA1	0	PCKB3	PCKB2	PCKB1	PWCLK
$0061	PCLK4	PCLK3	PCLK2	PCLK1	PPOL4	PPOL3	PPOL2	PPOL1	PWPOL
$0062	Bit 7	6	5	4	3	2	1	Bit 0	PWSCAL
$0063	TPWSL	DISCP	0	0	PWEN4	PWEN3	PWEN2	PWEN1	PWEN
$0064	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT1
$0065	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT2
$0066	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT3
$0067	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT4
$0068	Bit 7	6	5	4	3	2	1	Bit 0	PWPER1
$0069	Bit 7	6	5	4	3	2	1	Bit 0	PWPER2
$006A	Bit 7	6	5	4	3	2	1	Bit 0	PWPER3
$006B	Bit 7	6	5	4	3	2	1	Bit 0	PWPER4
$006C	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY1
$006D	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY2
$006E	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY3
$006F	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY4
$0070	BTST	BSPL	0	SBR12	SBR11	SBR10	SBR9	SBR8	SCBDH
$0071	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	SCBDL
$0072	LOOPS	WOMS	0	M	WAKE	ILT	PE	PT	SCCR1
$0073	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2
$0074	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF	SCSR1
$0075	0	0	0	0	0	0	0	RAF	SCSR2
$0076	R8	T8	0	0	0	0	0	0	SCDRH
$0077	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0	SCDRL
$0078									Reserved
to									
$007B									Reserved
$007C	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0	PORTH
$007D	DDH7	DDH6	DDH5	DDH4	DDH3	DDH2	DDH1	DDH0	DDRH
$007E	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	PORTG
$007F	DDG7	DDG6	DDG5	DDG4	DDG3	DDG2	DDG1	DDG0	DDRG

*MC68HC711K4 only.

3.2 RAM

All members of the M68HC11 K series have 768 bytes of static RAM. The RAM can be mapped to any 4-Kbyte boundary. Upon reset, the RAM is mapped at $0080–$037F. The registers are also mapped to this 4-Kbyte boundary. In previous versions of the M68HC11 devices the register block being mapped to the same boundary would cause the portion of RAM overlapped by the register block to be lost. However, a new RAM remapping feature has been added which automatically allows all of the RAM to be accessible even if the register block overlaps the RAM. Because the registers are located in the same

4-Kbyte boundary after reset, 128 bytes of the RAM are located at $0300 to $037F. Remapping is accomplished by writing appropriate values to the INIT register. Refer to the register and RAM mapping examples following the memory map diagram.

When power is removed from the MCU, RAM contents may be preserved using the MODB/V_{STBY} pin. A power source (2.0 Vdc –V_{DD}) applied to this pin protects all 768 bytes of RAM.

INIT — RAM and Register Mapping **$003D**

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

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

RAM[3:0] —Internal RAM Map Position

These bits determine the upper four bits of the RAM address. At reset RAM is mapped to $0000. Normally the RAM would be mapped at $0000–$02FF (768 bytes). However, the register block overlaps the first 128 bytes of RAM, causing them to be remapped to $0300–$037F. Refer to **Figure 4** and **Figure 5**.

REG[3:0] —128-Byte Register Block Map Position

These bits determine the upper four bits of the register block starting address. At reset registers are mapped to $0000 and overlap the first 128 bytes of RAM, causing them to be remapped to $0300–$037F. Refer to **Figure 4** and **Figure 5**.

3.3 ROM/EPROM

Standard devices have 24 kbytes of EPROM (OTPROM in a non-windowed package). Custom ROM devices have a 24-Kbyte ROM array that is mask programmed at the factory to customer specifications. The MC68HC11K0, MC68HC11K1, MC68L11K0, and MC68L11K1 have no ROM/EPROM. Refer to the **ordering information tables**.

The ROMAD and ROMON control bits in the CONFIG register control the position and presence of ROM/EPROM in the memory map. The ROM/EPROM can be mapped at $2000–$7FFF or $A000–$FFFF. If it is mapped to $A000–$FFFF, vector space is included. In single-chip mode the ROM/EPROM is forced to $A000–$FFFF (ROMAD = 1) and enabled (ROMON = 1), regardless of the value in the CONFIG register. This ensures that there will be ROM/EPROM at the vector space. In special test mode, the ROMON bit is forced to zero so that the ROM/EPROM is removed from the memory map. Refer to **Figure 4**.

Programming EPROM requires an external 12.25 volt nominal power supply (V_{PPE}) that must be applied to the $\overline{XIRQ}$/V_{PPE} pin. Three methods are used to program and verify EPROM/OTPROM.

Normal EPROM/OTPROM programming can be accomplished in any operating mode. Normal programming is accomplished using the EPROM/OTPROM programming register (EPROG). The EPROG register enables the EPROM programming voltage, controls the latching of data to be programmed, and selects single- or multiple-byte programming.

To program the EPROM, complete the following steps using the EPROG register:

1. Set the ELAT bit in EPROG register. EELAT bit in PPROG must be cleared as it negates the function of the ELAT bit.
2. Write data to the desired address.
3. Turn on programming voltage to the EPROM array by setting the EPGM bit in EPROG register.
4. Delay for 2 ms or more, as appropriate.
5. Clear the EPGM bit in EPROG to turn off the programming voltage.

6. Clear the EPROG register to reconfigure the EPROM address and data buses for normal operation.

In EPROM emulation mode (PROG mode), the EPROM/OTPROM is programmed as a stand-alone EPROM by adapting the MCU footprint to the 27C256-type EPROM and using an appropriate EPROM programmer. To put the MCU in PROG mode, pull the following pins low: MODA/$\overline{LIR}$, MODB/V_{STBY}, $\overline{RESET}$, PA[2:0]. Refer to **Figure 6**.

In the third method, the EPROM is programmed by software while in the special test or bootstrap modes. User-developed software can be uploaded through the SCI, or a ROM resident EPROM programming utility can be used. To use the resident utility, bootload a three-byte program consisting of a single jump instruction to $BF00. $BF00 is the starting address of a resident EPROM programming utility. The utility program sets the X and Y index registers to default values, then receives programming data from an external host and programs it into EPROM. The value in IX determines programming delay time. The value in IY is a pointer to the first address in EPROM to be programmed (default = $A000).

When the utility program is ready to receive programming data, it sends the host the $FF character. Then it waits. When the host sees the $FF character, the EPROM programming data is sent, starting with the first location in the EPROM array. 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.

Although the external 12.25 V programming voltage must be applied to the $\overline{XIRQ}$/V_{PPE} pin during EPROM programming, it should be equal to V_{DD} before verifying the data that was just programmed. It should equal V_{DD} during normal operation also. The $\overline{XIRQ}$/V_{PPE} pin has a high voltage detect circuit that inhibits assertion of the ELAT bit when programming voltage is at low levels.

CAUTION

If the MCU is used in any operating mode while high voltage (12.25 V nominal) is present on the $\overline{XIRQ}$/V_{PPE} pin, the $\overline{IRQ}$/$\overline{CE}$ pin must be pulled high to avoid accidental programming or corruption of EPROM contents. After programming an EPROM location, $\overline{IRQ}$ pin must also be pulled high before the address and data are changed to program the next location.

EPROG — EPROM Programming Control **$002B**

	Bit 7	6	5	4	3	2	1	Bit 0
	MBE	—	ELAT	EXCOL	EXROW	—	—	EPGM
RESET:	0	0	0	0	0	0	0	0

MBE —Multiple-Byte Programming Enable

0 = EPROM array configured for normal programming

1 = Program two bytes with the same data

When multiple-byte programming is enabled, address bit 5 is considered a don't care so that bytes with address bit 5 = 0 and address bit 5 = 1 both get programmed. MBE can be read in any mode and always reads zero in normal modes. MBE can only be written in special modes.

Bit 6 —Not implemented

Always reads zero

ELAT —EPROM Latch Control

ELAT can be read any time. ELAT can be written any time except when EPGM = 1, then the write to ELAT will be disabled. When ELAT = 1, writes to EPROM cause address and data to be latched and the EPROM cannot be read.

0 = EPROM address and data bus configured for normal reads

1 = EPROM address and data bus configured for programming

EXCOL —Select Extra Columns

0 = User array selected

1 = User array is disabled and extra columns are accessed at bits [7:0]. Addresses use bits [11:5] and bits [4:0] are don't care. EXCOL can only be read in special modes and always returns zero in normal modes. EXCOL can be written in special modes only.

EXROW —Select Extra Rows

0 = User array selected

1 = User array is disabled and two extra rows are available. Addresses use bits [5:0] and bits [11:6] are don't care. EXROW can only be read in special modes and always returns zero in normal modes. EXROW can be written in special modes only.

Bits [2:1] —Not implemented

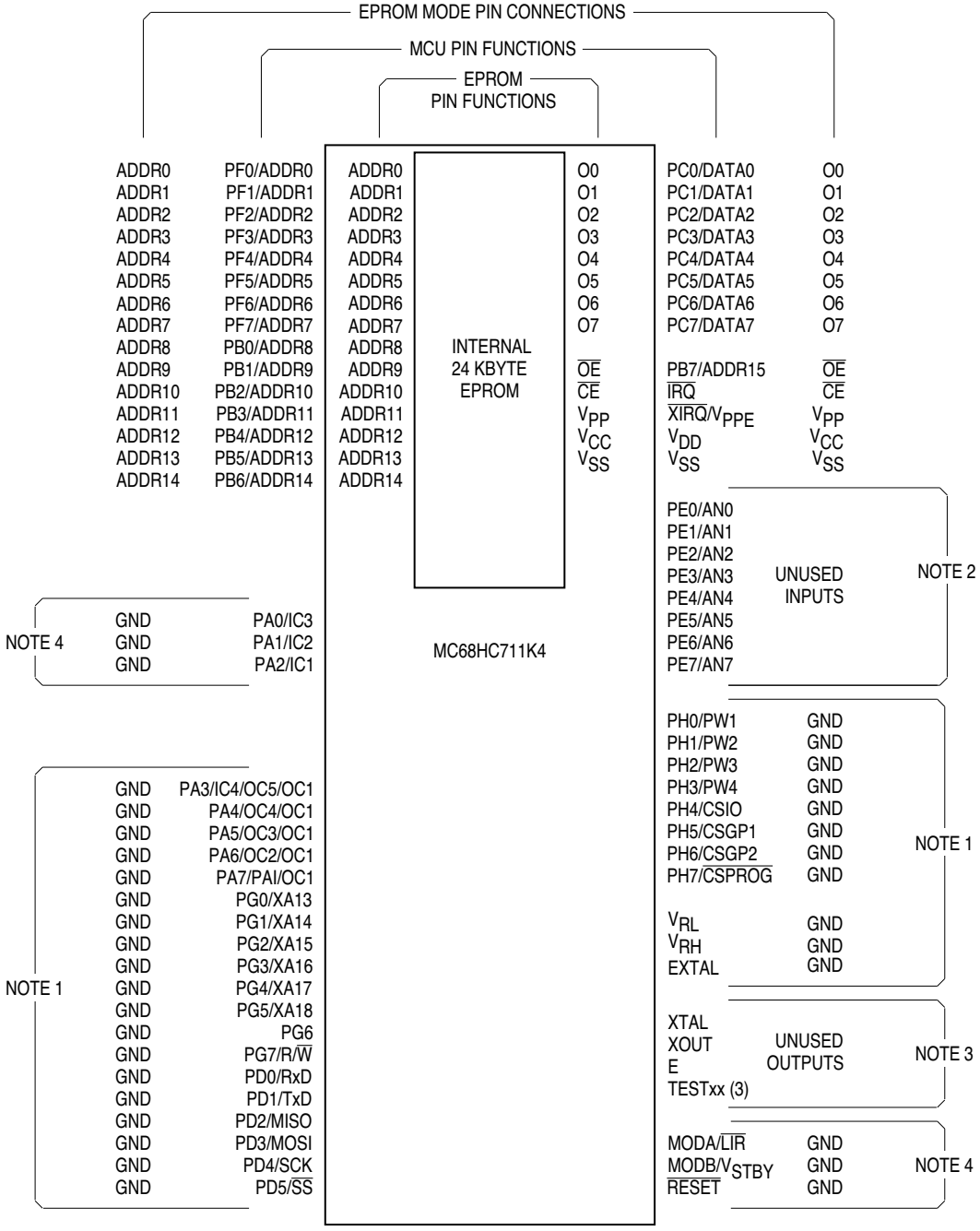
Always read zero

EPGM —EPROM Programming Voltage Enable

EPGM can be read any time and can only be written when ELAT = 1.

0 = Programming voltage to EPROM array disconnected

1 = Programming voltage to EPROM array connected



NOTES:

1. Unused Inputs – grounding is recommended.
2. Unused Inputs – these pins may be left unterminated.
3. Unused Outputs – these pins should be left unconnected.
4. Grounding these six pins configures the MC68HC711K4 for EPROM emulation mode.

Figure 6 Pin Assignments of the MC68HC711K4 MCU in PROG Mode

3.4 EEPROM

The 640-byte EEPROM is initially located at $0D80 after reset, assuming EEPROM is enabled in the memory map by the EEON bit in the CONFIG register. EEPROM can be placed at any 4-Kbyte boundary ($xD80) by writing appropriate values to the INIT2 register. Note that EEPROM can be mapped so that it contains the vector space. Refer to **Figure 4**. The MC68HC11K0, MC68HC11K3, MC68L11K0, and MC68L11K3 have no EEPROM. Refer to the **ordering information tables**.

Programming and erasing the EEPROM is controlled by the PPROG register, and dependent upon the block protect (BPROT) register value. An on-chip charge pump develops the high voltage required for programming and erasing. When the frequency of the E clock is less than 1 MHz, select the internal clock source to drive the EEPROM charge pump by writing one to the CSEL bit in the OPTION register.

The CONFIG register consists of a single EEPROM byte. Although the byte is not included in the 640-byte EEPROM array, programming the CONFIG register requires the same procedure as any byte in the array. The erased state of bits in the CONFIG register is logic one. Refer to the CONFIG register description that follows this section.

The erased state of an EEPROM byte is $FF (all ones).

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. Set the ERASE, EELAT, and appropriate BYTE and ROW bits in PPROG register.
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 done by writing to any location in the array.
3. Set the ERASE, EELAT, EEPGM, and appropriate BYTE and ROW bits in PPROG register.
4. Delay for 10 ms or more, as appropriate.
5. Clear the EEPGM bit in PPROG to turn off the programming voltage.
6. Clear the PPROG register to reconfigure the EEPROM address and data buses for normal operation.

To program the EEPROM, ensure the proper bits of the BPROT register are cleared and use the PPROG register to complete the following steps:

1. Set the EELAT bit in PPROG register.
2. Write data to the desired address.
3. Set EEPGM bit in PPROG.
4. Delay for 10 ms or more, as appropriate.
5. Clear the EEPGM bit in PPROG to turn off the programming voltage.
6. Clear the PPROG register to reconfigure the EEPROM address and data buses for normal operation.

CAUTION

Since it is possible to perform other operations while the EEPROM programming/erase operation is in progress, it is common to start the operation and then return to the main program until the 10 ms is completed. When the EELAT bit is set at the beginning of a program/erase operation, the EEPROM is electronically removed from the memory map; thus, it is not accessible during the program/erase cycle. Care must be taken to ensure that EEPROM resources will not be needed by any routines in the code during the 10 ms program/erase time.

PPROG —EEPROM Programming Control **$003B**

	Bit 7	6	5	4	3	2	1	Bit 0
	ODD	EVEN	LVPI	BYTE	ROW	ERASE	EELAT	EEPGM
RESET:	0	0	0	0	0	0	0	0

ODD —Program Odd Rows in Half of EEPROM (TEST)

EVEN —Program Even Rows in Half of EEPROM (TEST)

LVPI —Low Voltage Programming Inhibit

LVPI can be read at any time and writes to LVPI have no meaning nor effect. LVPI is set if LVPEN bit in BPROT register equals one and the LVPI circuit detects that V_{DD} has fallen below a safe operating voltage. Once set, LVPI is cleared when V_{DD} returns to a safe operating voltage or if LVPEN bit in BPROT register is cleared. If LVPEN equals zero, then LVPI is always zero and has no meaning nor effect.

0 = EEPROM programming enabled
1 = EEPROM programming disabled

BYTE —Byte/Other EEPROM Erase Mode

0 = Row or bulk erase mode used
1 = Erase only one byte of EEPROM

ROW —Row/All EEPROM Erase Mode (only valid when BYTE = 0)

0 = All 640 bytes of EEPROM erased
1 = Erase only one 16-byte row of EEPROM

BYTE	ROW	Action
0	0	Bulk Erase (All 640 Bytes)
0	1	Row Erase (16 Bytes)
1	0	Byte Erase
1	1	Byte Erase

ERASE —Erase/Normal Control for EEPROM

0 = Normal read or program mode
1 = Erase mode

EELAT —EEPROM Latch Control

0 = EEPROM address and data bus configured for normal reads
1 = EEPROM address and data bus configured for programming or erasing

EEPGM —EEPROM Program Command

0 = Program or erase voltage switched off to EEPROM array
1 = Program or erase voltage switched on to EEPROM array

BPROT — Block Protect **$0035**

	Bit 7	6	5	4	3	2	1	Bit 0
	BULKP	LVPEN	BPRT4	PTCON	BPRT3	BPRT2	BPRT1	BPRT0
RESET:	1	1	1	1	1	1	1	1

NOTE

Block protect register bits can be written to zero (protection disabled) only once within 64 cycles of a reset in normal modes, or at any time in special modes. Block protect register bits can be written to one (protection enabled) at any time.

BULKP —Bulk Erase of EEPROM Protect

0 = EEPROM can be bulk erased normally
1 = EEPROM cannot be bulk or row erased

LVPEN —Low Voltage Programming Protect Enable

If LVPEN = 1, programming of the EEPROM is enabled unless the LVPI circuit detects that V_{DD} has fallen below a safe operating voltage, thus setting the low voltage programming inhibit bit in PPROG register (LVPI = 1).

0 = Low voltage programming protect for EEPROM disabled
1 = Low voltage programming protect for EEPROM enabled

BPRT4 —Block Protect Bit for Upper 128 Bytes of EEPROM

Refer to description for BPRT[3:0].

PTCON —Protect for CONFIG

0 = CONFIG register can be programmed or erased normally
1 = CONFIG register cannot be programmed or erased

BPRT[3:0] —Block Protect Bits for EEPROM

0 = Protection disabled
1 = Protection enabled

Bit Name	Block Protected	Block Size
BPRT4	$xF80–$xFFF	128 Bytes
BPRT3	$xE60–$xF7F	288 Bytes
BPRT2	$xDE0–$xE5F	128 Bytes
BPRT1	$xDA0–$xDDF	64 Bytes
BPRT0	$xD80–$xD9F	32 Bytes

INIT2 —EEPROM Mapping **$0037**

	Bit 7	6	5	4	3	2	1	Bit 0
	EE3	EE2	EE1	EE0	0	0	0	0
RESET:	0	0	0	0	0	0	0	0

INIT2 can be written only once in normal modes, any time in special modes.

EE[3:0] —EEPROM Map Position

EEPROM is at $xD80–$xFFF, where x is the hexadecimal digit represented by EE[3:0].

Bits [3:0] —Not implemented

Always read zero

3.5 Configuration Control Register (CONFIG)

The CONFIG register is used to define several system functions. Although the CONFIG register is an address within the register block, it is actually an EEPROM byte with the address of $x03F. CONFIG is made up of EEPROM cells and static latches. The operation of the MCU is controlled directly by these latches and not the actual EEPROM byte. When programming the CONFIG register, the EEPROM byte is being accessed. When the CONFIG register is being read, the static latches are being accessed.

The CONFIG register can be read at any time. The value read is the one latched from the EEPROM cells during the last reset sequence. A new value programmed into this register cannot be read until a subsequent reset occurs. Unused bits always read as ones.

In normal modes (SMOD = 0), CONFIG bits can only be written using the EEPROM programming sequence, and are neither readable nor active until latched via the next reset. In special modes (SMOD = 1), CONFIG bits can be written at any time.

CONFIG —System Configuration Register **$003F**

	Bit 7	6	5	4	3	2	1	Bit 0
	ROMAD	1	CLKX	PAREN	NOSEC	NOCOP	ROMON	EEON
RESET:	—	1	—	—	—	—	—	—

ROMAD —ROM/EPROM Mapping Control

In single-chip mode ROMAD is forced to one out of reset.

0 = ROM/EPROM located at $2000–$7FFF

1 = ROM/EPROM located at $A000–$FFFF

Bit 6 —Not implemented

Always reads one

CLKX —XOUT Clock Enable

0 = XOUT pin disabled

1 = Buffered XTAL signal (four times E frequency) driven out on the XOUT pin

PAREN —Pull-Up Assignment Register Enable

0 = Pull-ups always disabled regardless of state of bits in PPAR

1 = Pull-ups either enabled or disabled through PPAR

NOSEC —Security Disable

NOSEC is invalid unless the security mask option is specified before the MCU is manufactured. If security mask option is omitted NOSEC always reads one. Refer to **3.6 Security Feature**.

0 = Security enabled

1 = Security disabled

NOCOP —COP System Disable

Resets to programmed value

0 = COP enabled (forces reset on timeout)

1 = COP disabled (does not force reset on timeout)

ROMON —ROM/EPROM Enable

In single-chip mode, ROMON is forced to one out of reset. In special test mode, ROMON is forced to zero out of reset.

0 = ROM/EPROM removed from memory map

1 = ROM/EPROM present in memory map

EEON —EEPROM Enable

0 = EEPROM disabled from memory map

1 = EEPROM present in memory map with location depending on value specified in EE[3:0] in INIT2

3.6 Security Feature

The security feature protects memory contents from unauthorized access. Although many devices in the M68HC11 family support the security feature, an enhancement has been added to the MC68S11K4 that protects the contents of EPROM/OTPROM.

The security feature affects how the MCU behaves in certain modes. When the optional security feature has been specified prior to manufacture and enabled via the NOSEC bit in CONFIG, the MCU is restricted to operation in single-chip modes only. When the NOSEC bit equals zero, the MCU ignores the state of the MODA pin during reset. This allows the MCU to be operated in single-chip and bootstrap modes only. These modes of operation do not allow external visibility of the internal address and data buses. Although the security feature can easily be disabled when in bootstrap mode, the bootloader firmware residing in bootstrap ROM checks to see if the NOSEC bit is clear. If NOSEC is clear (security enabled), the bootloader program performs the following:

- Output $FF on SCI transmitter.
- Erase EEPROM array.
- Verify that EEPROM has been erased. If it has not, repeat erase procedure.
- Write $FF to every location in RAM.
- Check EPROM for data. If data is present, stay in loop. Otherwise proceed.
- Erase the CONFIG register.
- Continue executing bootloader routine.

Notice that the bootloader routine checks the EPROM to see if it contains any data. The presence of data causes the routine to stay in a loop. At this time, devices with the security enhancement are only available as one-time-programmable (OTP) MCUs in non-windowed packages. Once they have been programmed and secured, they will not function in bootstrap mode.

For more information refer to *M68HC11 Reference Manual* (M68HC11RM/AD).

4 Memory Expansion and Chip Selects

Two additional on-chip blocks are provided with the M68HC11 K-series MCUs. The first block implements additional address lines that become active only when required by the CPU. The second block provides chip-select signals that simplify the interface to external peripheral devices. Both of these blocks are fully programmable by values written to associated control registers.

4.1 Memory Expansion

New to the M68HC11 family of microcontrollers is the ability of the M68HC11 K-series MCUs to extend the address range of the M68HC11 CPU beyond the physical 64 Kbyte limit of the 16 CPU address lines. The following is a brief description of how the extended addressing is achieved. For a more detailed discussion refer to application note *Using the MC68HC11K4 Memory Mapping Logic* (AN452/D).

Memory expansion is achieved by manipulating the CPU address lines such that, even though the CPU cannot distinguish more than 64 Kbytes of physical memory, up to 1 Mbyte can be accessed through a paged memory scheme. Additional address lines XA[18:13] are provided as alternate functions of port G pins. Bits in the port G assignment register (PGAR) define which port G pins are to be used for memory expansion address lines and which are to be used for general-purpose I/O.

In order to access expanded memory, the user must first allocate a range of the 64 Kbyte address space to be used for the window(s) through which external expanded memory is viewed by the CPU. The size and placement of the window(s) depend upon values written to the MMSIZ and MMWBR registers, respectively. Which bank or page of the expanded memory that is present in the window(s) at a given time is dependent upon values written to the MM1CR and MM2CR registers.

Up to two windows can be designated and each can be programmed to 0 (disabled), 8, 16, or 32 Kbytes. The base address for each window must be an integer multiple of the window size. When the window size is 32 Kbytes, the base address can be at $0000, $4000, or $8000.

If the windows are defined in such a way that they overlap, bank window 1 has priority and the part of window 2 that is not overlapped by bank window 1 remains active. If a window is defined such that it overlaps any internal registers, RAM, or EEPROM, the portion of the registers, RAM, or EEPROM that is overlapped is repeated in all banks associated with that window. However, if ROM/EPROM is enabled and overlapped by a window, the ROM/EPROM is present only in banks with XA[18:16] = 0:0:0.

Expanded memory is addressed by using a combination of the CPU's normal address lines ADDR[15:0] and the expansion address lines XA[18:13]. Window size and the number of banks associated with the window determine exactly which address lines are used. The additional address lines (XA[18:13]) determine which bank is present in a window at a given time. The lower three expansion address lines (XA[15:13]) are used only when needed by the CPU and replace the CPU's equivalent address lines (ADDR[15:13]). The following tables show which address lines are used for various configurations of expanded memory.

Five registers control operation of the memory expansion function. MM1CR and MM2CR registers indicate which bank of a window is active. Each contains the value to be output when the CPU selects addresses within the memory expansion window. PGAR selects which pins are used for I/O or memory expansion address lines, defining which extended address lines are used. The MMWBR register defines the starting address of each of the two windows within the CPU 64-Kbyte address range. The MMSIZ register sets the size of the windows in use and selects whether the on-board general-purpose chip selects are active for CPU addresses or for expansion addresses.

Table 5 CPU Address and Address Expansion Signals

Number of Banks	Window Size			
	8 Kbytes	16 Kbytes	32 Kbytes	32 Kbytes (Window Based at $4000)
2	ADDR[12:0] XA13	ADDR[13:0] XA14	ADDR[14:0] XA15	ADDR[13:0] XA[15:14]
4	ADDR[12:0] XA[14:13]	ADDR[13:0] XA[15:14]	ADDR[14:0] XA[16:15]	ADDR[13:0] XA[16:14]
8	ADDR[12:0] XA[15:13]	ADDR[13:0] XA[16:14]	ADDR[14:0] XA[17:15]	ADDR[13:0] XA[17:14]
16	ADDR[12:0] XA[16:13]	ADDR[13:0] XA[17:14]	ADDR[14:0] XA[18:15]	ADDR[13:0] XA[18:14]
32	ADDR[12:0] XA[17:13]	ADDR[13:0] XA[18:14]	— —	— —
64	ADDR[12:0] XA[18:13]	— —	— —	— —

PGAR — Port G Assignment **$002D**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	PGAR5	PGAR4	PGAR3	PGAR2	PGAR1	PGAR0
RESET:	0	0	0	0	0	0	0	0

Bits [7:6] — Not implemented
Always read zero

PGAR[5:0] —Port G Pin Assignment Bits [5:0]
0 = Corresponding port G pin is general-purpose I/O
1 = Corresponding port G pin is address line, XA[18:13]

NOTE

A special case exists for expansion address lines XA[15:13] that overlap the CPU address lines ADDR[15:13]. If these lines are selected as expansion address lines in PGAR, but are not used in either window, the corresponding CPU address line is output on the appropriate port G pin.

MMSIZ — Memory Mapping Size **$0056**

	Bit 7	6	5	4	3	2	1	Bit 0
	MXGS2	MXGS1	W2SZ1	W2SZ0	—	—	W1SZ1	W1SZ0
RESET:	0	0	0	0	0	0	0	0

MXGS[2:1] — Memory Expansion Select for General-Purpose Chip Select 2 or 1
0 = General-purpose chip select 2 or 1 based on 64 Kbyte CPU address
1 = General-purpose chip select 2 or 1 based on expansion address

W2SZ[1:0] — Window 2 Size
These bits select the size of memory expansion window 2. Refer to the table following W1SZ[1:0].

Bits [3:2] — Not implemented
Always read zero

W1SZ[1:0] —Window 1 Size

These bits select the size of memory expansion window 1.

WxSZ[1:0]	Window Size
0 0	Window disabled
0 1	8 K —Window can have up to 64 8-Kbyte banks
1 0	16 K —Window can have up to 32 16-Kbyte banks
1 1	32 K —Window can have up to 16 32-Kbyte banks

MMWBR — Memory Mapping Window Base **$0057**

	Bit 7	6	5	4	3	2	1	Bit 0
$0057	W2A15	W2A14	W2A13	—	W1A15	W1A14	W1A13	—
RESET:	0	0	0	0	0	0	0	0

W2A[15:13] —Window 2 Base Address

Selects the three most significant bit (MSB) of the base address for memory mapping window 2. Refer to the table following W1A[15:13].

Bit 4 —Not implemented

Always reads zero

W1A[15:13] —Window Base 1 Address

Selects the three MSB of the base address for memory mapping window 1. Refer to the following table for additional information.

MSB Bits	Window Base Address		
WxA[15:13]	8 K	16 K	32 K
0 0 0	$0000	$0000	$0000
0 0 1	$2000	$0000	$0000
0 1 0	$4000	$4000	$4000
0 1 1	$6000	$4000	$4000
1 0 0	$8000	$8000	$8000
1 0 1	$A000	$8000	$8000
1 1 0	$C000	$C000	$8000
1 1 1	$E000	$C000	$8000

Bit 0 —Not implemented

Always reads zero

NOTE

A special case exists when the bank size is 32 Kbytes and the window base address is $4000. The XA14 signal connected to the ADDR14 pin of the memory device automatically drives an inverted CPU ADDR14 signal onto the XA14 pin when the window is active. The effect occurs while the CPU address is in the $4000–$BFFF range, the XA pins and external physical memory range is $0000–$7FFF.

MM1CR–MM2CR —Memory Mapping Window 1 and 2 Control **$0058–$0059**

	Bit 7	6	5	4	3	2	1	Bit 0	
$0058	—	X1A18	X1A17	X1A16	X1A15	X1A14	X1A13	—	MM1CR
$0059	—	X2A18	X2A17	X2A16	X2A15	X2A14	X2A13	—	MM2CR
RESET:	0	0	0	0	0	0	0	0	

Bit 7 — Not implemented
Always reads zero

MM1CR — Memory Mapping Window 1 Control Register
When a 64 Kbyte CPU address falls within window 1, the value in MM1CR is driven out from the corresponding expansion address lines to enable the specified bank in the window.

MM2CR — Memory Mapping Window 2 Control Register
When a 64 Kbyte CPU address falls within window 2, the value in MM2CR is driven out from the corresponding expansion address lines to enable the specified bank in the window.

Bit 0 — Not implemented
Always reads zero

4.2 Overlap Guidelines

- On-chip registers, RAM, and EEPROM are higher priority than expansion windows. If a window overlaps RAM, registers, or EEPROM, they appear in all banks at their CPU address.
- If a window overlaps on-chip ROM/EPROM, the ROM/EPROM appears only in banks with XA[18:16] = 0:0:0.
- Window 1 is higher priority than window 2, therefore any overlapped portion of window 2 is inaccessible.

4.3 Chip Selects

M68HC11 K-series MCUs have four software configured chip selects that are enabled in expanded modes. The chip select for I/O (CSIO) is used for I/O expansion. The program chip select ($\overline{\text{CSPROG}}$) is used with an external memory that contains the reset vectors and program. The two general-purpose chip selects, CSGP1 and CSGP2, are used to enable external devices. These external devices can be in the 64 Kbyte memory space or in the expanded memory space. Chip select signals are a shared function of port H. When an MCU pin is not used for chip select functions it can be used for general-purpose I/O. The following table contains a summary of the attributes of each chip select that can be controlled by user software.

CSIO	Enable	IOEN in CSCTL —1 = On, off at reset (0)
	Valid	IOCSA in CSCTL —1 = Address valid, 0 = E valid
	Polarity	IOPL in CSCTL —1 = Active high, 0 = Active low
	Size	IOSZ in CSCTL —1 = 4K ($1000–$1FFF), 0 = 8K ($0000–$1FFF)
	Start Address	Fixed (see Size)
	Stretch	IO1SA:IO1SB in CSCSTR —0, 1, 2, or 3 E clocks

CSPROG	Enable	PSCEN in CSCTL —1 = On, ON at reset
	Valid	Fixed (Address valid)
	Polarity	Fixed (Active low)
	Size	PCSZA:PCSZB in CSCTL — 0:0 = 64K ($0000–$FFFF) 0:1 = 32K ($8000–$FFFF) 1:0 = 16K ($C000–$FFFF) 1:1 = 8K ($E000–$FFFF)
	Start Address	Fixed (see Size)
	Stretch	PCSA:PCSB in CSCSTR —0, 1, 2, or 3 E clocks
	Priority	GCSPR in CSCTL — 1 = CSGPx above $\overline{\text{CSPROG}}$ 0 = $\overline{\text{CSPROG}}$ above CSGPx

CSGP1, CSGP2	Enable	Set size to 0K to disable
	Valid	GxPOL in GPCS1C (GPCS2C) —1 = Address valid, 0 = E valid
	Polarity	GxAV in GPCS1C (GPCS2C) —1 = Active high, 0 = Active low
	Size	Refer to GPCS1C (GPCS2C) —2K to 512K in nine steps, 0K = disable, can also follow memory expansion window 1 or window 2
	Start Address	Refer to GPCS1A (GPCS2A)
	Stretch	Refer to CSCSTR —0, 1, 2, or 3 E clocks
	Other	G1DG2 in GPCS1C allows CSGP1 and CSGP2 to be connected to an internal OR gate and driven out the CSGP2 pin. G1DPC in GPCS1C allows CSGP1 and $\overline{\text{CSPROG}}$ to be connected to an internal OR gate and driven out the $\overline{\text{CSPROG}}$ pin. G2DPC in GPCS2C allows CSGP2 and $\overline{\text{CSPROG}}$ to be connected to an internal OR gate and driven out the $\overline{\text{CSPROG}}$ pin. MXGS2 in MMSIZ allows CSGP2 to follow either 64K CPU addresses or 512K expansion addresses. MXGS1 in MMSIZ allows CSGP1 to follow either 64K CPU addresses or 512K expansion addresses.

4.3.1 Program Chip Select ($\overline{\text{CSPROG}}$)

The program chip select ($\overline{\text{CSPROG}}$) is active in the range of memory where the main program exists. $\overline{\text{CSPROG}}$ is enabled out of reset in all modes. After reset in normal mode, the PCS stretch select bit is set to provide one cycle of stretch so that slow memory devices can be used.

4.3.2 I/O Chip Select (CSIO)

The I/O chip select (CSIO) is programmable for a four Kbyte size located at addresses $1000 to $1FFF or eight Kbyte size located at addresses $0000 to $1FFF. Polarity of the active state is programmable for active high or active low. Clock stretching can be set from zero to three cycles.

4.3.3 General-Purpose Chip Selects (CSGP1, CSGP2)

The general-purpose chip selects are the most flexible and programmable and have the most control bits. Polarity of active state, E valid or address valid, size, and starting address are all programmable. Clock stretching can be set from zero to three cycles. Each chip select can be programmed to become active whenever the CPU address enters a memory expansion window regardless of the actual bank selected. This is known as following a window.

Each general purpose chip select can be configured to drive the program chip select. CSGP1 can be configured to drive CSGP2 or the program chip select. Using one chip select to drive another allows the same device to cover the address space defined by both chip selects. The two chip selects are connected to an internal OR gate. The output of the OR gate is then driven onto the pin corresponding to the driven chip select. For example, this is useful when the same external device is used with both bank windows but the windows are opened independently. In cases where one chip select drives another, determine the priority from the following table.

Condition	Priority
GPCS1 drives GPCS2	GPCS1
GPCS1 drives PCS	GPCS1
GPCS2 drives PCS	GPCS2
GPCS1 and GPCS2 drive PCS	GPCS1

4.3.4 Chip Select Priorities

To minimize chip select conflicts (with one another or with internal memory and registers), the priority is determined by the GCSPR bit in the CSCTL register. Refer to the following table.

GCSPR = 0	GCSPR = 1
On-Chip Registers	On-Chip Registers
On-Chip RAM	On-Chip RAM
Bootloader ROM	Bootloader ROM
On-Chip EEPROM	On-Chip EEPROM
On-Chip ROM/EPROM	On-Chip ROM/EPROM
I/O Chip Select	I/O Chip Select
Program Chip Select	GP Chip Select 1
GP Chip Select 1	GP Chip Select 2
GP Chip Select 2	Program Chip Select

4.3.5 Chip Select Control Registers

There are six chip select control registers. Chip select functions are enabled by control bits in CSCTL register. Chip selects are configured by bits in CSCSTR, IOEN, IOPL, IOCSA, and IOSZ registers.

CSCTL — Chip Select Control **$005B**

	Bit 7	6	5	4	3	2	1	Bit 0
	IOEN	IOPL	IOCSA	IOSZ	GCSPR	PCSEN	PCSZA	PCSZB
RESET:	0	0	0	0	0	1	0	0

IOEN —I/O Chip Select Enable
0 = CSIO disabled
1 = CSIO enabled

IOPL —I/O Chip Select Polarity Select
0 = CSIO active low
1 = CSIO active high

IOCSA —I/O Chip Select Address Valid
0 = Valid during E-clock high time
1 = Valid during address valid time

IOSZ —I/O Chip Select Size Select
0 = $1000–$1FFF (4 Kbyte)
1 = $0000–$1FFF (8 Kbyte)

GCSPR —General-Purpose Chip Select Priority
0 = Program chip select has priority over general-purpose chip selects
1 = General-purpose chip selects have priority over program chip select

PCSEN —Program Chip Select Enable
0 = $\overline{\text{CSPROG}}$ disabled
1 = $\overline{\text{CSPROG}}$ enabled

PCSZA, PCSZB —Program Chip Select Size (A or B)

PCSZA	PCSZB	Size (Bytes)	Address Range
0	0	64 K	$0000–$FFFF
0	1	32 K	$8000–$FFFF
1	0	16 K	$C000–$FFFF
1	1	8 K	$E000–$FFFF

CSCSTR —Chip Select Clock Stretch **$005A**

	Bit 7	6	5	4	3	2	1	Bit 0	
	IOSA	IOSB	GP1SA	GP1SB	GP2SA	GP2SB	PCSA	PCSB	
RESET:	0	0	0	0	0	0	0	1	Normal Modes
	0	0	0	0	0	0	0	0	Special Modes

IOSA, IOSB —CSIO Stretch Select

GP1SA, GP1SB —CSGP1 Stretch Select

GP2SA, GP2SB —CSGP2 Stretch Select

PCSA, PCSB —$\overline{\text{CSPROG}}$ Stretch Select

Bit [A:B]	Clock Stretch
0 0	None
0 1	1 Cycle
1 0	2 Cycles
1 1	3 Cycles

GPCS1A —General-Purpose Chip Select 1 Address **$005C**

	Bit 7	6	5	4	3	2	1	Bit 0
	G1A18	G1A17	G1A16	G1A15	G1A14	G1A13	G1A12	G1A11
RESET:	0	0	0	0	0	0	0	0

G1A[18:11] —General-Purpose Chip Select 1 Address
Selects the starting address of general-purpose chip select 1 range. Refer to the G1SZA–G1SZD table.

GPCS1C —General-Purpose Chip Select 1 Control **$005D**

	Bit 7	6	5	4	3	2	1	Bit 0
	G1DG2	G1DPC	G1POL	G1AV	G1SZA	G1SZB	G1SZC	G1SZD
RESET:	0	0	0	0	0	0	0	0

G1DG2 —General-Purpose Chip Select 1 Drives General-Purpose Chip Select 2
0 = CSGP1 does not affect CSGP2
1 = CSGP1 and CSGP2 are connected to an OR gate and driven out CSGP2

G1DPC —General-Purpose Chip Select 1 Drives Program Chip Select
0 = CSGP1 does not affect $\overline{\text{CSPROG}}$
1 = CSGP1 and $\overline{\text{CSPROG}}$ are connected to an OR gate and driven out $\overline{\text{CSPROG}}$

G1POL —General-Purpose Chip Select 1 Polarity Select
0 = CSGP1 active low
1 = CSGP1 active high

G1AV —General-Purpose Chip Select 1 Address Valid Select
0 = CSGP1 active during E high time
1 = CSGP1 active during address valid time

G1SZA–G1SZD —General-Purpose Chip Select 1 Size

G1SZx					Valid Bits	Valid Bits
A	B	C	D	Size (Bytes)	(MXGS1 = 0)	(MXGS1 = 1)
0	0	0	0	Disabled	None	None
0	0	0	1	2 K	ADDR[15:11]	G1A[18:11]
0	0	1	0	4 K	ADDR[15:12]	G1A[18:12]
0	0	1	1	8 K	ADDR[15:13]	G1A[18:13]
0	1	0	0	16 K	ADDR[15:14]	G1A[18:14]
0	1	0	1	32 K	ADDR15	G1A[18:15]
0	1	1	0	64 K	None	G1A[18:16]
0	1	1	1	128 K	None	G1A[18:17]
1	0	0	0	256 K	None	G1A18
1	0	0	1	512 K	None	None
1	0	1	0	Follow Window 1	None	None
1	0	1	1	Follow Window 2	None	None
1100–1111				Default to 512 K	None	None

GPCS2A —General-Purpose Chip Select 2 Address **$005E**

	Bit 7	6	5	4	3	2	1	Bit 0
	G2A18	G2A17	G2A16	G2A15	G2A14	G2A13	G2A12	G2A11
RESET:	0	0	0	0	0	0	0	0

G2A[18:11] —General-Purpose Chip Select 2 Address
Selects the Starting Address of General-Purpose Chip Select 2 Range. Refer to G2SZA–G2SZD table.

GPCS2C —General-Purpose Chip Select 2 Control **$005F**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	G2DPC	G2POL	G2AV	G2SZA	G2SZB	G2SZC	G2SZD
RESET:	0	0	0	0	0	0	0	0

Bit 7 — Not implemented
Always reads zero

G2DPC — General-Purpose Chip Select 2 Drives Program Chip Select
0 = CSGP2 does not affect $\overline{\text{CSPROG}}$
1 = CSGP2 and $\overline{\text{CSPROG}}$ are connected to an OR gate and driven out $\overline{\text{CSPROG}}$

G2POL — General-Purpose Chip Select 2 Polarity Select
0 = CSGP2 active low
1 = CSGP2 active high

G2AV — General-Purpose Chip Select 2 Address Valid Select
0 = Active during E high time
1 = Active during address valid time

G2SZA–G2SZD — General-Purpose Chip Select 2 Size

G2SZx				Size (Bytes)	Valid Bits (MXGS2 = 0)	Valid Bits (MXGS2 = 1)
A	B	C	D			
0	0	0	0	Disabled	None	None
0	0	0	1	2 K	ADDR[15:11]	G2A[18:11]
0	0	1	0	4 K	ADDR[15:12]	G2A[18:12]
0	0	1	1	8 K	ADDR[15:13]	G2A[18:13]
0	1	0	0	16 K	ADDR[15:14]	G2A[18:14]
0	1	0	1	32 K	ADDR15	G2A[18:15]
0	1	1	0	64 K	None	G2A[18:16]
0	1	1	1	128 K	None	G2A[18:17]
1	0	0	0	256 K	None	G2A18
1	0	0	1	512 K	None	None
1	0	1	0	Follow Window 1	None	None
1	0	1	1	Follow Window 2	None	None
1100–1111				Default to 512 K	None	None

4.3.6 Examples of Memory Expansion Using Chip Selects

On the following two pages are examples of memory expansion schemes that use chip select signals to simplify the interface to the external memory devices. Although schematics are not provided, careful study of the memory map diagram for each example will reveal the simplicity with which an expanded system can be created. Both examples require a minimum of external circuitry as well as very little program code.

This example is a system consisting of the MCU and a single 27C512-type memory device. This system uses one chip select and has one window containing eight banks of eight Kbytes each. In this example, a total of 64 Kbytes is added to the address range of the MCU. Three of the expansion address lines (XA[15:13]) are used. Register values particular to this example are given below the diagram.

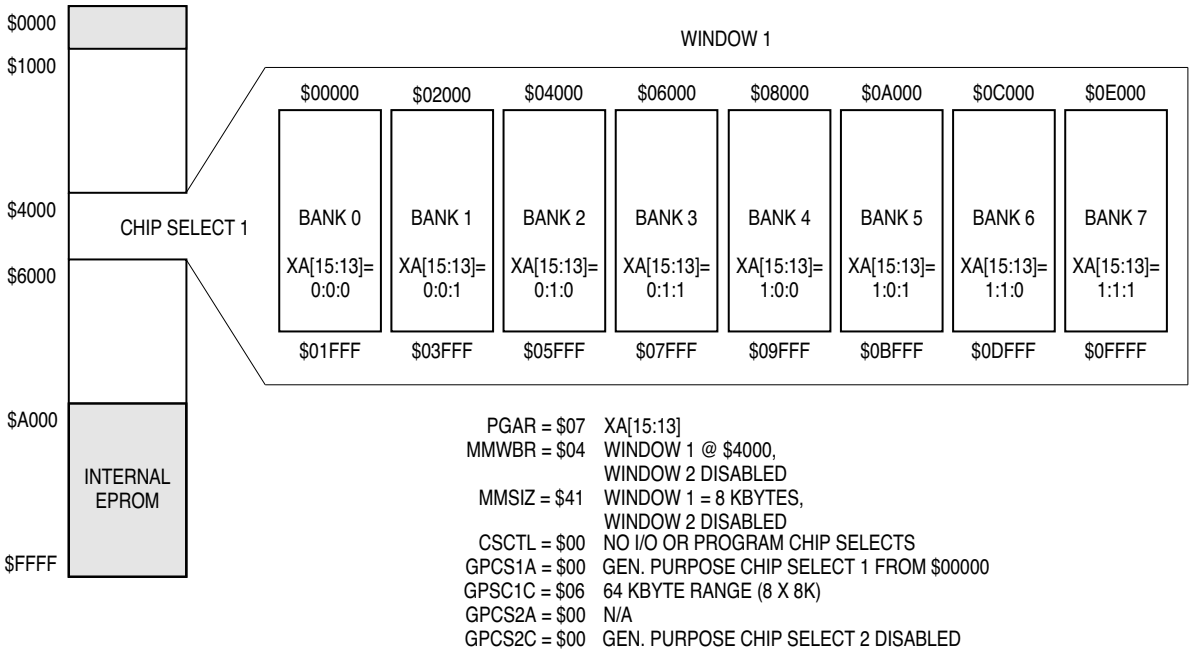
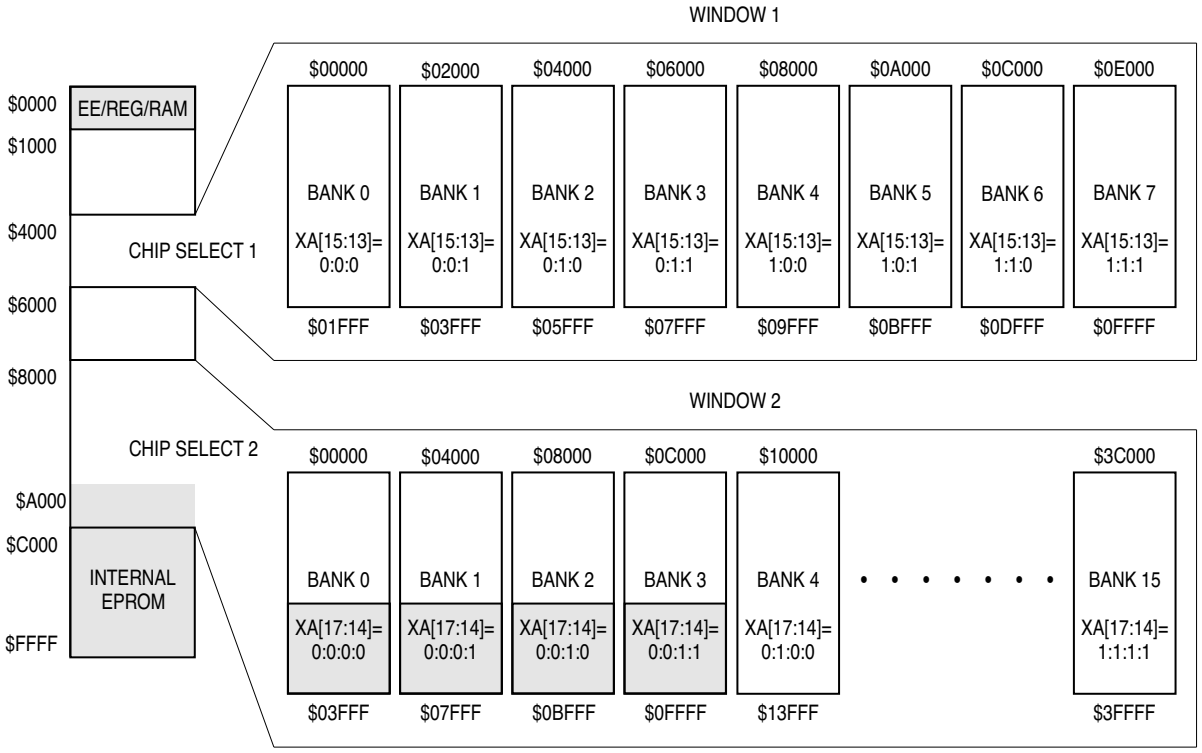


Figure 7 Memory Expansion Example 1

This example is a system consisting of the MCU, a single 27C512-type memory device as in the previous example, and two 6226-type memory devices as well. This system uses two chip selects and has two windows. For purposes of explanation, the setup of the first window is identical to the previous example. In addition, a second window consisting of 16 banks of 16 Kbytes each uses the second chip select signal. Window 1 contains 64 Kbytes of expanded memory pages, window 2 contains a total of 256 Kbytes of expanded memory. A total of five expansion address lines are used. Register values particular to this example are given below the diagram.



PGAR = $1F XA[17:13]
MMWBR = $84 WINDOW 1 @ $4000, WINDOW 2 @ $8000
MMSIZ = $E1 WINDOW 1 = 8 KBYTES, WINDOW 2 = 16 KBYTES

CSCTL = $00 NO I/O OR PROGRAM CHIP SELECTS
GPCS1A = $00 GEN. PURPOSE CHIP SELECT 1 FROM $00000
GPSC1C = $06 64 KBYTE RANGE (8 X 8K)
GPCS2A = $00 GEN. PURPOSE CHIP SELECT 2 FROM $00000
GPCS2C = $08 256 KBYTE RANGE (16 X 16K)

Figure 8 Memory Expansion Example 2

5 Resets and Interrupts

All M68HC11 MCUs have three reset vectors and 18 interrupt vectors. The reset vectors are as follows:

- $\overline{\text{RESET}}$, or Power-On Reset
- Clock Monitor Fail
- COP Failure

The 18 interrupt vectors service 22 interrupt sources (three nonmaskable, 19 maskable). The three nonmaskable interrupt sources are as follows:

- $\overline{\text{XIRQ}}$ Pin (X-Bit Interrupt)
- Illegal Opcode Trap
- Software Interrupt

On-chip peripheral systems generate maskable interrupts, which are recognized only if the global interrupt mask bit (I) in the condition code register (CCR) is clear. Maskable interrupts are prioritized according to a default arrangement; however, any one source can be elevated to the highest maskable priority position by a software-accessible control register (HPRIO). The HPRIO register can be written at any time, provided bit I in the CCR is set.

Nineteen interrupt sources in the M68HC11 K series devices are subject to masking by the global interrupt mask bit (bit I in the CCR). In addition to the global bit I, all of these sources, except the external interrupt ($\overline{\text{IRQ}}$) pin, are controlled by local enable bits in control registers. Most interrupt sources in M68HC11 devices have separate interrupt vectors; therefore, there is usually no need for software to poll control registers to determine the cause of an interrupt.

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 invoked by 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.

Refer to the following table for a list of interrupt and reset vector assignments.

Vector Address	Interrupt Source	CCR Mask Bit	Local Mask	Priority (1 = High)
FFC0, C1 —FFD4, D5	Reserved	—	—	—
FFD6, D7	SCI Serial System	I		
	• SCI Receive Data Register Full		RIE	19
	• SCI Receiver Overrun		RIE	20
	• SCI Transmit Data Register Empty		TIE	21
	• SCI Transmit Complete		TCIE	22
	• SCI Idle Line Detect		ILIE	23
FFD8, D9	SPI Serial Transfer Complete	I	SPIE	18
FFDA, DB	Pulse Accumulator Input Edge	I	PAII	17
FFDC, DD	Pulse Accumulator Overflow	I	PAOVI	16
FFDE, DF	Timer Overflow	I	TOI	15
FFE0, E1	Timer Input Capture 4/Output Compare 5	I	I4/O5I	14
FFE2, E3	Timer Output Compare 4	I	OC4I	13
FFE4, E5	Timer Output Compare 3	I	OC3I	12
FFE6, E7	Timer Output Compare 2	I	OC2I	11
FFE8, E9	Timer Output Compare 1	I	OC1I	10
FFEA, EB	Timer Input Capture 3	I	IC3I	9
FFEC, ED	Timer Input Capture 2	I	IC2I	8
FFEE, EF	Timer Input Capture 1	I	IC1I	7
FFF0, F1	Real Time Interrupt	I	RTII	6
FFF2, F3	$\overline{\text{IRQ}}$	I	None	5
FFF4, F5	$\overline{\text{XIRQ}}$ Pin	X	None	4
FFF6, F7	Software Interrupt	None	None	*
FFF8, F9	Illegal Opcode Trap	None	None	*
FFFA, FB	COP Failure	None	NOCOP	3
FFFC, FD	Clock Monitor Fail	None	CME	2
FFFE, FF	RESET	None	None	1

*Same level as an instruction

OPTION —System Configuration Options **$0039**

	Bit 7	6	5	4	3	2	1	Bit 0
	ADPU	CSEL	IRQE*	DLY*	CME	FCME*	CR1*	CR0*
RESET:	0	0	0	1	0	0	0	0

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

ADPU —A/D Converter Power up
Refer to **9 Analog-to-Digital Converter**.

CSEL —Clock Select
Refer to **9 Analog-to-Digital Converter**.

IRQE —$\overline{\text{IRQ}}$ Select Edge Sensitive Only
0 = Low level recognition
1 = Falling edge recognition

DLY —Enable Oscillator Start-Up Delay on Exit from STOP
0 = No stabilization delay on exit from STOP
1 = Stabilization delay enabled on exit from STOP

CME —Clock Monitor Enable
0 = Clock monitor disabled; slow clocks can be used
1 = Slow or stopped clocks cause clock failure reset

FCME —Force Clock Monitor Enable
0 = Clock monitor follows the state of the CME bit
1 = Clock monitor circuit is enabled until next reset

CR[1:0] —COP Timer Rate Select
Refer to NOCOP bit in CONFIG register.

Table 6 COP Timer Rate Select (Timeout Period Length)

CR[1:0]	Rate Selected	XTAL = 8.0 MHz Timeout –0 ms, +16.4 ms	XTAL = 12.0 MHz Timeout –0 ms, +10.9 ms	XTAL = 16.0 MHz Timeout –0 ms, +8.2 ms
0 0	2^{15}	16.384 ms	10.923 ms	8.192 ms
0 1	2^{17}	65.536 ms	43.691 ms	32.768 ms
1 0	2^{19}	262.14 ms	174.76 ms	131.07 ms
1 1	2^{21}	1.049 sec	699.05 ms	524.29 ms
	E =	2.0 MHz	3.0 MHz	4.0 MHz

COPRST —Arm/Reset COP Timer Circuitry **$003A**

	Bit 7	6	5	4	3	2	1	Bit 0
	7	6	5	4	3	2	1	0
RESET:	0	0	0	0	0	0	0	0

Write $55 (%01010101) to COPRST to arm COP watchdog clearing mechanism. Write $AA (%10101010) to COPRST to reset COP watchdog. Refer to NOCOP bit in CONFIG register.

HPRIO —Highest Priority I-Bit Interrupt and Miscellaneous **$003C**

	Bit 7	6	5	4	3	2	1	Bit 0
	RBOOT*	SMOD*	MDA*	PSEL4	PSEL3	PSEL2	PSEL1	PSEL0
RESET:	—	—	—	0	0	1	1	0

*RBOOT, SMOD, and MDA reset depend on power-up initialization mode and can only be written in special mode.

RBOOT —Read Bootstrap ROM
Refer to **2 Operating Modes**.

SMOD —Special Mode Select
Refer to **2 Operating Modes**.

MDA —Mode Select A
Refer to **2 Operating Modes**.

PSEL[4:0] —Priority Select Bit 4 through Bit 0

Can be written only while the I-bit in the CCR is set (interrupts disabled). These bits select one interrupt source to be elevated above all other I-bit related sources.

PSELx					Interrupt Source Promoted
4	3	2	1	0	
0	0	0	X	X	Reserved (Default to $\overline{IRQ}$)
0	0	1	0	0	Reserved (Default to $\overline{IRQ}$)
0	0	1	0	1	Reserved (Default to $\overline{IRQ}$)
0	0	1	1	0	$\overline{IRQ}$
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	SPI Serial Transfer Complete
1	0	1	0	0	SCI Serial System
1	0	1	0	1	Reserved (Default to $\overline{IRQ}$)
1	0	1	1	0	Reserved (Default to $\overline{IRQ}$)
1	0	1	1	1	Reserved (Default to $\overline{IRQ}$)
1	1	X	X	X	Reserved (Default to $\overline{IRQ}$)

6 Parallel Input/Output

M68HC11 K-series MCUs have up to 62 input/output lines, depending on the operating mode. To enhance the I/O functions, the data bus of this microcontroller is nonmultiplexed. The following table is a summary of the configuration and features of each port.

Port	Input Pins	Output Pins	Bidirectional Pins	Shared Functions
Port A	—	—	8	Timer
Port B	—	—	8	High Order Address
Port C	—	—	8	Data Bus
Port D	—	—	6	SCI and SPI
Port E	8	—	—	A/D Converter
Port F	—	—	8	Low Order Address
Port G	—	—	8	Memory Expansion
Port H	—	—	8	PWM, Chip Select

NOTE

Port pin function is mode dependent. Do not confuse pin function with the electrical state of the pin at reset. Port pins are either driven to a specified logic level or are configured as high impedance inputs. I/O pins configured as high-impedance inputs have port data that is indeterminate. The contents of the corresponding latches are dependent upon the electrical state of the pins during reset. In port descriptions, an "I" indicates this condition. Port pins that are driven to a known logic level during reset are shown with a value of either one or zero. Some control bits are unaffected by reset. Reset states for these bits are indicated with a "U".

PORTA —Port A Data **$0000**

	Bit 7	6	5	4	3	2	1	Bit 0
	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
RESET:	I	I	I	I	I	I	I	I
Alt. Pin Func.:	PAI	OC2	OC3	OC4	IC4/OC5	IC1	IC2	IC3
And/or:	OC1	OC1	OC1	OC1	OC1	—	—	—

NOTE

To enable PA3 as 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, 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. PA7 drives the pulse accumulator input but also can be configured for general-purpose I/O or output compare. Note that even when PA7 is configured as an output, the pin still drives the pulse accumulator input.

DDRA —Data Direction Register for Port A **$0001**

	Bit 7	6	5	4	3	2	1	Bit 0
	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0
RESET:	0	0	0	0	0	0	0	0

DDA[7:0] —Data Direction for Port A

0 = Corresponding pin configured for input

1 = Corresponding pin configured for output

PORTB —Port B Data **$0004**

	Bit 7	6	5	4	3	2	1	Bit 0
	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
S. Chip or Boot:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
RESET:	I	I	I	I	I	I	I	I
Expan. or Test:	ADDR15	ADDR14	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8

Reset state is mode dependent. In single-chip or bootstrap modes, port B pins are high-impedance inputs with selectable internal pull-up resistors. In expanded or test modes, port B pins are high order address outputs and PORTB is not in the memory map.

DDRB —Data Direction Register for Port B **$0002**

	Bit 7	6	5	4	3	2	1	Bit 0
	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
RESET:	0	0	0	0	0	0	0	0

DDB[7:0] —Data Direction for Port B
0 = Corresponding pin configured for input
1 = Corresponding pin configured for output

PORTC —Port C Data **$0006**

	Bit 7	6	5	4	3	2	1	Bit 0
	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
S. Chip or Boot:	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
RESET:	0	0	0	0	0	0	0	0
Expan. or Test:	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

Reset state is mode dependent. In single-chip or bootstrap modes, port C pins are high-impedance inputs with selectable internal pull-up resistors. In expanded or test modes, port C pins are data bus inputs and outputs and PORTC is not in the memory map. Refer to CWOM bit in OPT2 register description that follows.

DDRC —Data Direction Register for Port C **$0007**

	Bit 7	6	5	4	3	2	1	Bit 0
	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
RESET:	0	0	0	0	0	0	0	0

DDC[7:0] —Data Direction for Port C. Refer to CWOM bit in OPT2 register description that follows.
0 = Corresponding pin configured for input
1 = Corresponding pin configured for output

OPT2 —System Configuration Options 2 **$0038**

	Bit 7	6	5	4	3	2	1	Bit 0
	LIRDV	CWOM	—	IRVNE	LSBF	SPR2	XDV1	XDV0
RESET:	0	0	0	—	0	0	0	0

LIRDV—LIR Driven
Refer to **2 Operating Modes**.

CWOM —Port C Wired-OR Mode
0 = Port C operates normally.
1 = Port C outputs are open-drain.

Bit 5 —Not implemented
Always read zero

IRVNE —Internal Read Visibility/Not E
Refer to **2 Operating Modes**.

LSBF —SPI LSB First Enable
Refer to **8 Serial Peripheral Interface**.

SPR2 —SPI Clock (SCK) Rate Select
Refer to **8 Serial Peripheral Interface**.

XDV[1:0] —XOUT Clock Divide Select
Refer to **2 Operating Modes**.

PORTD —Port D Data **$0008**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	PD5	PD4	PD3	PD2	PD1	PD0
RESET:	0	0	I	I	I	I	I	I
Alt. Pin Func.:	—	—	$\overline{SS}$	SCK	MOSI	MISO	TxD	RxD

DDRD —Data Direction Register for Port D **$0009**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
RESET:	0	0	0	0	0	0	0	0

Bits [7:6] — Not implemented
Always read zero

DDD[5:0] — Data Direction for Port D
0 = Corresponding pin configured for input
1 = Corresponding pin configured for output

NOTE

When the SPI system is in slave mode, DDD5 has no meaning nor effect. When the SPI system is in master mode, DDD5 determines whether bit 5 of PORTD is an error detect input (DDD5 = 0) or a general-purpose output (DDD5 = 1). If the SPI system is enabled and expects any of bits [4:2] to be an input that bit will be an input regardless of the state of the associated DDR bit. If any of bits [4:2] are expected to be outputs that bit will be an output **only** if the associated DDR bit is set.

SPCR —Serial Peripheral Control **$0028**

	Bit 7	6	5	4	3	2	1	Bit 0	
	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	
RESET:	0	0	1	0	0	0	0	0	Boot Mode
	0	0	0	0	0	0	0	0	Other Modes

SPIE —SPI Interrupt Enable
Refer to **8 Serial Peripheral Interface**.

SPE —SPI System Enable
Refer to **8 Serial Peripheral Interface**.

DWOM —Port D Wired-OR Mode Option for SPI Pins PD[5:2] (See also WOMS bit in SCCR1)
0 = PD[5:2] are normal CMOS outputs
1 = PD[5:2] are open-drain outputs

MSTR —Master/Slave Mode Select
Refer to **8 Serial Peripheral Interface**.

CPOL —Clock Polarity
Refer to **8 Serial Peripheral Interface**.

CPHA —Clock Phase
Refer to **8 Serial Peripheral Interface**.

SPR[1:0] —SPI Clock Rate Selects
Refer to **8 Serial Peripheral Interface**.

SCCR1 —SCI Control 1 **$0072**

	Bit 7	6	5	4	3	2	1	Bit 0	
	LOOPS	WOMS	—	M	WAKE	ILT	PE	PT	
RESET:	0	1	0	0	0	0	0	0	Boot Mode
	0	0	0	0	0	0	0	0	Other Modes

LOOPS —SCI LOOP Mode Enable
Refer to **7 Serial Communications Interface**.

WOMS —Port D Wired-OR Mode Option for SPI Pins PD[5:2] (See also DWOM bit in SPCR.)
0 = TxD and RxD operate normally
1 = TxD and RxD are open drains if operating as an output

Bit 5 —Not implemented
Always reads zero

M —Mode (Select Character Format)
Refer to **7 Serial Communications Interface**.

WAKE —Wakeup by Address Mark/Idle
Refer to **7 Serial Communications Interface**.

ILT —Idle Line Type
Refer to **7 Serial Communications Interface**.

PE —Parity Enable
Refer to **7 Serial Communications Interface**.

PT —Parity Type

Refer to **7 Serial Communications Interface**.

PORTE —Port E Data **$000A**

	Bit 7	6	5	4	3	2	1	Bit 0
	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
RESET:	I	I	I	I	I	I	I	I
Alt. Pin Func.:	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0

DDRF —Data Direction Register for Port F **$0003**

	Bit 7	6	5	4	3	2	1	Bit 0
	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0
RESET:	0	0	0	0	0	0	0	0

DDF[7:0] —Data Direction for Port F

0 = Corresponding pin configured for input

1 = Corresponding pin configured for output

PORTF —Port F Data **$0005**

	Bit 7	6	5	4	3	2	1	Bit 0
	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
S. Chip or Boot:	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
RESET:	I	I	I	I	I	I	I	I
Expan. or Test:	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0

Reset state is mode dependent. In single-chip or bootstrap modes, port F is high-impedance input with selectable internal pull-up resistors. In expanded or test modes, port F pins are low order address outputs and PORTF is not in the memory map.

PORTH —Port H Data **$007C**

	Bit 7	6	5	4	3	2	1	Bit 0
	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
RESET:	I	I	I	I	I	I	I	I
Alt. Pin Func.:	$\overline{\text{CSPROG}}$	CSGP2	CSGP1	CSIO	PW4	PW3	PW2	PW1

Port H pins reset to high-impedance inputs with selectable internal pull-up resistors. In expanded and special test modes, reset also causes PH7 to be configured as $\overline{\text{CSPROG}}$.

DDRH —Data Direction Register for Port H **$007D**

	Bit 7	6	5	4	3	2	1	Bit 0
	DDH7	DDH6	DDH5	DDH4	DDH3	DDH2	DDH1	DDH0
RESET:	0	0	0	0	0	0	0	0

DDH[7:0] —Data Direction for Port H

0 = Bits set to zero to configure corresponding I/O pin for input only

1 = Bits set to one to configure corresponding I/O pin for output

NOTE

In expanded and special test modes, chip-select circuitry forces the I/O state to be an output for each port H pin associated with an enabled chip select. In any mode, PWM circuitry forces the I/O state to be an output for each port H line associated with an enabled pulse width modulator channel. In these cases, data direction bits are not changed and have no effect on these lines. DDRH reverts to controlling the I/O state of a pin when the associated function is disabled. Refer to **4.3 Memory Expansion and Chip Selects** and **12 Pulse-Width Modulation Timer** for further information.

PORTG —Port G Data **$007E**

	Bit 7	6	5	4	3	2	1	Bit 0
	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
RESET:	I	I	I	I	I	I	I	I
Alt. Pin Func.:	R/$\overline{W}$	—	XA18	XA17	XA16	XA15	XA14	XA13

Port G pins reset to high-impedance inputs with selectable internal pull-up resistors. In expanded and special test modes PG7 becomes R/$\overline{W}$. Refer to PGAR register description.

DDRG —Data Direction Register for Port G **$007F**

	Bit 7	6	5	4	3	2	1	Bit 0
	DDG7	DDG6	DDG5	DDG4	DDG3	DDG2	DDG1	DDG0
RESET:	0	0	0	0	0	0	0	0

DDG[7:0] —Data Direction for Port G

0 = Configure corresponding I/O pin for input only

1 = Configure corresponding I/O pin for output

In expanded and test modes, bit 7 is configured for R/$\overline{W}$, forcing the state of this pin to be an output although the DDRG value remains zero. Refer to PGAR register description.

PGAR — Port G Assignment **$002D**

	Bit 7	6	5	4	3	2	1	Bit 0
$**0**02D	—	—	PGAR5	PGAR4	PGAR3	PGAR2	PGAR1	PGAR0
RESET:	0	0	0	0	0	0	0	0

Bits [7:6] —Not implemented

Always read zero

PGAR[5:0] —Port G Pin Assignment Bits [5:0]

0 = Corresponding port G pin is general-purpose I/O

1 = Corresponding port G pin is memory expansion address line (XA[18:13])

NOTE

Each PGAR bit forces the I/O state to be an output for each port G pin associated with an enabled expansion address line. In this case, data direction bits are not changed and have no effect on these lines. DDRG reverts to controlling the I/O state of a pin when the associated function is disabled. Refer to **4.1 Memory Expansion** for further information.

PPAR —Port Pull-Up Assignment **$002C**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	—	—	HPPUE	GPPUE	FPPUE	BPPUE
RESET:	0	0	0	0	1	1	1	1

Bits [7:4] —Not implemented
Always read zero

xPPUE —Port x Pin Pull-Up Enable
Valid only when PAREN = 1. Refer to PAREN bit in the CONFIG register description.
0 = Port x pin on-chip pull-up devices disabled
1 = Port x pin on-chip pull-up devices enabled

NOTE

FPPUE and BPPUE have no effect in expanded mode because port F and port B are address outputs.

7 Serial Communications Interface

The SCI, a universal asynchronous receiver transmitter (UART) serial communications interface, is one of two independent serial I/O subsystems in M68HC11 K-series MCUs. Rearranging registers and control bits used in previous M68HC11 family devices has enhanced the existing SCI system and added new features, which include the following:

- A 13-bit modulus prescaler that allows greater baud rate control
- A new idle mode detect, independent of preceding serial data
- A receiver active flag
- Hardware parity for both transmitter and receiver

The enhanced baud rate generator is shown in the following diagram. Refer to **Table 7** for standard values.

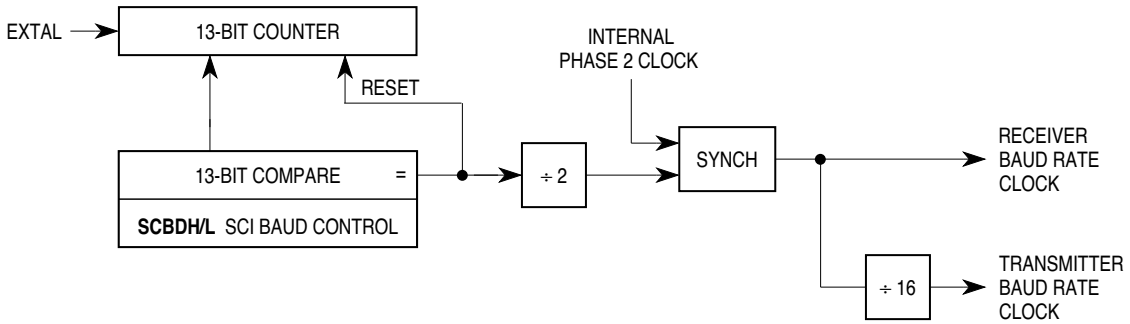


Figure 9 SCI Baud Generator Circuit Diagram

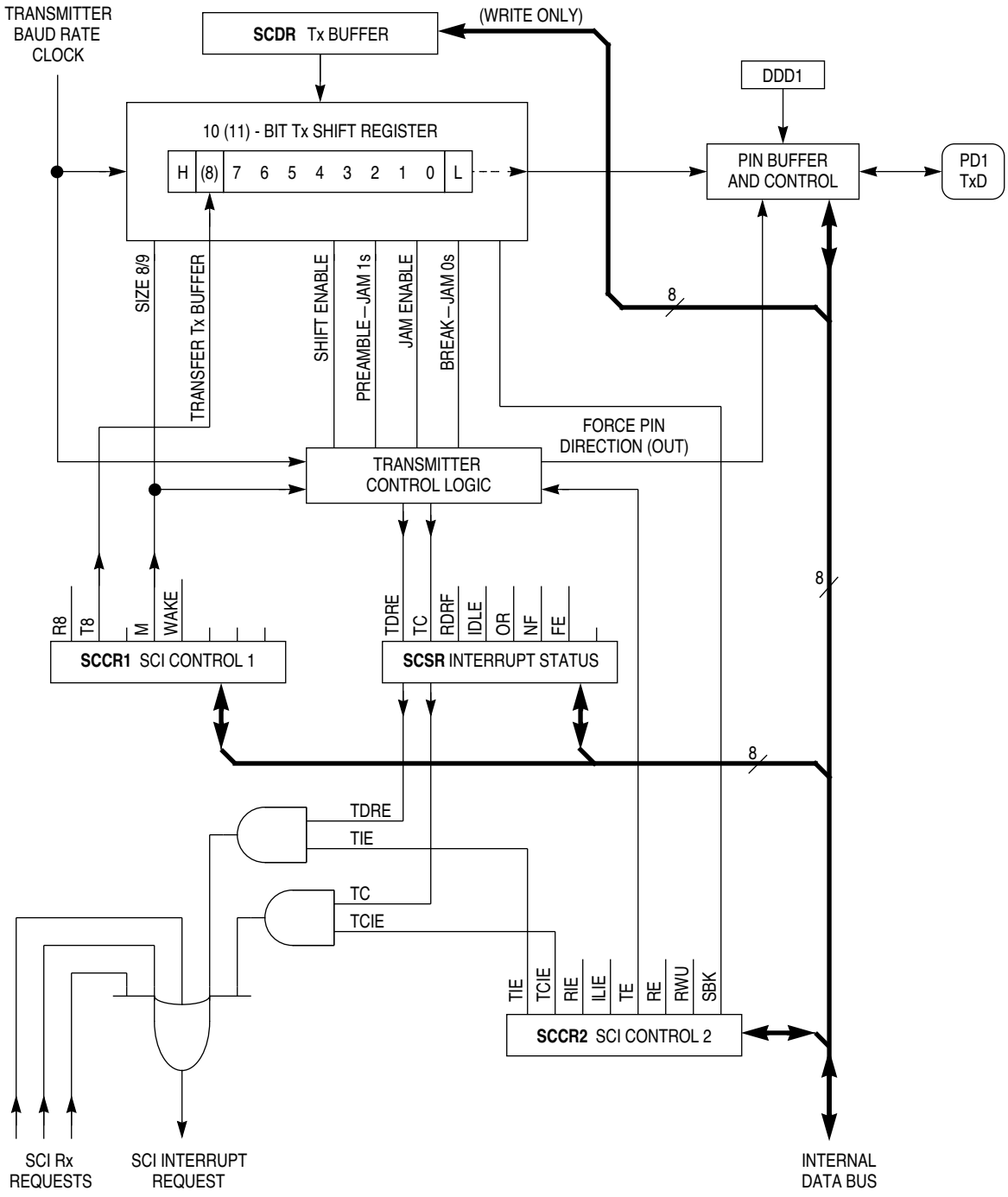


Figure 10 SCI Transmitter Block Diagram

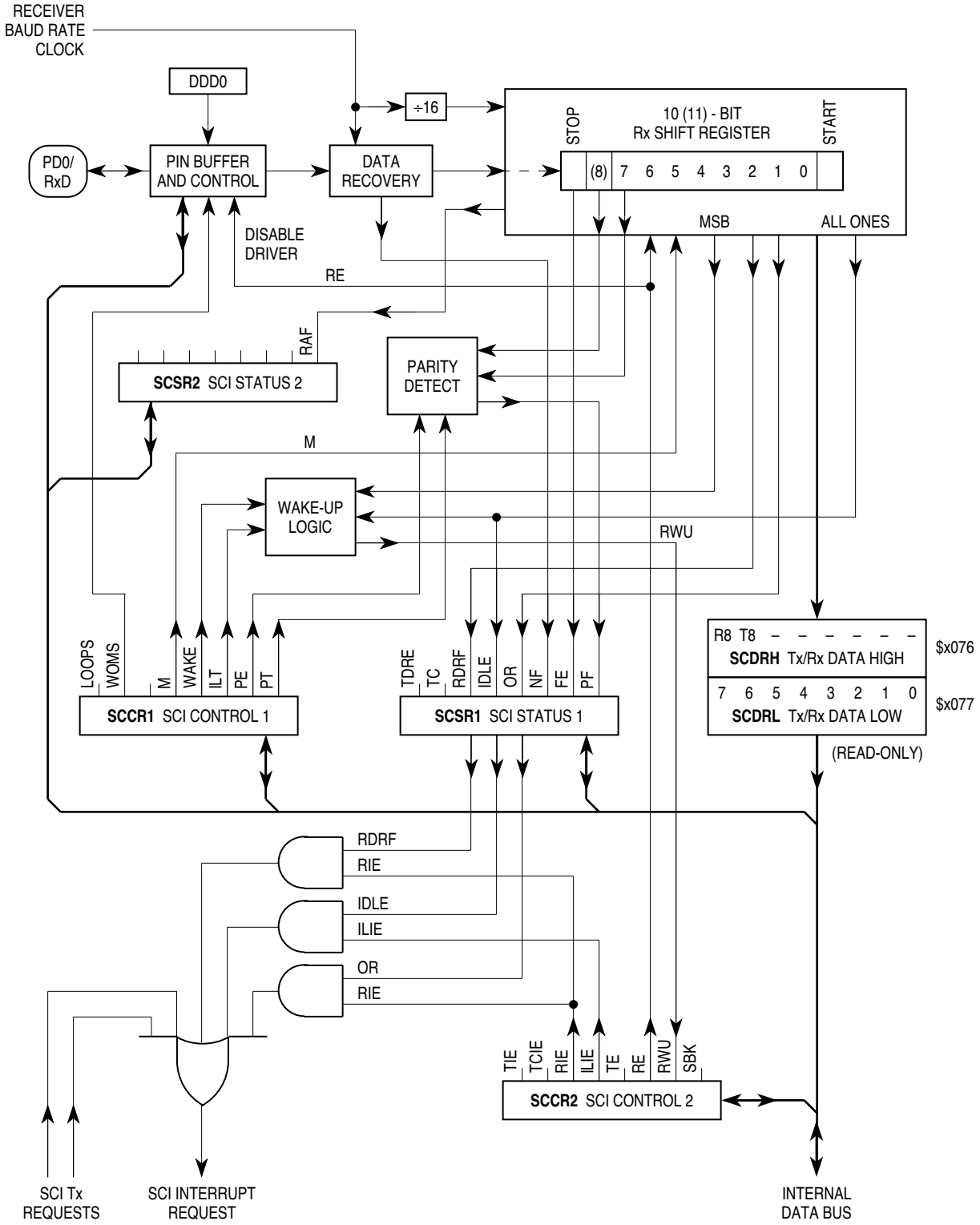


Figure 11 SCI Receiver Block Diagram

SCBDH/L —SCI Baud Rate Control High/Low **$0070, $0071**

	Bit 7	6	5	4	3	2	1	Bit 0	
$0070	BTST	BSPL	—	SBR12	SBR11	SBR10	SBR9	SBR8	High
RESET:	0	0	0	0	0	0	0	0	
$0071	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0	Low
RESET:	0	0	0	0	0	1	0	0	

BTST —Baud Register Test (TEST)
Factory test only

BSPL —Baud Rate Counter Split (TEST)
Factory test only

Bit 5 —Not implemented
Always reads zero

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.

SCI baud rate = EXTAL ÷[16 ∗ (2 ∗ BR)]

Where BR is the contents of SCBDH/L (BR = 1, 2, 3 ... 8191).

BR = 0 disables the baud rate generator.

Table 7 SCI Baud Rate Control Values

Target Baud Rate	Crystal Frequency (EXTAL)					
	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
19.2 K	13	$000D	20	$0014	26	$001A
38.4 K	—	—	—	—	13	$000D

SCCR1 —SCI Control 1 **$0072**

	Bit 7	6	5	4	3	2	1	Bit 0	
	LOOPS	WOMS	—	M	WAKE	ILT	PE	PT	
RESET:	0	1	0	0	0	0	0	0	Bootstrap Mode
	0	0	0	0	0	0	0	0	Other Modes

LOOPS —SCI LOOP Mode Enable
0 = SCI transmit and receive operate normally
1 = SCI transmit and receive are disconnected from TxD and RxD pins, and transmitter output is fed back into the receiver input

WOMS —Wired-OR Mode Option for PD[1:0] (See also DWOM bit in SPCR.)
0 = TxD and RxD operate normally
1 = TxD and RxD are open drains if operating as an output

Bit 5 —Not implemented
Always reads zero

M —Mode (Select Character Format)
0 = Start bit, 8 data bits, 1 stop bit
1 = Start bit, 9 data bits, 1 stop bit

WAKE —Wakeup by Address Mark/Idle
0 = Wakeup by IDLE line recognition
1 = Wakeup by address mark (most significant data bit set)

ILT —Idle Line Type
0 = Short (SCI counts consecutive ones after start bit)
1 = Long (SCI counts ones only after stop bit)

PE —Parity Enable
0 = Parity disabled
1 = Parity enabled

PT —Parity Type
0 = Parity even (even number of ones causes parity bit to be zero, odd number of ones causes parity bit to be one)
1 = Parity odd (odd number of ones causes parity bit to be zero, even number of ones causes parity bit to be one)

SCCR2 —SCI Control 2 **$0073**

	Bit 7	6	5	4	3	2	1	Bit 0
	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET:	0	0	0	0	0	0	0	0

TIE —Transmit Interrupt Enable
0 = TDRE interrupts disabled
1 = SCI interrupt requested when TDRE status flag is set

TCIE —Transmit Complete Interrupt Enable
0 = TC interrupts disabled
1 = SCI interrupt requested when TC status flag is set

RIE —Receiver Interrupt Enable
0 = RDRF and OR interrupts disabled
1 = SCI interrupt requested when RDRF flag or the OR status flag is set

ILIE —Idle Line Interrupt Enable
0 = IDLE interrupts disabled
1 = SCI interrupt requested when IDLE status flag is set

TE —Transmitter Enable
0 = Transmitter disabled
1 = Transmitter enabled

RE —Receiver Enable

0 = Receiver disabled
1 = Receiver enabled

RWU —Receiver Wakeup Control

0 = Normal SCI receiver
1 = Wakeup enabled and receiver interrupts inhibited

SBK —Send Break

0 = Break generator off
1 = Break codes generated as long as SBK = 1

SCSR1 —SCI Status Register 1 **$0074**

	Bit 7	6	5	4	3	2	1	Bit 0
	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
RESET:	1	1	0	0	0	0	0	0

TDRE —Transmit Data Register Empty Flag

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

0 = SCDR busy
1 = SCDR empty

TC —Transmit Complete Flag

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

0 = Transmitter busy
1 = Transmitter idle

RDRF —Receive Data Register Full Flag

RDRF is set if a received character is ready to be read from SCDR. Clear the RDRF flag by reading SCSR1 and then reading SCDR.

0 = SCDR empty
1 = SCDR full

IDLE —Idle Line Detected Flag

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 and then reading SCDR.

0 = RxD line is active
1 = RxD line is idle

OR —Overrun Error Flag

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 and then reading SCDR.

0 = No overrun
1 = Overrun detected

NF —Noise Error Flag

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

0 = Unanimous decision
1 = Noise detected

FE —Framing Error

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

0 = Stop bit detected
1 = Zero detected

PF —Parity Error Flag

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

0 = Parity correct

1 = Incorrect parity detected

SCSR2 —SCI Status Register 2 **$0075**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	—	—	—	—	—	RAF
RESET:	0	0	0	0	0	0	0	0

Bits [7:1] —Not implemented

Always read zero

RAF —Receiver Active Flag (Read Only)

0 = A character is not being received

1 = A character is being received

SCDRH, SCDRL —SCI Data Register High/Low **$0076, $0077**

	Bit 7	6	5	4	3	2	1	Bit 0	
$0076	R8	T8	—	—	—	—	—	—	SCDRH (High)
$0077	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0	SCDRL (Low)

R8 —Receiver Bit 8

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

T8 —Transmitter Bit 8

Ninth serial data bit transmitted when SCI is configured for 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.

8 Serial Peripheral Interface

The SPI allows the MCU to communicate synchronously with peripheral devices and other microprocessors. Data rates can be as high as 2 Mbits per second when configured as a master and 4 Mbits per second when configured as a slave (assuming 4 MHz bus speed).

Two control bits in OPT2 allow the transfer of data either MSB or LSB first and select an additional divide by four stage to be inserted before the SPI baud rate clock divider.

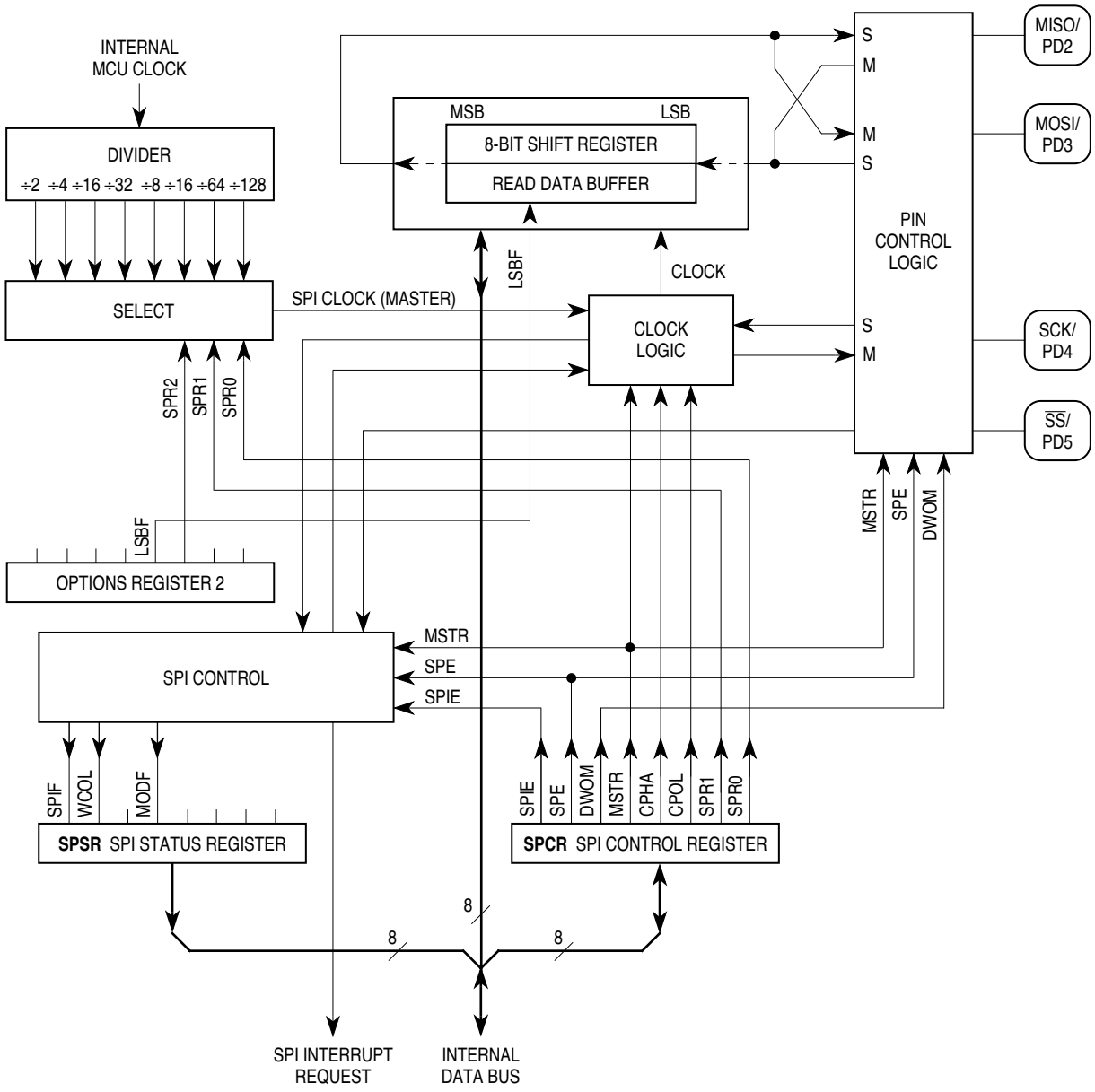


Figure 12 SPI Block Diagram

SPCR —Serial Peripheral Control Register **$0028**

	Bit 7	6	5	4	3	2	1	Bit 0
	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0
RESET:	0	0	0	0	0	1	U	U

SPIE —Serial Peripheral Interrupt Enable
0 = SPI interrupts disabled
1 = SPI interrupts enabled

SPE —Serial Peripheral System Enable
0 = SPI off
1 = SPI on

DWOM —Port D Wired-OR Mode Option for SPI Pins PD[5:2] (See also WOMS bit in SCCR1.)
0 = Normal CMOS outputs
1 = Open-drain outputs

MSTR —Master Mode Select
0 = Slave mode
1 = Master mode

CPOL, CPHA —Clock Polarity, Clock Phase
Refer to the following figure, **SPI Transfer Format**.

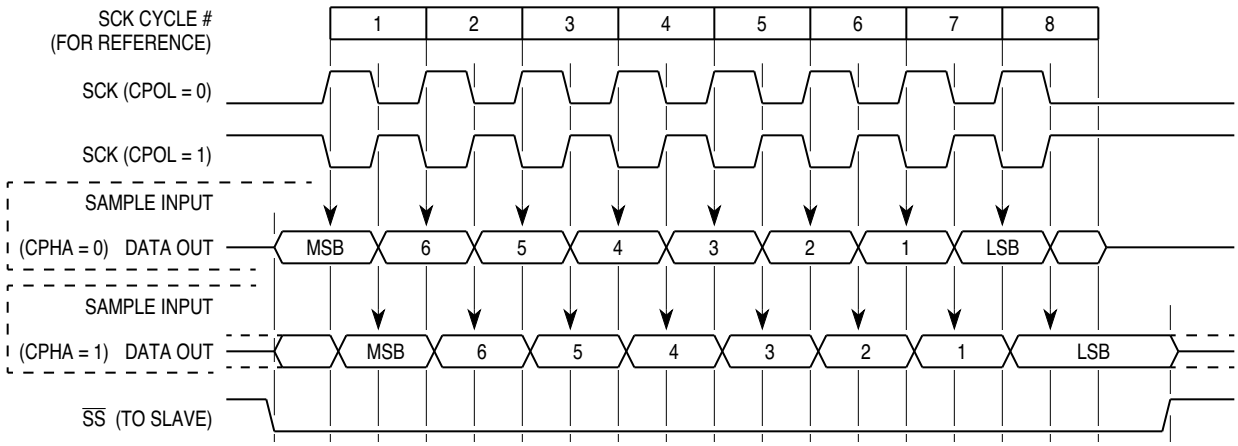


Figure 13 SPI Transfer Format

NOTE

This figure shows transmission order when LSBF = 0 default. If LSBF = 1, data is transferred in reverse order (LSB first).

SPR[2:0] —SPI Clock Rate Selects (SPR2 is located in OPT2 register)

Table 8 SPI Clock Rate Selects

SPR[2:0]	Divide E Clock By	Frequency at E = 2 MHz (Baud)	Frequency at E = 3 MHz (Baud)	Frequency at E = 4 MHz (Baud)
0 0 0	2	1.0 MHz	3.0 MHz	4.0 MHz
0 0 1	4	500 kHz	750 kHz	1.0 MHz
0 1 0	16	125 kHz	187.5 kHz	250 kHz
0 1 1	32	62.5 kHz	93.75 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	31.25 kHz	46.875 kHz	62.5 kHz
1 1 1	128	15.625 kHz	23.438 kHz	31.25 kHz

SPSR —Serial Peripheral Status Register **$0029**

	Bit 7	6	5	4	3	2	1	Bit 0
	SPIF	WCOL	—	MODF	—	—	—	—
RESET:	0	0	0	0	0	0	0	0

SPIF —SPI Transfer Complete Flag

This flag is set when an SPI transfer is complete (after eight SCK cycles in a data transfer). Clear this flag by reading SPSR, then access SPDR.

0 = No SPI transfer complete or SPI transfer still in progress

1 = SPI transfer complete

WCOL —Write Collision Error Flag

This flag is set if the MCU tries to write data into SPDR while an SPI data transfer is in progress. Clear this flag by reading SPSR, then access SPDR.

0 = No write collision error

1 = SPDR written while SPI transfer in progress

Bit 5 —Not implemented

Always reads zero

MODF —Mode Fault (Mode fault terminates SPI operation)

Set when $\overline{SS}$ is pulled low while MSTR = 1. Cleared by SPSR read followed by SPCR write.

0 = No mode fault error

1 = $\overline{SS}$ pulled low in master mode

Bits [3:0] —Not implemented

Always read zero

SPDR —SPI Data **$002A**

Bit 7	6	5	4	3	2	1	Bit 0
Bit 7	6	5	4	3	2	1	Bit 0

SPI is double buffered in, single buffered out.

OPT2 —System Configuration Options 2 **$0038**

	Bit 7	6	5	4	3	2	1	Bit 0
	LIRDV	CWOM	—	IRVNE	LSBF	SPR2	XDV1	XDV0
RESET:	0	0	0	—	0	0	0	0

LIRDV—LIR Driven
Refer to **2 Operating Modes**.

CWOM —Port C Wired-OR Mode
Refer to **6 Parallel Input/Output**.

Bit 5 —Not implemented
Always read zero

IRVNE —Internal Read Visibility/Not E
Refer to **2 Operating Modes**.

LSBF —SPI LSB First Enable
0 = SPI data transferred MSB first
1 = SPI data transferred LSB first

SPR2 —SPI Clock (SCK) Rate Select
Adds a divide by four prescaler to SPI clock chain. Refer to SPCR register.

XDV[1:0] —XOUT Clock Divide Select
Refer to **2 Operating Modes**.

9 Analog-to-Digital Converter

The analog-to-digital (A/D) converter system uses an all-capacitive charge-redistribution technique to convert analog signals to digital values. The A/D converter system contained in M68HC11 K-series MCUs is an 8-channel,8-bit, multiplexed-input, successive-approximation converter. It does not require external sample and hold circuits.

The clock source for the A/D converter's charge pump, like the clock source for the EEPROM charge pump, is selected with the CSEL bit in the OPTION register. When the E clock is slower than 1 MHz, the CSEL bit must be set to ensure that the successive approximation sequence for the A/D converter will be completed before any charge loss occurs. In the case of the EEPROM, it is the efficiency of the charge pump that is affected.

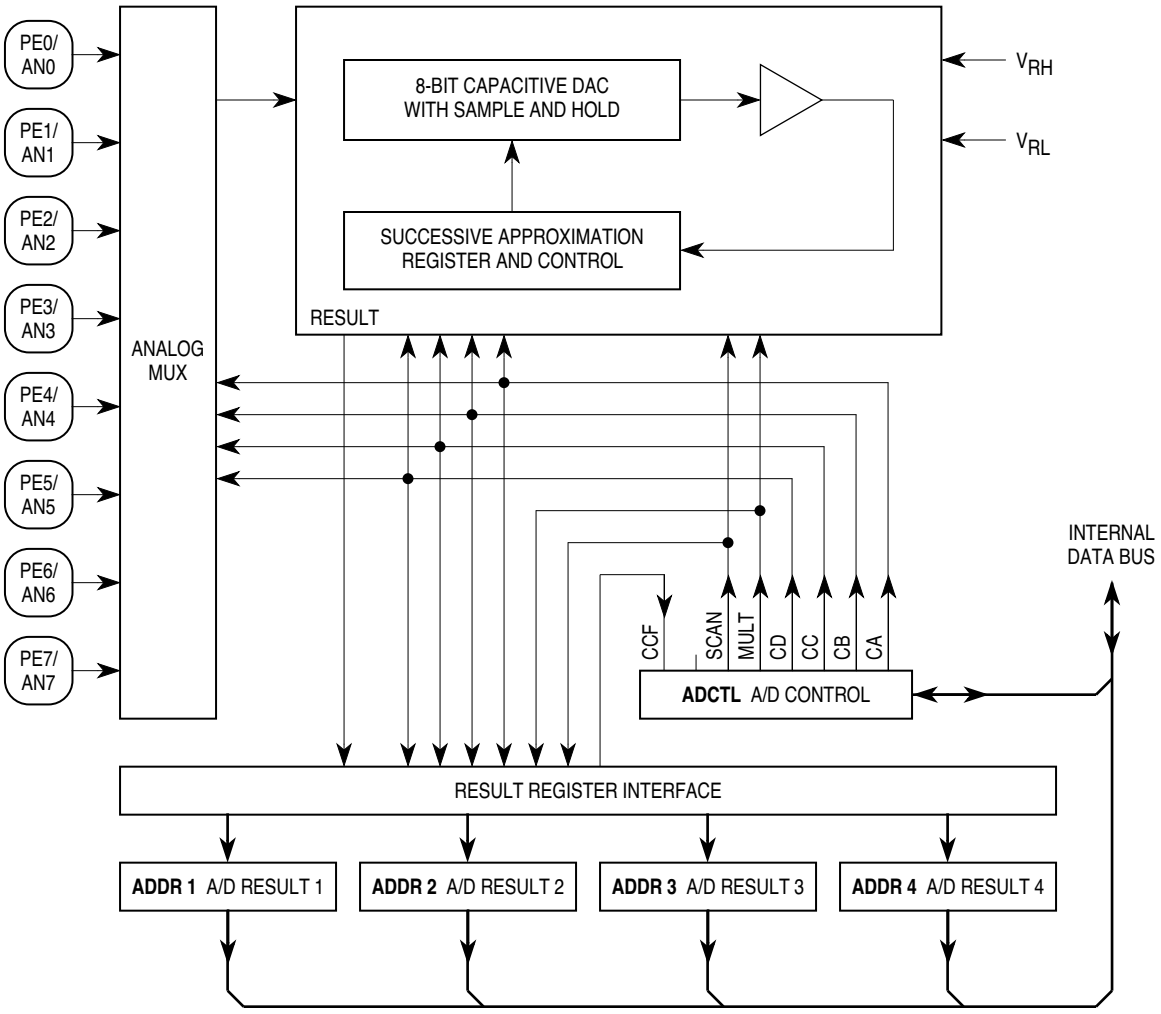


Figure 14 A/D Converter Block Diagram

The A/D converter can operate in single or multiple conversion modes. Multiple conversions are performed in sequences of four. Sequences can be performed on a single channel or an a group of channels.

Dedicated lines V_{RH} and V_{RL} provide the reference supply voltage inputs.

A multiplexer allows the single A/D converter to select one of 16 analog input signals.

The A/D converter control logic implements automatic conversion sequences on a selected channel four times or on four channels once each. A write to the ADCTL register initiates conversions and, if made while a conversion is in progress, a write to ADCTL also halts that conversion operation, sets CCF, and proceeds to the next instruction.

When the SCAN bit is zero, four requested conversions are performed, once each, to fill the four result registers. When SCAN is one, conversions continue in a round-robin fashion with the result registers being updated as new data becomes available. When the MULT bit is zero, the A/D converter system is configured to perform conversions on each channel in the group of four channels specified by the CD and CC channel select bits.

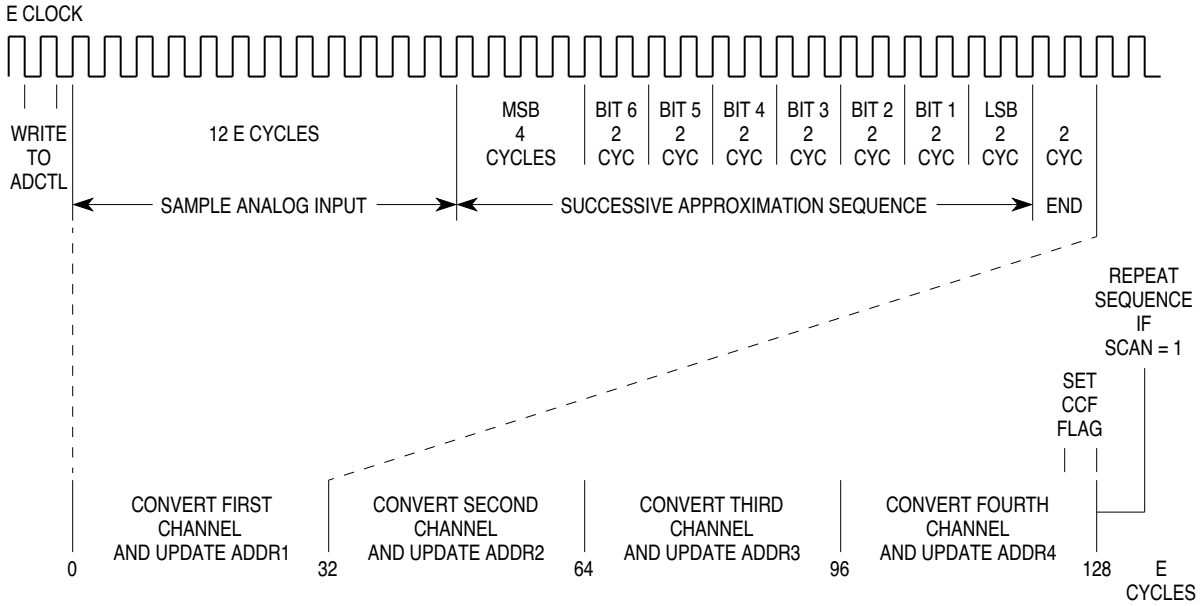
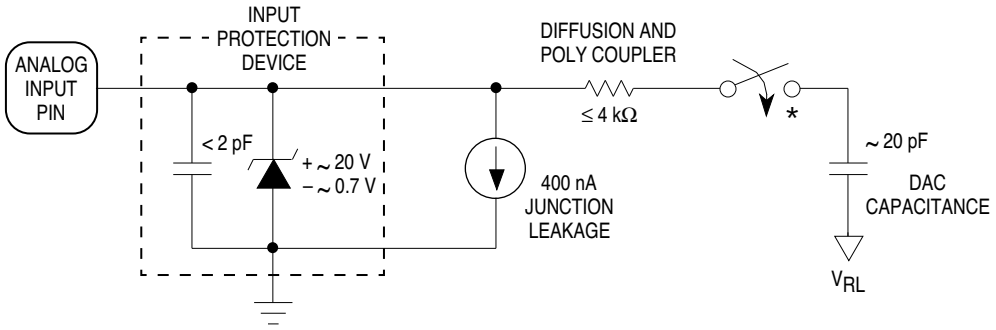


Figure 15 Timing Diagram for a Sequence of Four A/D Conversions



* This analog switch is closed only during the 12-cycle sample time.

Figure 16 Electrical Model of an Analog Input Pin (Sample Mode)

ADCTL —A/D Control/Status **$0030**

	Bit 7	6	5	4	3	2	1	Bit 0
	CCF	—	SCAN	MULT	CD	CC	CB	CA
RESET:	0	0	0	0	0	0	0	0

CCF — Conversions Complete Flag
0 = Write to ADCTL is complete
1 = A/D conversion cycle is complete

Bit 6 — Not implemented
Always reads zero

SCAN —Continuous Scan Control
0 = Do four conversions and stop
1 = Convert four channels in selected group continuously

MULT —Multiple Channel/Single Channel Control
0 = Convert single channel selected
1 = Convert four channels in selected group

CD:CA —Channel Select D through A

Table 9 A/D Converter Channel Assignments

Channel Select Control Bits				Channel Signal	Result in ADRx if MULT = 1
CD	CC	CB	CA		
0	0	0	0	AN0	ADR1
0	0	0	1	AN1	ADR2
0	0	1	0	AN2	ADR3
0	0	1	1	AN3	ADR4
0	1	0	0	AN4	ADR1
0	1	0	1	AN5	ADR2
0	1	1	0	AN6	ADR3
0	1	1	1	AN7	ADR4
1	0	0	0	Reserved	—
1	0	0	1	Reserved	—
1	0	1	0	Reserved	—
1	0	1	1	Reserved	—
1	1	0	0	V_{RH}*	ADR1
1	1	0	1	V_{RL}*	ADR2
1	1	1	0	$(V_{RH})/2$*	ADR3
1	1	1	1	Reserved*	ADR4

*Used for factory testing

ADR[4:1] —A/D Results **$0031 – $0034**

$0031	Bit 7	6	5	4	3	2	1	Bit 0	ADR1
$0032	Bit 7	6	5	4	3	2	1	Bit 0	ADR2
$0033	Bit 7	6	5	4	3	2	1	Bit 0	ADR3
$0034	Bit 7	6	5	4	3	2	1	Bit 0	ADR4

OPTION —System Configuration Options **$0039**

	Bit 7	6	5	4	3	2	1	Bit 0
	ADPU	CSEL	IRQE*	DLY*	CME	FCME*	CR1*	CR0*
RESET:	0	0	0	1	0	0	0	0

ADPU —A/D Converter Power-Up
0 = A/D converter powered down
1 = A/D converter powered up

CSEL — Clock Select
0 = A/D and EEPROM use system E clock
1 = A/D and EEPROM use internal RC clock source

IRQE —$\overline{\text{IRQ}}$ Select Edge Sensitive Only
Refer to **5 Resets and Interrupts**

DLY —Enable Oscillator Startup Delay on Exit from Stop
Refer to **5 Resets and Interrupts**

CME —Clock Monitor Enable
Refer to **5 Resets and Interrupts**

FCME —Force Clock Monitor Enable
Refer to **5 Resets and Interrupts**

CR[1:0] —COP Timer Rate Select
Refer to **10 Main Timer**

10 Main Timer

The timing system is based on a free-running 16-bit counter with a four-stage programmable prescaler. A timer overflow function allows software to extend the timing capability of the system beyond the 16-bit range of the counter.

The timer has three channels for input capture, four channels for output compare, and one channel that can be configured as a fourth input capture or a fifth output compare. In addition, the timing system includes pulse accumulator and real-time interrupt (RTI) functions, as well as a clock monitor function, which can be used to detect clock failures that are not detected by the COP.

Refer to **11 Pulse Accumulator** and **10.1 Real-Time Interrupt** for further information about these functions. Refer to the following table for a summary of the crystal-related frequencies and periods.

Table 10 Timer Summary

Control Bits	Common System Frequencies			Definition
	8.0 MHz	**12.0 MHz**	**16.0 MHz**	**XTAL**
	2.0 MHz	**3.0 MHz**	**4.0 MHz**	**E**
PR[1:0]	**Main Timer Count Rates (Period Length)**			
0 0 1 count — overflow —	 500 ns 32.768 ms	 333 ns 21.845 ms	 250 ns 16.384 ms	 1/E 2^{16}/E
0 1 1 count — overflow —	 2.0 µs 131.07 ms	 1.333 µs 87.381 ms	 1.0 µs 65.536 ms	 4/E 2^{18}/E
1 0 1 count — overflow —	 4.0 µs 262.14 ms	 2.667 µs 174.76 ms	 2.0 µs 131.07 ms	 8/E 2^{19}/E
1 1 1 count — overflow —	 8.0 µs 524.29 ms	 5.333 µs 349.52 ms	 4.0 µs 262.14 ms	 16/E 2^{20}/E
RTR[1:0]	**Periodic (RTI) Interrupt Rates (Period Length)**			
0 0 0 1 1 0 1 1	4.096 ms 8.192 ms 16.384 ms 32.768 ms	2.731 ms 5.461 ms 10.923 ms 21.845 ms	2.048 ms 4.096 ms 8.192 ms 16.384 ms	2^{13}/E 2^{14}/E 2^{15}/E 2^{16}/E
CR[1:0]	**COP Watchdog Timeout Rates (Period Length)**			
0 0 0 1 1 0 1 1	16.384 ms 65.536 ms 262.14 ms 1.049 s	10.923 ms 43.691 ms 174.76 ms 699.05 ms	8.192 ms 32.768 ms 131.07 ms 524.28 ms	2^{15}/E 2^{17}/E 2^{19}/E 2^{21}/E
Timeout Tolerance (–0 ms/+...)	 16.4 ms	 10.9 ms	 8.192 ms	 2^{15}/E

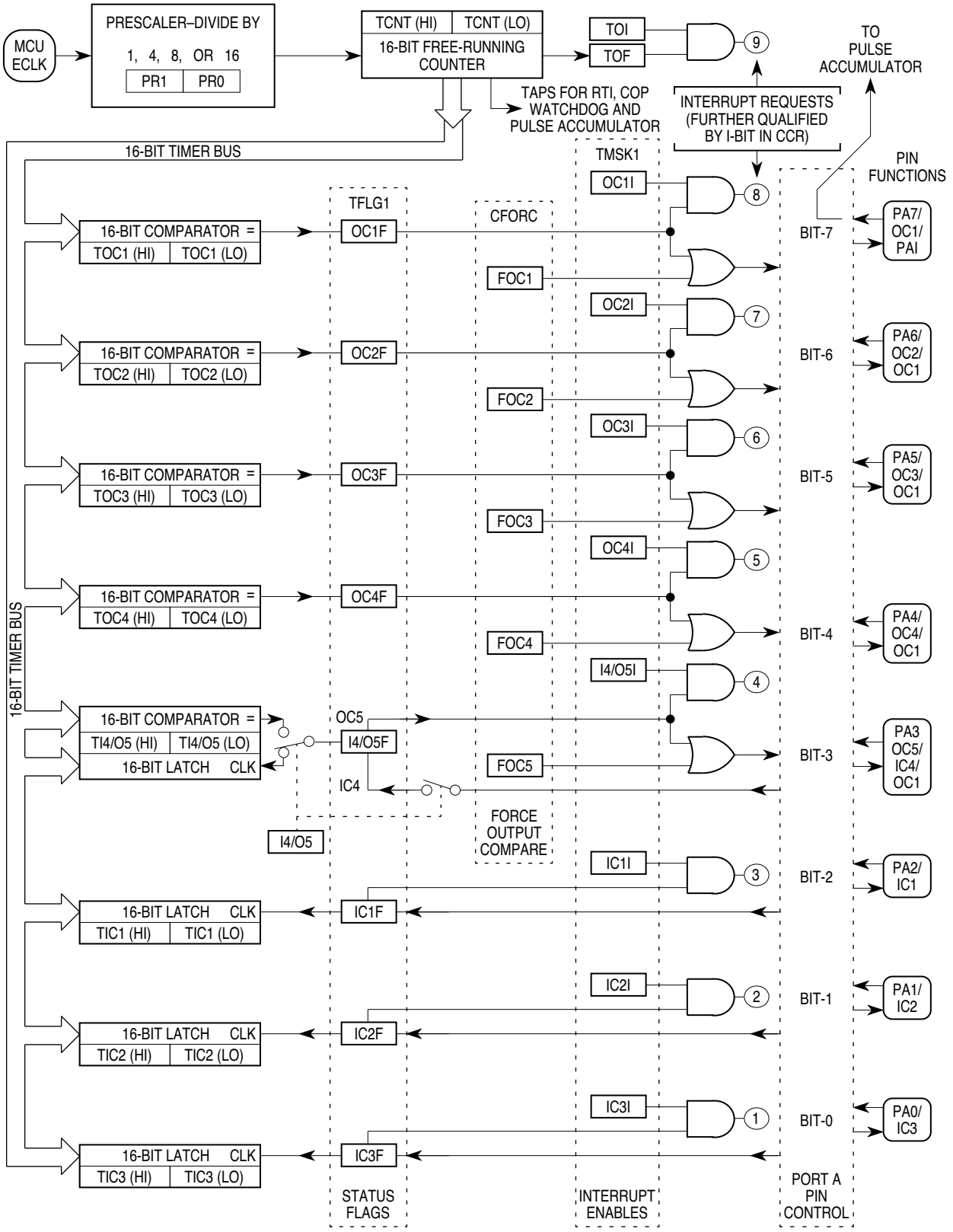


Figure 17 Timer Block Diagram

CFORC —Timer Compare Force **$000B**

	Bit 7	6	5	4	3	2	1	Bit 0
	FOC1	FOC2	FOC3	FOC4	FOC5	—	—	—
RESET:	0	0	0	0	0	0	0	0

FOC[5:1] —Force Output Compare
Write ones to force compare(s)
0 = Not affected
1 = Output x action occurs

Bits [2:0] —Not implemented
Always read zero

OC1M —Output Compare 1 Mask **$000C**

	Bit 7	6	5	4	3	2	1	Bit 0
	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	—	—	—
RESET:	0	0	0	0	0	0	0	0

Set bit(s) to enable OC1 to control corresponding pin(s) of port A

Bits [2:0] —Not implemented
Always read zero

OC1D —Output Compare 1 Data **$000D**

	Bit 7	6	5	4	3	2	1	Bit 0
	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	—	—	—
RESET:	0	0	0	0	0	0	0	0

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

Bits [2:0] —Not implemented
Always read zero

TCNT —Timer Count **$000E, $000F**

$000E	Bit 15	14	13	12	11	10	9	Bit 8	High	**TCNT**
$000F	Bit 7	6	5	4	3	2	1	Bit 0	Low	

TCNT resets to $0000. In normal modes, TCNT is read only.

TIC1–TIC3 —Timer Input Capture **$0010–$0015**

$0010	Bit 15	14	13	12	11	10	9	Bit 8	High	**TIC1**
$0011	Bit 7	6	5	4	3	2	1	Bit 0	Low	
$0012	Bit 15	14	13	12	11	10	9	Bit 8	High	**TIC2**
$0013	Bit 7	6	5	4	3	2	1	Bit 0	Low	
$0014	Bit 15	14	13	12	11	10	9	Bit 8	High	**TIC3**
$0015	Bit 7	6	5	4	3	2	1	Bit 0	Low	

TICx not affected by reset

TOC1–TOC4 —Timer Output Compare **$0016–$001D**

$0016	Bit 15	14	13	12	11	10	9	Bit 8	High	**TOC1**
$0017	Bit 7	6	5	4	3	2	1	Bit 0	Low	
$0018	Bit 15	14	13	12	11	10	9	Bit 8	High	**TOC2**
$0019	Bit 7	6	5	4	3	2	1	Bit 0	Low	
$001A	Bit 15	14	13	12	11	10	9	Bit 8	High	**TOC3**
$001B	Bit 7	6	5	4	3	2	1	Bit 0	Low	
$001C	Bit 15	14	13	12	11	10	9	Bit 8	High	**TOC4**
$001D	Bit 7	6	5	4	3	2	1	Bit 0	Low	

All TOCx register pairs reset to ones ($FFFF).

TI4/O5 —Timer Input Capture 4/Output Compare 5 **$001E–$001F**

$001E	Bit 15	14	13	12	11	10	9	Bit 8	High
$001F	Bit 7	6	5	4	3	2	1	Bit 0	Low

This is a shared register and is either input capture 4 or output compare 5 depending on the state of bit I4/O5 in PACTL. Writes to TI4/O5 have no effect when this register is configured as input capture 4. The TI4/O5 register pair resets to ones ($FFFF).

TCTL1 —Timer Control 1 **$0020**

	Bit 7	6	5	4	3	2	1	Bit 0
	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5
RESET:	0	0	0	0	0	0	0	0

OM[5:2] —Output Mode

OL[5:2] —Output Level

OMx	OLx	Action Taken on Successful Compare
0	0	Timer disconnected from output pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to zero
1	1	Set OCx output line to one

TCTL2 —Timer Control 2 **$0021**

	Bit 7	6	5	4	3	2	1	Bit 0
	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
RESET:	0	0	0	0	0	0	0	0

Table 11 Timer Control Configuration

EDGxB	EDGxA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any edge

TMSK1 —Timer Interrupt Mask 1 **$0022**

	Bit 7	6	5	4	3	2	1	Bit 0
	OC1I	OC2I	OC3I	OC4I	I4/O5I	IC1I	IC2I	IC3I
RESET:	0	0	0	0	0	0	0	0

OC1I–OC4I —Output Compare x Interrupt Enable
If the OCxF flag bit is set while the OCxI enable bit is set, a hardware interrupt sequence is requested.

I4/O5I —Input Capture 4 or Output Compare 5 Interrupt Enable
When I4/O5 in PACTL is one, I4/O5I is the input capture 4 interrupt bit. When I4/O5 in PACTL is zero, I4/O5I is the output compare 5 interrupt control bit.

IC1I–IC3I —Input Capture x Interrupt Enable
If the ICxF flag bit is set while the ICxI enable bit is set, a hardware interrupt sequence is requested.

TFLG1 —Timer Interrupt Flag 1 **$0023**

	Bit 7	6	5	4	3	2	1	Bit 0
	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F
RESET:	0	0	0	0	0	0	0	0

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

OC1F–OC5F —Output Compare x Flag
Set each time the counter matches output compare x value

I4/O5F —Input Capture 4/Output Compare 5 Flag
Set by IC4 or OC5, depending on which function was enabled by I4/O5 of PACTL

IC1F–IC3F —Input Capture x Flag
Set each time a selected active edge is detected on the ICx input line

NOTE

Control bits in TMSK1 correspond bit for bit with flag bits in TFLG1. Ones in TMSK1 enable the corresponding interrupt sources.

TMSK2 —Timer Interrupt Mask 2 **$0024**

	Bit 7	6	5	4	3	2	1	Bit 0
	TOI	RTII	PAOVI	PAII	—	—	PR1	PR0
RESET:	0	0	0	0	0	0	0	0

TOI —Timer Overflow Interrupt Enable
0 = Timer overflow interrupt disabled
1 = Timer overflow interrupt enabled

RTII —Real-Time Interrupt Enable
0 = RTIF interrupts disabled
1 = Interrupt requested when RTIF is set to one.

PAOVI —Pulse Accumulator Overflow Interrupt Enable
Refer to **11 Pulse Accumulator**.

PAII —Pulse Accumulator Interrupt Enable
Refer to **11 Pulse Accumulator**.

NOTE

Control bits [7:4] in TMSK2 correspond bit for bit with flag bits [7:4] in TFLG2. Ones in TMSK2 enable the corresponding interrupt sources.

Bits [3:2] —Not implemented
Always read zero

PR[1:0] —Timer Prescaler Select
In normal modes, PR1 and PR0 can only be written once, and the write must occur within 64 cycles after reset. Refer to **Table 10** for specific timing values.

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

TFLG2 —Timer Interrupt Flag 2 **$0025**

	Bit 7	6	5	4	3	2	1	Bit 0
	TOF	RTIF	PAOVF	PAIF	—	—	—	—
RESET:	0	0	0	0	0	0	0	0

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

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

RTIF —Real-Time (Periodic) Interrupt Flag
0 = No RTI interrupt
1 = RTI interrupt request pending

PAOVF —Pulse Accumulator Overflow Flag
Refer to **11 Pulse Accumulator**.

PAIF —Pulse Accumulator Input Edge Flag
Refer to **11 Pulse Accumulator**.

Bits [3:0] —Not implemented
Always read zero

PACTL —Pulse Accumulator Control **$0026**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	PAEN	PAMOD	PEDGE	—	I4/O5	RTR1	RTR0
RESET:	0	0	0	0	0	0	0	0

Bit 7 —Not implemented
Always read zero

PAEN —Pulse Accumulator System Enable
Refer to **11 Pulse Accumulator**.

PAMOD —Pulse Accumulator Mode
Refer to **11 Pulse Accumulator**.

PEDGE —Pulse Accumulator Edge Control
Refer to **11 Pulse Accumulator**.

Bit 3 —Not implemented
Always reads zero

I4/O5 —Input Capture 4/Output Compare 5
Configure TI4/O5 for input capture or output compare
0 = OC5 enabled
1 = IC4 enabled

RTR[1:0] —Real-Time Interrupt (RTI) Rate
Refer to **10.1 Real-Time Interrupt**.

10.1 Real-Time Interrupt

These rates are a function of the MCU oscillator frequency and the value of the software-accessible control bits, RTR1 and RTR0. These bits determine the rate at which interrupts are requested by the RTI system. The RTI system is driven by an E divided by 2^{13} rate clock compensated so that it is independent of the timer prescaler. The RTR1 and RTR0 control bits select an additional division factor. RTI is set to its fastest rate by default out of reset and can be changed at any time.

Table 12 Real-Time Interrupt Rates (Period Length)

RTR[1:0]	Period Length Selected	Period Length		
		E = 2.0 MHz	E = 3.0 MHz	E = 4.0 MHz
0 0	$2^{13} \div E$	4.096 ms	2.731 ms	2.048 ms
0 1	$2^{14} \div E$	8.192 ms	5.461 ms	4.096 ms
1 0	$2^{15} \div E$	16.384 ms	10.923 ms	8.192 ms
1 1	$2^{16} \div E$	32.768 ms	21.845 ms	16.383 ms

Table 13 Real-Time Interrupt Rates (Frequency)

RTR[1:0]	Rate Selected	Frequency		
		E = 2.0 MHz	E = 3.0 MHz	E = 4.0 MHz
0 0	$E \div 2^{13}$	244.141 Hz	366.211 Hz	488.281 Hz
0 1	$E \div 2^{14}$	122.070 Hz	183.105 Hz	244.141 Hz
1 0	$E \div 2^{15}$	61.035 Hz	91.553 Hz	122.070 Hz
1 1	$E \div 2^{16}$	30.518 Hz	45.776 Hz	61.035 Hz

11 Pulse Accumulator

M68HC11 K-series MCUs have an 8-bit counter that can be configured as a simple event counter or for gated time accumulation. The counter can be read or written at any time.

The port A bit 7 I/O pin can be configured to act as a clock in event counting mode, or as a gate signal to enable a free-running clock (E divided by 64) to the 8-bit counter in gated time accumulation mode.

		Common XTAL Frequencies		
		8.0 MHz	**12.0 MHz**	**16.0 MHz**
CPU Clock	**(E)**	**2.0 MHz**	**3.0 MHz**	**4.0 MHz**
Cycle Time	**(1/E)**	**500 ns**	**333 ns**	**250 ns**
Pulse Accumulator (Gated Mode)				
1 Count —	**(2^6/E)**	**32.0 μs**	**21.33 μs**	**16.0 μs**
Overflow —	**(2^{14}/E)**	**8.192 ms**	**5.461 ms**	**4.096 ms**

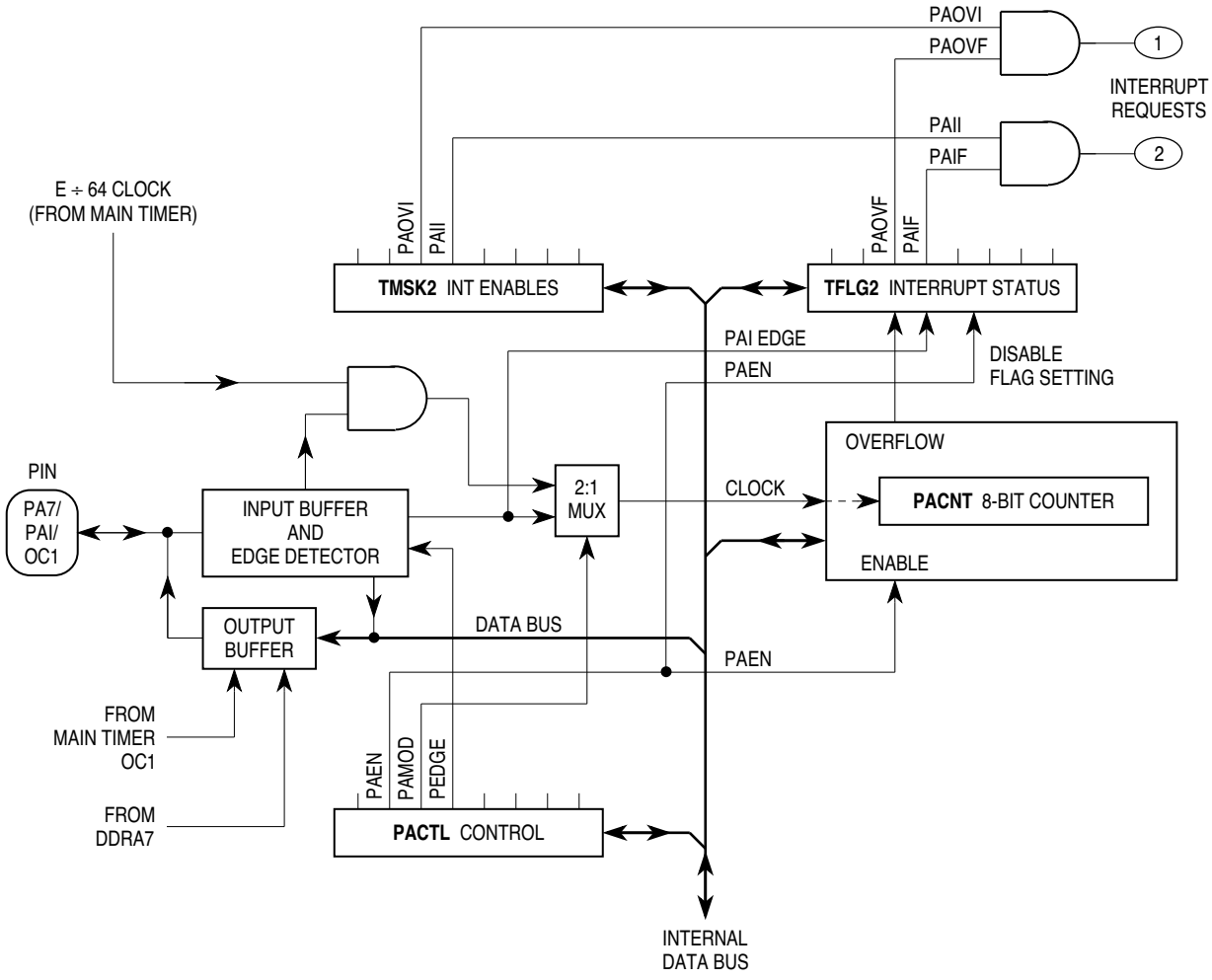


Figure 18 Pulse Accumulator System Block Diagram

TMSK2 —Timer Interrupt Mask 2 **$0024**

	Bit 7	6	5	4	3	2	1	Bit 0
	TOI	RTII	PAOVI	PAII	—	—	PR1	PR0
RESET:	0	0	0	0	0	0	0	0

TOI —Timer Overflow Interrupt Enable
Refer to **10 Main Timer**.

RTII —Real-Time Interrupt Enable
Refer to **10 Main Timer**.

PAOVI —Pulse Accumulator Overflow Interrupt Enable
0 = Pulse accumulator overflow interrupt disabled
1 = Pulse accumulator overflow interrupt enabled

PAII —Pulse Accumulator Input Interrupt Enable
0 = Pulse accumulator input interrupt disabled
1 = Pulse accumulator input interrupt enabled if PAIF bit in TFLG2 register is set

Bits [3:2] —Not implemented
Always read zero

PR[1:0] —Timer Prescaler Select
Refer to **10 Main Timer**.

NOTE

Control bits [7:4] in TMSK2 correspond bit for bit with flag bits [7:4] in TFLG2. Ones in TMSK2 enable the corresponding interrupt sources.

TFLG2 —Timer Interrupt Flag 2 **$0025**

	Bit 7	6	5	4	3	2	1	Bit 0
	TOF	RTIF	PAOVF	PAIF	—	—	—	—
RESET:	0	0	0	0	0	0	0	0

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

TOF —Timer Overflow Enable
Refer to **10 Main Timer**.

RTIF —Real-Time Interrupt Flag
Refer to **10 Main Timer**.

PAOVF —Pulse Accumulator Overflow Flag
Set when PACNT changes from $FF to $00

PAIF —Pulse Accumulator Input Edge Flag
Set each time a selected active edge is detected on the PAI input line

Bits [3:0] —Not implemented
Always read zero

PACTL —Pulse Accumulator Control **$0026**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	PAEN	PAMOD	PEDGE	—	I4/O5	RTR1	RTR0
RESET:	0	0	0	0	0	0	0	0

Bit 7 —Not implemented
Always reads zero

PAEN —Pulse Accumulator System Enable
0 = Pulse accumulator disabled
1 = Pulse accumulator enabled

PAMOD —Pulse Accumulator Mode
0 = Event counter
1 = Gated time accumulation

PEDGE —Pulse Accumulator Edge Control
0 = In event mode, falling edges increment counter. In gated accumulation mode, high level enables accumulator and falling edge sets PAIF.
1 = In event mode, rising edges increment counter. In gated accumulation mode, low level enables accumulator and rising edge sets PAIF.

I4/O5 —Input Capture 4/Output Compare 5
Refer to **10 Main Timer**.

RTR[1:0] —Real-Time Interrupt Rate
Refer to **10 Main Timer**.

PACNT —Pulse Accumulator Counter **$0027**

Bit 7	6	5	4	3	2	1	Bit 0
Bit 7	6	5	4	3	2	1	Bit 0

Can be read and written.

12 Pulse-Width Modulation Timer

M68HC11 K-series MCUs contains a PWM timer that is composed of a four-channel 8-bit modulator. Each of the modulators can create independent continuous waveforms with software-selectable duty rates from 0% to 100%.

The PWM provides up to four pulse-width modulated waveforms on port H pins. Each channel has its own counter. Pairs of counters can be concatenated to create 16-bit PWM outputs based on 16-bit counts. Three clock sources (A, B, and S) and a flexible clock select scheme give the PWM system a wide range of frequencies.

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 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 channels configured for 8-bit mode and E = 4 MHz, PWM signals of 40 kHz (1% duty cycle resolution) to less than 10 Hz (approximately 0.4% duty cycle resolution) can be produced. By configuring the channels for 16-bit mode with E = 4 MHz, PWM periods greater than one minute are possible.

In 16-bit mode, duty cycle resolution of almost 15 parts per million can be achieved (at a PWM frequency of about 60 Hz). In the same system, a PWM frequency of 1 kHz corresponds to a duty cycle resolution of 0.025%.

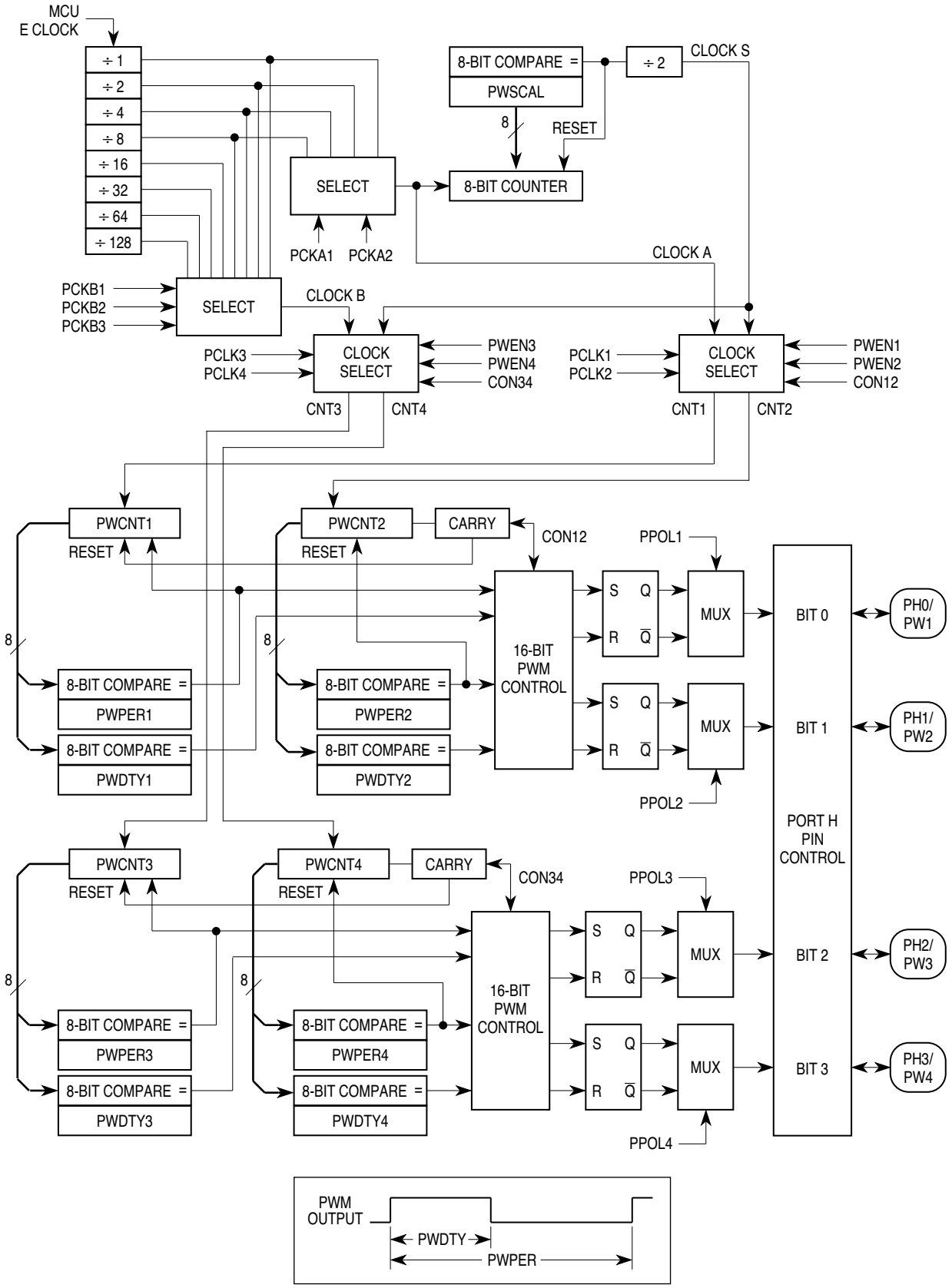


Figure 19 Pulse-Width Modulation Block Diagram

PWCLK —Pulse-Width Modulation Clock Select **$0060**

	Bit 7	6	5	4	3	2	1	Bit 0
	CON34	CON12	PCKA2	PCKA1	—	PCKB3	PCKB2	PCKB1
RESET:	0	0	0	0	0	0	0	0

CON34 —Concatenate Channels 3 and 4

Channel 3 is high-order byte, and channel 4 is the low-order byte. The resulting output is available on port H, pin 3. Clock source is determined by PCLK4.

0 = Channels 3 and 4 are separate 8-bit PWMs.

1 = Channels 3 and 4 are concatenated to create one 16-bit PWM channel.

CON12 —Concatenate Channels One and Two

Channel 1 is high-order byte, and channel 2 is the low-order byte. The resulting output is available on port H, pin 1. Clock source is determined by PCLK2.

0 = Channels 1 and 2 are separate 8-bit PWMs.

1 = Channels 1 and 2 are concatenated to create one 16-bit PWM channel.

PCKA[2:1] —Prescaler for Clock A (See also PWSCAL register)

Determines the rate of clock A

PCKA[2:1]	Value of Clock A
0 0	E
0 1	E/2
1 0	E/4
1 1	E/8

Bit 3 —Not implemented

Always reads zero

PCKB[3:1] —Prescaler for Clock B

Determines the rate for clock B

PCKB[3:1]	Value of 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

PWPOL —Pulse-Width Modulation Timer Polarity **$0061**

	Bit 7	6	5	4	3	2	1	Bit 0
	PCLK4	PCLK3	PCLK2	PCLK1	PPOL4	PPOL3	PPOL2	PPOL1
RESET:	0	0	0	0	0	0	0	0

PCLK4 —Pulse-Width Channel 4 Clock Select

0 = Clock B is source

1 = Clock S is source

PCLK3 —Pulse-Width Channel 3 Clock Select
0 = Clock B is source
1 = Clock S is source

PCLK2 —Pulse-Width Channel 2 Clock Select
0 = Clock A is source
1 = Clock S is source

PCLK1 —Pulse-Width Channel 1 Clock Select
0 = Clock A is source
1 = Clock S is source

PPOL[4:1] —Pulse-Width Channel x Polarity
0 = PWM channel x output is low at the beginning of the clock cycle and goes high when duty count is reached
1 = PWM channel x output is high at the beginning of the clock cycle and goes low when duty count is reached

PWSCAL —Pulse-Width Modulation Timer Prescaler **$0062**

	Bit 7	6	5	4	3	2	1	Bit 0
	7	6	5	4	3	2	1	0
RESET:	0	0	0	0	0	0	0	0

Scaled clock S is generated by dividing clock A by the value in PWSCAL, then dividing the result by 2. If PWSCAL = $00, divide clock A by 256, then divide the result by 2.

PWEN —Pulse-Width Modulation Timer Enable **$0063**

	Bit 7	6	5	4	3	2	1	Bit 0
	TPWSL	DISCP	—	—	PWEN4	PWEN3	PWEN2	PWEN1
RESET:	0	0	0	0	0	0	0	0

TPWSL —PWM Scaled Clock Test Bit (TEST)
Factory test only

DISCP —Disable Compare Scaled E Clock (TEST)
Factory test only

Bits [5:4] —Not implemented
Always read zero

PWEN[4:1] —Pulse-Width Channel 4–1
0 = Channel disabled
1 = Channel enabled

PWCNT1–PWCNT4 —Pulse-Width Modulation Timer Counter 1 to 4 **$0064–$0067**

	Bit 7	6	5	4	3	2	1	Bit 0	
$0064	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT1
$0065	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT2
$0066	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT3
$0067	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT4
RESET:	0	0	0	0	0	0	0	0	

PWCNT1–PWCNT4
Begins count using whichever clock was selected

PWPER1–PWPER4 —Pulse-Width Modulation Timer Period 1 to 4 **$0068–$006B**

$0068	Bit 7	6	5	4	3	2	1	Bit 0	PWPER1
$0069	Bit 7	6	5	4	3	2	1	Bit 0	PWPER2
$006A	Bit 7	6	5	4	3	2	1	Bit 0	PWPER3
$006B	Bit 7	6	5	4	3	2	1	Bit 0	PWPER4
RESET:	1	1	1	1	1	1	1	1	

PWPER1–PWPER4

Determines period of associated PWM channel

PWDTY1–4 —Pulse-Width Modulation Timer Duty Cycle 1 to 4 **$006C–$006F**

	Bit 7	6	5	4	3	2	1	Bit 0	
$006C	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY1
$006D	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY2
$006E	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY3
$006F	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY4
RESET:	1	1	1	1	1	1	1	1	

PWDTY1–4

Determines duty cycle of associated PWM channel

12.1 PWM Boundary Cases

Certain values written to PWM control registers, counters, etc. can cause outputs that are not what the user might expect. These are referred to as boundary cases. Boundary cases occur when the user specifies a value that is either a maximum or a minimum. This value combined with other conditions causes unexpected behavior of the PWM system.

The following conditions always cause the corresponding output to be high:

PWDTYx = $00, PWPERx > $00, and PPOLx = 0

PWDTYx ≥PWPERx, and PPOLx = 1

PWPERx = $00 and PPOLx = 1

The following conditions always cause the corresponding output to be low:

PWDTYx = $00, PWPERx > $00, and PPOLx = 1

PWDTYx ≥PWPERx, and PPOLx = 0

PWPERx = $00 and PPOLx = 0

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 which may be provided in Motorola data sheets and/or specifications 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.

MCUinit, MCUasm, MCUdebug, and RTEK are trademarks of Motorola, Inc. Motorola and the Motorola logo are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

How to reach us:
USA/EUROPE/Locations Not Listed: Motorola Literature Distribution;
P.O. Box 5405, Denver Colorado 80217. 1-800-441-2447, (303) 675-2140
Mfax™: RMFAX0@email.sps.mot.com - TOUCHTONE (602) 244-6609
INTERNET: http://Design-NET.com
JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC,
6F Seibu-Butsuryu-Center, 3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 81-3-3521-8315
ASIA PACIFIC: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park,
51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298
Mfax is a trademark of Motorola, Inc.

MC68HC11KTS/D