MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

Order this document
by MC68HC11C0TS/D

MC68HC11C0

Technical Summary
8-Bit Microcontroller

1 Introduction

The MC68HC11C0 high-performance microcontroller unit (MCU) is an enhanced member of the M68HC11 family of microcontrollers. Excluding its new features, the MC68HC11C0 is very similar to the MC68HC11E9 MCU. This device incorporates highly sophisticated on-chip peripheral functions and, with a multiplexed expanded bus, is characterized by high speed and low power consumption. Its fully static design allows this device to operate at frequencies from 3 MHz to dc.

The MC68HC11C0 has the ability to extend the address range of the M68HC11 CPU beyond the 64-Kbyte limit of the 16 CPU address lines. The extra addressing capability is provided by a register-based paging scheme using two additional expansion address lines and the 64 Kbytes of CPU address space. Six chip-select signals are provided to simplify the interface to external peripheral devices.

Two 8-bit pulse-width modulation timer outputs have been added to the timer system. The two outputs have selectable polarity, duty cycle, and period. They can be concatenated to form a single 16-bit output.

In addition to the $\overline{\text{IRQ}}$ and $\overline{\text{XIRQ}}$ pins found on other M68HC11 devices, seven more interrupt request lines have been added, creating a total of one nonmaskable and eight maskable interrupt sources. Refer to the MC68HC11C0 block diagram.

Table 1 Device Ordering Information

Package	Description	CONFIG	Frequency	Temperature	MC Order Number
64-Pin QFP	No ROM	$00	2 MHz	–40°to + 85°C	MC68HC11C0CFU2
				–40°to + 105°C	MC68HC11C0VFU2
				–40°to + 125°C	MC68HC11C0MFU2
			3 MHz	0°to + 70°C	MC68HC11C0FU3
				–40°to + 85°C	MC68HC11C0CFU3
68-Pin PLCC	No ROM	$00	2 MHz	–40°to + 85°C	MC68HC11C0CFN2
				–40°to + 105°C	MC68HC11C0VFN2
				–40°to + 125°C	MC68HC11C0MFN2
			3 MHz	0°to + 70°C	MC68HC11C0FN3
				–40°to + 85°C	MC68HC11C0CFN3

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

1.1 Features

- M68HC11 CPU
- 256 Bytes of On-Chip Static RAM
- 1024 Bytes of Bootstrap ROM (Available in Single-Chip, Bootstrap, and Special-Test Modes)
- Power Saving STOP and WAIT Modes
- 64 Kbyte Address Space, Expandable to 256 Kbytes Using On-Chip Memory Mapping Logic
- Multiplexed Address/Data Bus
- 16-Bit Timer System
  - — Three Input Capture (IC) Channels
  - — Four Output Compare (OC) Channels
  - — One Additional Channel, Software Selectable as Fourth IC or Fifth OC
- 8-Bit Pulse Accumulator
- Real-Time Interrupt Circuit
- Computer Operating Properly (COP) Watchdog Timer
- Clock Monitor
- Five External General-Purpose Chip Select Signals, Each with Programmable Clock Stretching
- One External Vector/Program Chip Select with Programmable Clock Stretching
- Nine External Interrupt Request Inputs (One Nonmaskable Interrupt)
- Two 8-Bit Pulse-Width Modulation (PWM) Timer Channels (Concatenate for a Single 16-Bit PWM)
- Four-Channel 8-Bit Analog-to-Digital (A/D) Converter
- Asynchronous Nonreturn to Zero (NRZ) Serial Communications Interface (SCI)
- Synchronous Serial Peripheral Interface (SPI)
- Six Input/Output (I/O) Ports (35 Pins)
  - — 31 Bidirectional
  - — 4 Input Only
- All Bidirectional Port Pins Have Selectable Internal Pull-Up Devices
- Available in 68-Pin Plastic Leaded Chip Carrier (PLCC) and 64-Pin Quad Flat Pack (QFP)

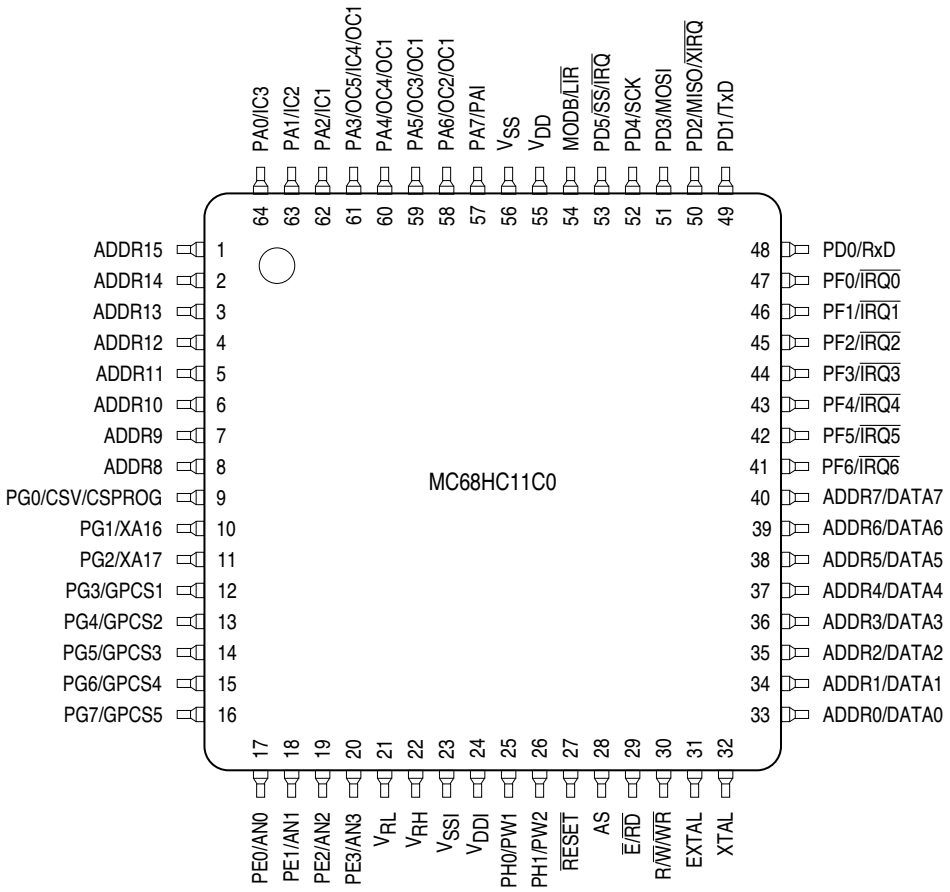


Figure 1 Pin Assignments for 64-Pin Quad Flat Pack

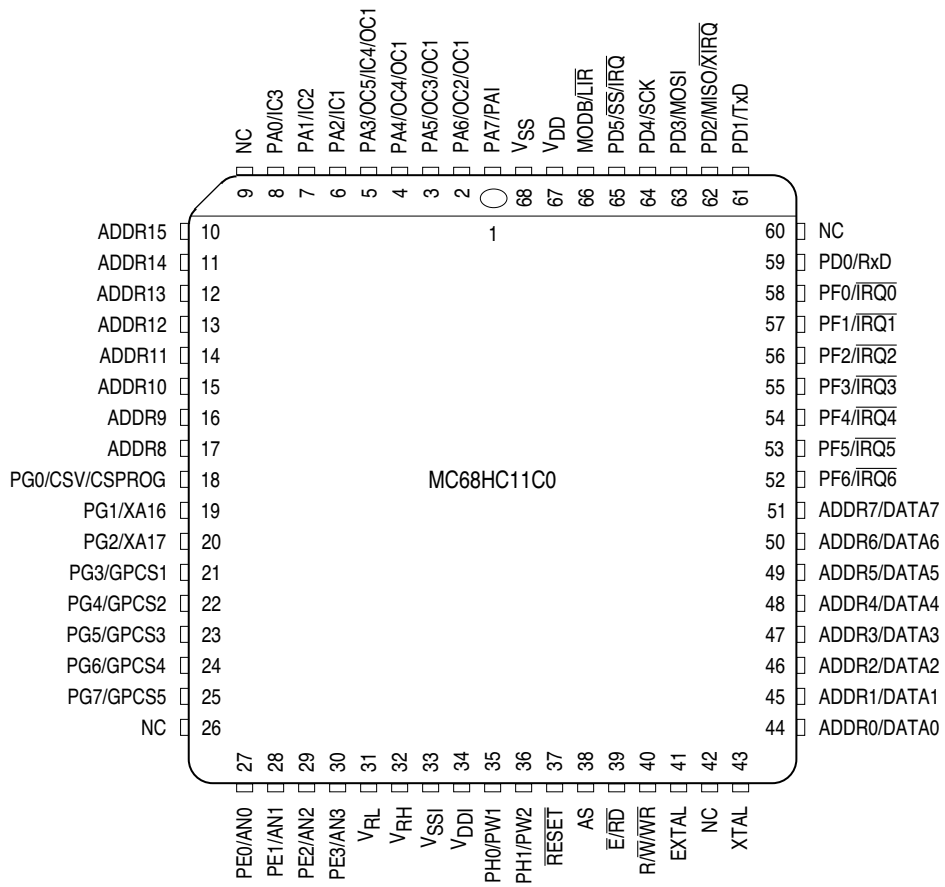


Figure 2 Pin Assignments for 68-Pin PLCC

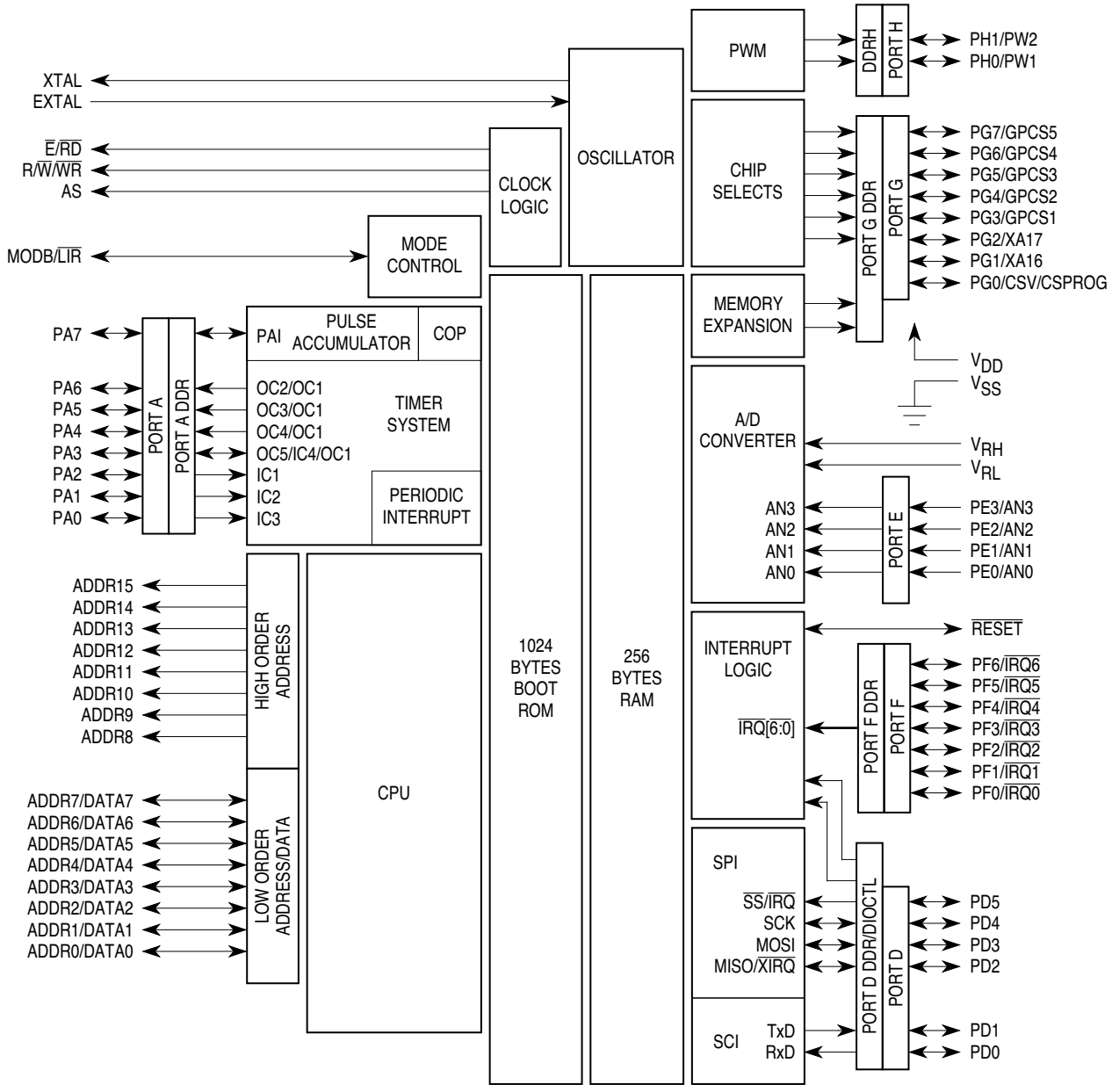


Figure 3 MC68HC11C0 Block Diagram

TABLE OF CONTENTS

2 Operating Modes

The MC68HC11C0 has four modes of operation. These modes directly affect the address space and the memory map differs for each of them. Refer to the memory map diagram and the following paragraphs.

2.1 Expanded Mode

In expanded mode, the MCU can access the full 64-Kbyte address space. The space includes the same on-chip memory addresses used for single-chip mode as well as addresses for external peripherals and memory devices. The 256-byte block of RAM is accessible in expanded mode but is disabled after reset. To enable RAM in expanded mode, set the RAMON bit in the CONFIG register. Vectors are fetched from **external** locations $FFC0–$FFFF. The expansion bus consists of sixteen address lines (ADDR[15:0]) and eight data lines (DATA[7:0]). The read/write (R/$\overline{W}$), read ($\overline{RD}$), write ($\overline{WR}$) and address strobe (AS) signals are outputs that reflect the state of the internal data bus and are used to control the direction of data on the data bus. The low-order address lines and the 8-bit data bus are time multiplexed on the same pins. During the first half of each bus cycle address information is present. During the second half of each bus cycle the pins become the bidirectional data bus. AS is an active-high latch enable signal for an external address latch. Address information is allowed through the transparent latch while AS is high and is latched when AS drives low. The address, R/$\overline{W}$, and AS signals are active and valid for all bus cycles, including accesses to internal memory locations. The E-clock signal ($\overline{E}$) is used to enable external devices to drive data onto the internal data bus during the second half of a read bus cycle (E clock low). Unlike other M68HC11 devices, the MC68HC11C0 inverts the E clock signal before driving it out of the chip. R/$\overline{W}$ controls the direction of data transfers. R/$\overline{W}$ drives low when data is being written to the data bus. R/$\overline{W}$ will remain low during consecutive data bus write cycles, such as when a double-byte store occurs. Notice that the write enable signal for an external memory is the NAND of the inverted E clock and the inverted R/$\overline{W}$ signal. Refer to the example diagram of address and data demultiplexing. A more efficient method of controlling data on the bus can be employed by use of the $\overline{RD}$ and the $\overline{WR}$ signals. Setting the RWMC bit in the CONFIG register causes the $\overline{RD}$ and the $\overline{WR}$ signals to be driven out of the chip instead of $\overline{E}$ and R/$\overline{W}$. $\overline{RD}$ asserts while a data bus read cycle is in progress. $\overline{WR}$ asserts while a data bus write cycle is in progress.

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

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	RWMC	—	—	NOCOP	RAMON	—
RESET:	0	0	0	0	0	0	0	0

In single-chip and expanded modes (SMOD = 0), CONFIG can only be written once. In special test modes (SMOD = 1), CONFIG can be written any time. Changes do not take effect until the first cycle of the instruction following the write to CONFIG.

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

RWMC — Read/Write Strobe Mode Control
0 = R/$\overline{W}$ is driven out of the chip
1 = $\overline{RD}$ and $\overline{WR}$ are driven out of the chip

Bits [4:3] — Not implemented
Always read zero

NOCOP — COP Watchdog Timer Disable
Refer to **6 Resets and Interrupts**.

RAMON — RAM Enable
Refer to **3 On-Chip Memory**.

Bit 0 — Not implemented
Always reads zero

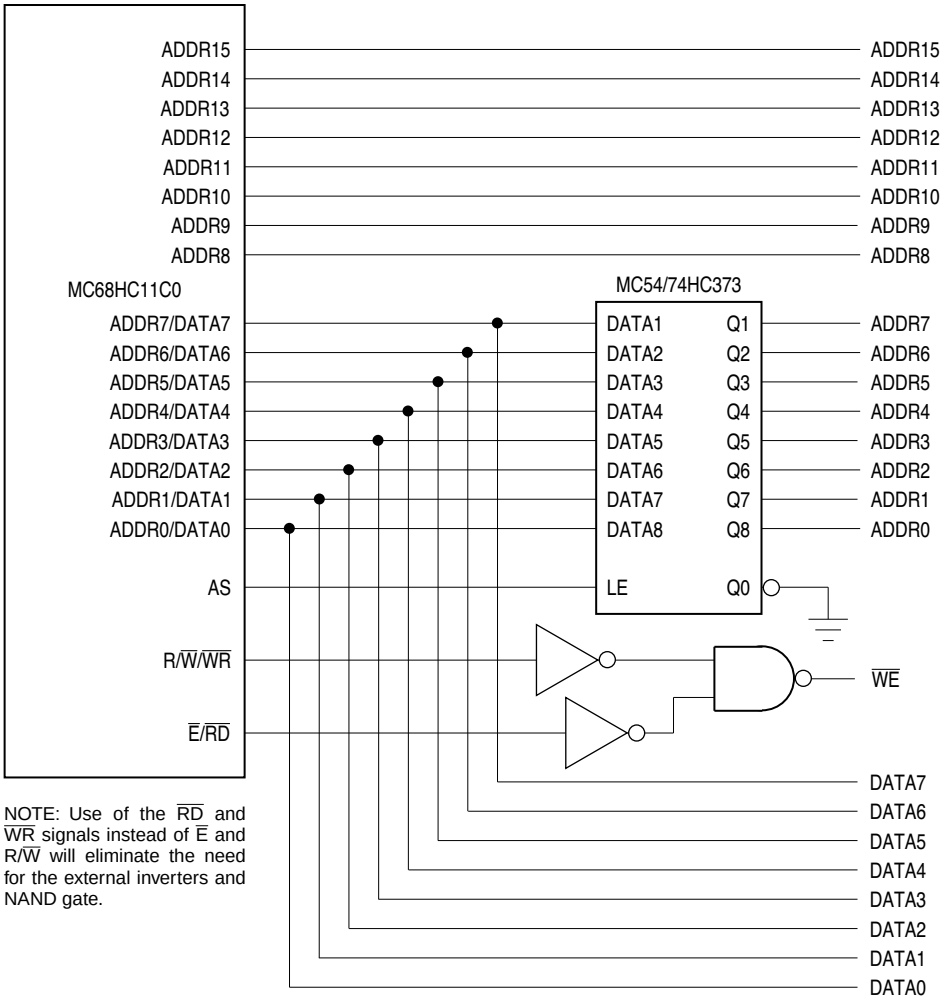


Figure 4 Address/Data Demultiplexing

2.2 Single-Chip Mode

In single-chip operating mode, the MC68HC11C0 is a stand-alone microcontroller with no external address or data bus. External address and data lines are disabled. Bootloader ROM appears in the memory map at locations $BC00–$BFFF **and** $FC00–$FFFF. Since there is no external address or data bus, the user must make certain that the RAM contains valid code before entering single-chip mode. The register block is initially located at $0000 and can be remapped to any 1-Kbyte boundary. RAM is initially located at $0400–$04FF and can be remapped to any 1-Kbyte boundary. Vectors are fetched internally from locations $FFC0–$FFFF. Refer to the memory map diagram.

2.3 Bootstrap Mode

Bootstrap mode allows special-purpose programs to be entered into internal RAM. This mode is entered by resetting to special test mode and then clearing the MDA bit in HPRIO register. When this mode is selected, a 1024-byte bootstrap ROM becomes present in the memory map. Reset and interrupt vectors

are located in internal RAM at locations $04C4–$04FD. The bootstrap ROM contains a small program which initializes the SCI and allows the user to download a program of up to 256 bytes into on-chip RAM. The program must begin at $0400. After an idle time of four-characters, or after receiving the character for address $04FF, control passes to the loaded program at $0400. Refer to the memory map diagram.

2.4 Special Test Mode

Special test mode is similar to expanded mode and is used primarily for production testing. The 1024-byte bootstrap ROM is enabled and present at locations $FC00–$FFFF. In this operating mode, vectors are fetched from external locations $BFC0–$BFFF.

2.5 Mode Selection

Although it is intended primarily for operation in expanded mode, the MC68HC11C0 has four possible operating modes. The MC68HC11C0 can be reset to either expanded mode or special-test mode. The initial operating mode is determined by the logic level present on the MODB pin during reset. After reset, the operating mode may be changed according to the table contained in the following description of the HPRIO register.

The function of internal read visibility/not E is determined by the state of the IRVNE bit and the mode selected. When enabled, internal read visibility (IRV) causes the data from internal reads to be driven out the data bus. The user must be cautioned that even though the R/$\overline{W}$ line suggests that the data bus is in a high-impedance state, data will be driven out each time an internal read occurs. The not E clock (NE) function of this bit determines whether the E clock is on or off. Refer to the description of IRVNE in the OPT2 register.

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

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

*The reset value of SMOD depends on the logic level present on the MODB pin at the rising edge of reset.

RBOOT — Read Bootstrap ROM

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

0 = Bootloader ROM disabled and not in map

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

SMOD and MDA — Special Mode Select and Mode Select A

The initial value of SMOD is the **inverse** of the logic level present on the MODB pin at the rising edge of reset. The reset value of MDA is one. The value of MDA determines which operating mode is selected **after** reset. These two bits can be read at any time. They can be written anytime in test modes (SMOD = 1). MDA can only be written once in normal modes. SMOD cannot be set once it has been cleared.

Logic Level of MODB Pin at Reset	Value of SMOD Latched at Reset	Programmed Value of MDA	Mode Selected
1	0	1	Expanded
0	1	1	Special Test
1	0	0	Single Chip
0	1	0	Bootstrap

Bit 4 — Not implemented
Always reads zero

PSEL[3:0] — Priority Select Bits [3:0]
Refer to **6 Resets and Interrupts**.

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

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

Bits [7:5] — Not implemented
Always read zero

IRVNE — Internal Read Visibility/Not E
IRVNE can be written once in any mode. In expanded and special-test modes, IRVNE determines whether IRV is on or off. In special test mode, IRVNE is reset to one. In all other modes, IRVNE is reset to zero.
0 = No internal read visibility on external bus
1 = Data from internal reads is driven out the external data bus.
In single-chip and bootstrap 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
Expanded	0	On	Off	IRV	Once
Special Test	1	On	On	IRV	Once
Single Chip	0	On	Off	E	Once
Bootstrap	0	On	Off	E	Once

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

3 On-Chip Memory

The MC68HC11C0 has 256 bytes of RAM. A 1024-byte block of bootstrap ROM is present in single-chip, bootstrap, and test modes. The following paragraphs describe the memory systems of this MCU.

3.1 Memory Map and Register Block

The INIT register control the location of the registers in the 64-Kbyte CPU address space. The 128-byte register block originates at $0000 after reset and can be placed at any 1-Kbyte boundary by writing an appropriate value to the INIT register. The INIT register can be written only in the first 64 cycles after reset. If the register block and RAM are placed at the same 1-Kbyte boundary, the first 128 bytes of RAM are inaccessible. This is due to an on-chip hardware priority scheme which eliminates conflicts which could arise from multiple resources sharing address locations. Refer to the memory map diagram.

INIT — Register Mapping **$003D**

	Bit 7	6	5	4	3	2	1	Bit 0
	REG15	REG14	REG13	REG12	REG11	REG10	—	—
RESET:	0	0	0	0	0	0	0	0

Can be written only once in first 64 cycles in expanded and single-chip modes or at any time in other modes.

REG[15:10] — Internal Register Map Position

These bits determine the upper six bits of the register block address. At reset registers are mapped to $0000–$007F. Refer to the memory map diagram.

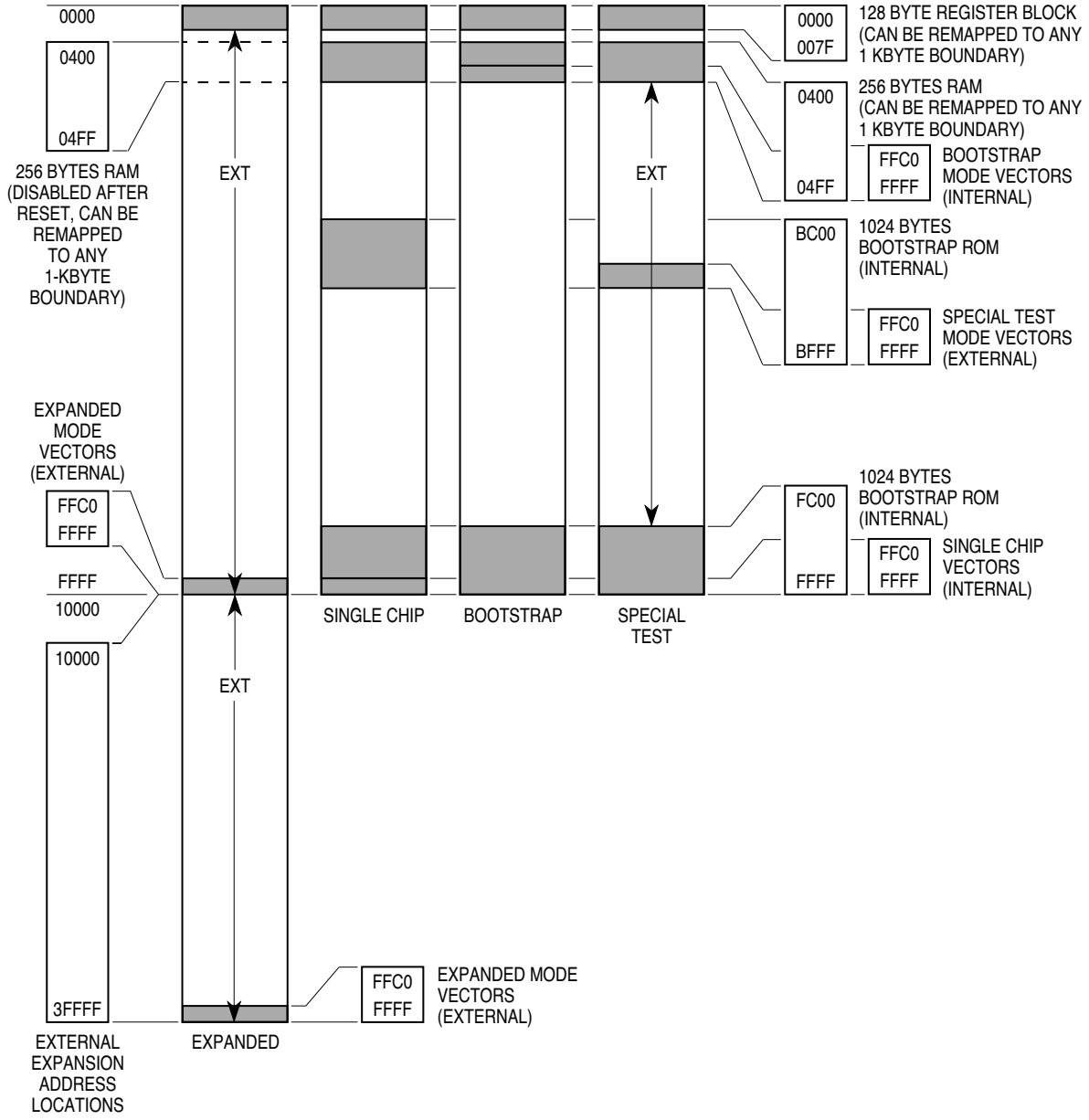


Figure 5 MC68HC11C0 Memory Map

Table 2 MC68HC11C0 Register and Control Bit Assignments

The register block begins at $0000 out of reset and can be remapped to any 1K 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	0	PF6	PF5	PF4	PF3	PF2	PF1	PF0	PORTF
$0003	0	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	DDRF
$0004	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0	FISTAT
$0005	0	IE6	IE5	IE4	IE3	IE2	IE1	IE0	FINTEN
$0006									Reserved
$0007	0	0	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0	DIOCTL
$0008	0	0	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
$0009	0	0	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
$000A	0	0	0	0	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)
$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

Table 2 MC68HC11C0 Register and Control Bit Assignments (Continued)

The register block begins at $0000 out of reset and can be remapped to any 1K boundary.

	Bit 7	6	5	4	3	2	1	Bit 0	
$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	0	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR
$0029	SPIF	WCOL	0	MODF	0	0	0	0	SPSR
$002A	Bit 7	6	5	4	3	2	1	Bit 0	SPDR
$002B	TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD
$002C	R8	T8	0	M	WAKE	0	0	0	SCCR1
$002D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2
$002E	TDRE	TC	RDRF	IDLE	OR	NF	FE	0	SCSR
$002F	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0	SCDR
$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									Reserved
$0036									Reserved
$0037	RAM15	RAM14	RAM13	RAM12	RAM11	RAM10	0	0	INIT2
$0038	0	0	0	IRVNE	0	0	0	0	OPT2
$0039	ADPU	CSEL	IRQE	DLY	CME	0	CR1	CR0	OPTION
$003A	Bit 7	6	5	4	3	2	1	Bit 0	COPRST
$003B									Reserved
$003C	RBOOT	SMOD	MDA	0	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO
$003D	REG15	REG14	REG13	REG12	REG11	REG10	0	0	INIT
$003E	TILOP	0	OCCR	CBYP	DISR	FCM	FCOP	TCON	TEST1
$003F	0	0	RWMC	0	0	NOCOP	RAMON	0	CONFIG
$0040	VA15	VA14	VA13	VA12	VA11	VA10	0	0	VCSADR
$0041									Reserved
$0042	PSA15	PSA14	PSA13	PSA12	PSA11	PSA10	PSTHA	PSTHB	PGSADR
$0043	PEA15	PEA14	PEA13	PEA12	PEA11	PEA10	0	0	PGEADR
$0044	0	0	0	0	0	0	XA17	XA16	MXHADR
$0045	XA15	XA14	XA13	XA12	XA11	XA10	0	0	MXLADR
$0046	GS1A15	GS1A14	GS1A13	GS1A12	GS1A11	GS1A10	G1STHA	G1STHB	GP1SADR
$0047	GE1A15	GE1A14	GE1A13	GE1A12	GE1A11	GE1A10	0	0	GP1EADR
$0048	GS2A15	GS2A14	GS2A13	GS2A12	GS2A11	GS2A10	G2STHA	G2STHB	GP2SADR
$0049	GE2A15	GE2A14	GE2A13	GE2A12	GE2A11	GE2A10	0	0	GP2EADR
$004A	GS3A15	GS3A14	GS3A13	GS3A12	GS3A11	GS3A10	G3STHA	G3STHB	GP3SADR
$004B	GE3A15	GE3A14	GE3A13	GE3A12	GE3A11	GE3A10	0	0	GP3EADR

Table 2 MC68HC11C0 Register and Control Bit Assignments (Continued)

The register block begins at $0000 out of reset and can be remapped to any 1K boundary.

	Bit 7	6	5	4	3	2	1	Bit 0	
$004C	GS4A15	GS4A14	GS4A13	GS4A12	GS4A11	GS4A10	G4STHA	G4STHB	GP4SADR
$004D	GE4A15	GE4A14	GE4A13	GE4A12	GE4A11	GE4A10	0	0	GP4EADR
$004E	GS5A15	GS5A14	GS5A13	GS5A12	GS5A11	GS5A10	G5STHA	G5STHB	GP5SADR
$004F	GE5A15	GE5A14	GE5A13	GE5A12	GE5A11	GE5A10	0	0	GP5EADR
$0050									Reserved
$0051									Reserved
$0052									Reserved
$0053									Reserved
$0054									Reserved
$0055									Reserved
$0056									Reserved
$0057									Reserved
$0058									Reserved
$0059									Reserved
$005A									Reserved
$005B									Reserved
$005C									Reserved
$005D									Reserved
$005E									Reserved
$005F									Reserved
$0060	0	CON12	0	0	0	PCKA3	PCKA2	PCKA1	PWCLK
$0061	0	0	PCLK2	PCLK1	0	0	PPOL2	PPOL1	PWPOL
$0062	Bit 7	6	5	4	3	2	1	Bit 0	PWSCAL
$0063	TPWSL	DISCP	0	0	0	0	PWEN2	PWEN1	PWEN
$0064									Reserved
$0065									Reserved
$0066	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT1
$0067	Bit 7	6	5	4	3	2	1	Bit 0	PWCNT2
$0068									Reserved
$0069									Reserved
$006A	Bit 7	6	5	4	3	2	1	Bit 0	PWPER1
$006B	Bit 7	6	5	4	3	2	1	Bit 0	PWPER2
$006C									Reserved
$006D									Reserved
$006E	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY1
$006F	Bit 7	6	5	4	3	2	1	Bit 0	PWDTY2
$0070	HPPUE	GPPUE	FPPUE	0	DPPUE	0	0	APPUE	PPAR
$0071	PGEN7	PGEN6	PGEN5	PGEN4	PGEN3	MEM1	MEM0	PGEN0	PGEN

Table 2 MC68HC11C0 Register and Control Bit Assignments (Continued)

The register block begins at $0000 out of reset and can be remapped to any 1K boundary.

	Bit 7	6	5	4	3	2	1	Bit 0	
$0072									Reserved
$0073									Reserved
$0074									Reserved
$0075	0	0	DOD5	DOD4	DOD3	DOD2	DOD1	DOD0	DODM
$0076									Reserved
$0077									Reserved
$0078									Reserved
$0079									Reserved
$007A									Reserved
$007B									Reserved
$007C	0	0	0	0	0	0	PH1	PH0	PORTH
$007D	0	0	0	0	0	0	DDH1	DDH0	DDRH
$007E	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0	PORTG
$007F	DDG7	DDG6	DDG5	DDG3	DDG3	DDG2	DDG1	DDG0	DDRG

3.2 RAM

In expanded mode, RAM is disabled after reset. In single-chip, bootstrap, and special test modes RAM is enabled and present at locations $0400–$04FF. The RAM can be mapped to any 1-Kbyte boundary by writing an appropriate value to the INIT2 register. The INIT2 register must be written during the first 64 cycles after reset in expanded and single-chip modes. If RAM and the register block are placed at the same 1-Kbyte boundary, the first 128 bytes of RAM are inaccessible. This is due to an on-chip hardware priority scheme which eliminates conflicts which could arise from multiple resources sharing address locations. Refer to the memory map diagram.

INIT2 — RAM Mapping **$0037**

	Bit 7	6	5	4	3	2	1	Bit 0
	RAM15	RAM14	RAM13	RAM12	RAM11	RAM10	—	—
RESET:	0	0	0	0	0	1	0	0

Can be written anytime in first 64 cycles in expanded or single-chip modes or at any time in other modes.

RAM[15:10] — Internal RAM Map Position

These bits determine the upper six bits of the RAM address. At reset RAM is mapped to $0400–$04FF. Refer to the memory map diagram.

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

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	RWMC	—	—	NOCOP	RAMON	—
RESET:	0	0	0	0	0	1	0	0

In single-chip and expanded modes (SMOD = 0), CONFIG can only be written once. In special test and bootstrap modes (SMOD = 1), CONFIG can be written any time. Changes do not take effect until the first cycle of the instruction following the write to CONFIG.

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

RWMC — Read/Write Mode Strobe Mode Control
Refer to **2 Operating Modes**.

Bits [4:3] — Not implemented
Always read zero

NOCOP — COP Watchdog Timer Disable
Refer to **6 Resets and Interrupts**.

RAMON — RAM Enable
In all modes except expanded mode, RAMON is forced to one out of reset.
0 = 256 bytes of RAM present in the memory map
1 = 256 bytes of RAM removed from the memory map and powered off

Bit 0 — Not implemented
Always reads zero

3.3 Bootstrap ROM

When operating in expanded mode, the bootstrap ROM is disabled and removed from the memory map. In bootstrap and special test modes, bootstrap ROM is present at $FC00–$FFFF. In single-chip mode the bootstrap ROM appears at locations $FC00–$FFFF **and** $BC00–$BFFF. Bootstrap ROM cannot be remapped to other locations.

The bootstrap ROM contains a small program that allows program code to be downloaded into on-chip RAM. When the MC68HC11C0 enters bootstrap mode, bootloader firmware residing in bootstrap ROM begins the downloading procedure by initializing the SCI system and transmitting a break out the SCI TxD pin. The SCI then waits for the first character to be received. After the first character is received on the RxD pin of the SCI, bootloader firmware begins counting the number of bytes received. When an idle time of four characters or the character for address $04FF is received, the bootloader program terminates the download and control is passed to the loaded program at $0400. For a detailed description of the M68HC11 bootstrap mode, refer to application note *M68HC11 Bootstrap Mode* (AN1060/D).

4 Memory Expansion and Chip Selects

The MC68HC11C0 has the ability to extend the addressing range of the M68HC11 CPU beyond the physical 64-Kbyte limit of the 16 CPU address lines. The extra addressing capability is provided by a register-based paging scheme using two expansion address lines and the physical 64 Kbytes of CPU address space.

Two additional on-chip blocks are necessary to support extended addressing. 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

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 256 Kbytes can be accessed through a paged memory scheme. Additional address lines XA[17:16] are provided to allow banks of expanded memory to be selected to appear in a specified bank window. XA[17:16] are implemented as alternate functions of port G pins PG[2:1]. Bits in the port G enable register (PGEN) define which port G pins are used for chip selects and memory expansion address lines and which are used for general-purpose I/O. Port G pull-ups are enabled out of reset in order to provide a logic level one on all chip select and memory expansion address lines.

The MEM[1:0] bits in PGEN select one of the three memory expansion modes. XA[17:10] in MXHADR/MXLADR contain the expansion address offset with respect to the current CPU address. When memory expansion is enabled, and the CPU address falls within the memory expansion window defined by values in PGxADR registers, $\overline{\text{CSPROG}}$ is activated and the CPU address ADDR[15:0] will be added to the value in MXHADR/MXLADR and possibly extended to include XA[17:16] before being driven out to the external device. XA[17:16] will be used only if addressing is extended beyond 64 Kbytes. If the CPU address falls outside the expansion window, ADDR[15:10] simply reflect the internal CPU address. ADDR[9:0] always reflect the internal CPU address. Refer to the Memory Expansion and Program Chip Select Block Diagram.

In 64-Kbyte mode with no expansion, addressing is limited to 64 Kbytes and ADDR[15:10] are used to decode the chip selects. Chip select granularity is 1 Kbyte. Memory expansion is disabled. The program/vector chip select is disabled. ADDR[15:10] reflect the internal CPU address signals.

In 64-Kbyte expansion mode, addressing is limited to 64 Kbytes and ADDR[15:10] are used to decode the chip selects. CPU address ADDR[15:10] are recalculated based on the value in MXHADR/MXLADR registers before being driven out on ADDR[15:10] pins. The program chip select is active if the CPU address falls within the chip select range defined by values in PGxADR registers. If the CPU address falls outside the chip select window, ADDR[15:10] remain unchanged and simply reflect the internal address signals. XA[17:16] are general-purpose I/O. ADDR[9:0] always reflect the CPU address signals.

In 128-Kbyte expansion mode, ADDR[15:10] are used to decode the chip selects. If the CPU address falls within the chip select range defined by values in PGxADR registers, it is activated and CPU address ADDR[15:10] are recalculated based on the value in MXHADR/MXLADR registers before being driven out on XA16 **and** ADDR[15:10] pins. If the CPU address falls outside the chip select window, ADDR[15:10] simply reflect the internal address signals. XA16 reflects the state of the XA16 bit in MXHADR. XA17 is general-purpose I/O. ADDR[9:0] always reflect the CPU address.

In 256-Kbyte expansion mode, ADDR[15:10] are used to decode the chip selects. If the CPU address falls within the chip select range defined by values in PGxADR registers, it is activated and CPU address ADDR[15:10] are recalculated based on the value in MXHADR/MXLADR registers before being driven out on XA[17:16] **and** ADDR[15:10] pins. If the CPU address falls outside the chip select window, ADDR[15:10] simply reflect the internal address signals. XA[17:16] reflect the state of the XA[17:16] bits in MXHADR/MXLADR. ADDR[9:0] always reflect the CPU address.

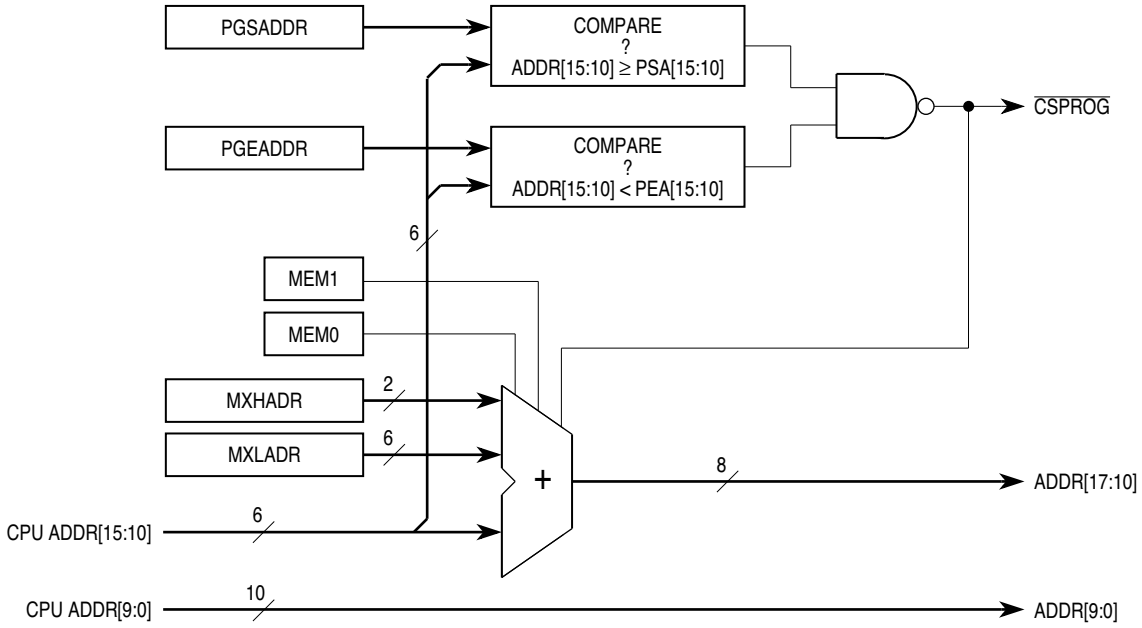


Figure 6 Memory Expansion and Program Chip Select Block Diagram

In the following example the user has constructed a system composed of the MC68HC11C0 MCU and a single 256-Kbyte external memory device. Since the user wishes to access all locations in the external memory, the 256-Kbyte expansion mode is chosen (MEM[1:0] = 1:1). The window in which the pages of expanded memory will appear is programmed to be located at $6000–$AFFF (PGSADR = $60, PGEADR = $B0). Note that this also defines the range of the program chip select. PGEN0 must be set in order for $\overline{\text{CSPROG}}$ to drive the external pin. Since the expansion window has been defined as 20 Kbytes in length, the 256-Kbyte external memory will be divided into 20-Kbyte segments (twelve 20-Kbyte pages and one 16-Kbyte page). The vector chip select has been programmed to begin at $D000. If the CPU address falls within the vector chip select range the vector chip select (CSV) is activated and XA[17:16] are forced high to ensure that the very top of the external memory is accessed. Refer to the Memory Expansion Address Translation Example diagram.

In this example the user will access the 20-Kbyte block in the external memory starting at external address $0A000. The following are the corresponding parameters involved in the translation:

Desired Starting Expansion Page Address — MA[17:10] = $0A0 = %00 1010 00
Desired Starting Window Address — PSA[15:10] = $60 = %0110 00

In general, the formula for the value required in MXHADR/MXLADR is:

XA[17:10] = MA[17:10] –PSA[15:10]

The equation for this example translation is:

XA[17:10] = $0A0 –$60

Since the translation is performed as a two's complement operation the following calculation is performed:

17	16	15			12	11			8	7			4	3			0		
0	0	1	0	1	0	0	0	x	x	x	x	x	x	x	x	x	x	MA[17:10]	$0A0
x	x	0	1	1	0	0	0	x	x	x	x	x	x	x	x	x	x	–PSA[15:10]	–$60
0	**0**	**0**	**1**	**0**	**0**	**0**	**0**											**XA[17:10]**	**$040**

Thus, MXHADR/MXLADR must contain the value $040 to cause the external 20-Kbyte page beginning at $0A000 to appear in the memory expansion window.

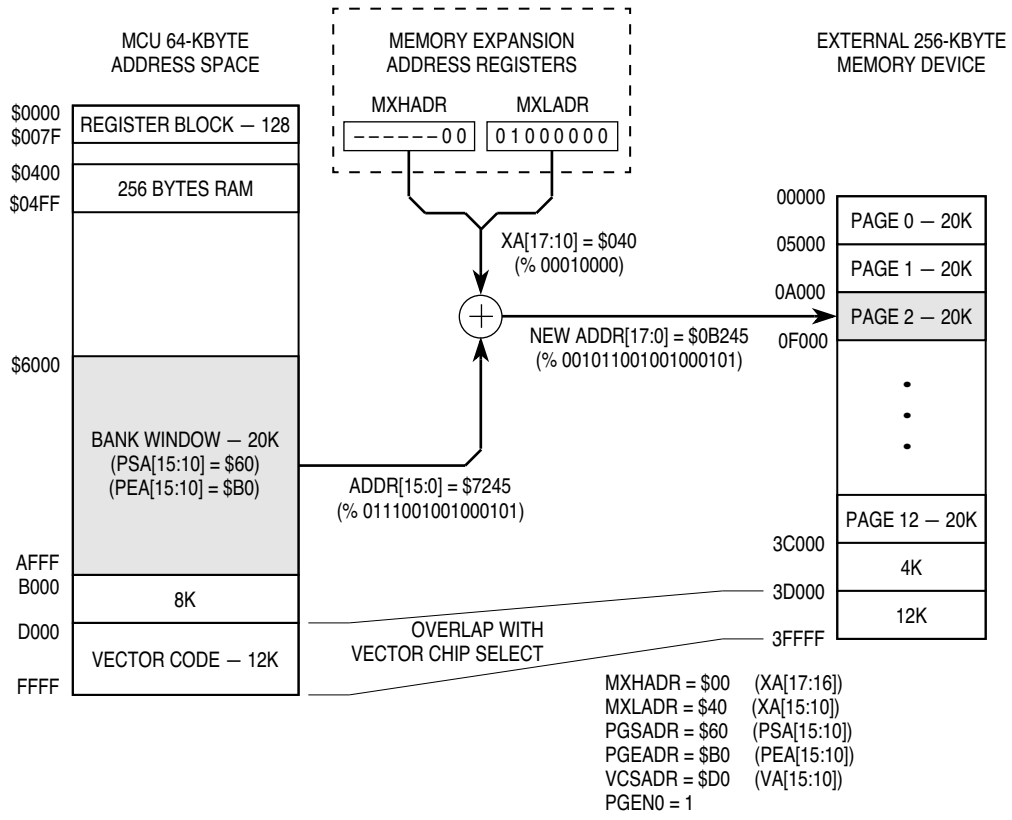


Figure 7 Memory Expansion Address Translation Example (256K Mode)

Although there are three expansion modes, the manner in which the expansion address is defined is identical. Each of the memory expansion modes has a slightly different formula for calculating the expansion address. They are defined as follows:

64K Expansion Mode:

XA[15:10] = MA[15:10] – PSA[15:10]; XA[17:16] are not used

128K Expansion Mode:

XA[16:10] = MA[16:10] – PSA[15:10]; XA17 is not used

256K Expansion Mode:

$$XA[17:10] = MA[17:10] – PSA[15:10]$$

where,

XA = expansion address
MA = most significant bits of the starting address of the active page of expanded memory
PSA = bits in PGSADR register

PGEN — Port G Enable **$0071**

	Bit 7	6	5	4	3	2	1	Bit 0
	PGEN7	PGEN6	PGEN5	PGEN4	PGEN3	MEM1	MEM0	PGEN0
RESET:	0	0	0	0	0	0	0	0

PGEN[7:3] — Port G Enable Bits [7:3]
0 = Corresponding port G pin configured for general-purpose I/O.
1 = Corresponding port G pin configured for general-purpose chip select output.

MEM[1:0] — Memory Expansion Mode Select Bits

MEM1	MEM0	Expansion Mode	PG2	PG1
0	0	64-Kbyte CPU Address, No Expansion	I/O	I/O
0	1	64-Kbyte Expansion	I/O	I/O
1	0	128-Kbyte Expansion	I/O	XA16
1	1	256-Kbyte Expansion	XA17	XA16

PGEN0 — Port G Enable Bits 0
0 = PG0 configured for general-purpose I/O.
1 = PG0 configured for vector/program chip select output.

MXHADR — Memory Expansion Address High **$0044**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	—	—	—	—	XA17	XA16
RESET:	0	0	0	0	0	0	0	0

XA[17:16] — Memory Expansion Address [17:16]
Refer to Memory Expansion and Program Chip Select Block Diagram.

MXLADR — Memory Expansion Address Low **$0045**

	Bit 7	6	5	4	3	2	1	Bit 0
	XA15	XA14	XA13	XA12	XA11	XA10	—	—
RESET:	0	0	0	0	0	0	0	0

XA[15:10] — Memory Expansion Address [15:10]
Refer to Memory Expansion and Program Chip Select Block Diagram.

4.2 Chip Selects

Seven chip select signals are provided to simplify the interface to external components. Five general-purpose and one program/vector chip select pin are implemented as alternate functions of port G pins. Port G pull-ups are enabled out of reset in order to provide a logic level one on all chip select and memory expansion address lines. The chip selects are designed to operate with or without memory expansion. All chip selects are prioritized so that they never conflict with each other or with on-chip resources.

General-purpose chip selects are automatically activated (if enabled) whenever the current CPU address falls within a range defined by the associated control registers for each chip select. All general-purpose chip selects use the same format for selecting starting address, ending address, and clock stretch. Each of the five general-purpose chip selects has two control registers associated with it. The first register (GPxSADR) selects the upper six MSB of the starting address and selects the clock stretch. The second register (GPxEADR) selects the upper six MSB of the ending address. Since these bits are the upper six MSB, the granularity of each chip select range is 1024Êbytes. Each general-purpose chip select is an active-low signal with a programmable clock stretch from zero to three E-clock cycles. Bits in the PGEN register enable each of the five general-purpose chip selects. When a chip select is enabled, the corresponding port G pin is forced to be an output regardless of the state of the DDGx bit.

The program chip select ($\overline{\text{CSPROG}}$) simplifies the interface to external devices and functions with or without memory expansion. CPU address lines ADDR[15:10] are always used to decode the program chip select; therefore, its granularity is fixed at 1 Kbyte. The range is defined by bits in PGSADR and PGEADR. When the CPU address falls within the defined range, $\overline{\text{CSPROG}}$ is asserted. If memory expansion is enabled, the range of the program chip select corresponds to the memory expansion window. In this case, $\overline{\text{CSPROG}}$ will be asserted and the current CPU address will become modified according to the contents of the memory expansion address registers MXHADR and MXLADR before being driven out to the external device. Refer to **4.1 Memory Expansion** for more information.

The vector chip select (CSV) is provided for the vector space and, because there is no internal memory at the reset vector address, is enabled for the entire address space out of reset in expanded mode. VCSADR selects the upper six MSB of the starting address. Since these bits are the upper six MSB, the granularity of the vector chip select range is 1024 bytes. The ending address is the highest address ($FFFF). The vector chip select is an active-low signal with a programmable clock stretch from zero to three E-clock cycles. Bits in the PGEN register enable each of the five general-purpose chip selects. Whenever the CPU logical address falls within the range defined by VCSADR, CSV is asserted and the current CPU logical address is driven out ADDR[15:0] to the external memory device. XA[17:16] (if enabled) are always driven high when vector space is selected to ensure that the vector space is always located at the top of the address space. CSV is configured for one cycle of clock stretch out of reset in expanded mode. This can be altered by changing the values in PSTHA and PSTHB in PGSADR register. When CSV is enabled, PG0 is forced to be an output regardless of the state of the DDG0 bit.

CAUTION

If program code is contained in an external memory, the range for $\overline{\text{CSPROG}}$ must be defined before the vector chip select range is changed. This prevents the program from being lost at the point when CSV is changed.

The range of the program chip select is defined as follows:

$$PSA[15:10] \leq ADDR[15:10] < PEA[15:10]$$

The range of the vector chip select is defined as follows:

$$VSA[15:10] \leq ADDR[15:10] \leq \$3FFF$$

where,

PSA = bits in PGSADR register
PEA = bits in PGEADR register
VSA = bits in VCSADR register
ADDR = CPU logical address

Chip selects are arranged in the following priority:

Priority	Resource
Highest	On-Chip Registers
•	On-Chip Boot ROM (if enabled)
•	On-Chip RAM (if enabled)
•	Vector Chip Select (CSV)
•	Program Chip Select ($\overline{\text{CSPROG}}$)
•	General-Purpose Chip Select 1 (CSGP1)
•	General-Purpose Chip Select 2 (CSGP2)
•	General-Purpose Chip Select 3 (CSGP3)
•	General-Purpose Chip Select 4 (CSGP4)
Lowest	General-Purpose Chip Select 5 (CSGP5)

VCSADR — Vector Chip Select Base Address **$0040**

	Bit 7	6	5	4	3	2	1	Bit 0
	VA15	VA14	VA13	VA12	VA11	VA10	—	—
RESET:	0	0	0	0	0	0	0	0

VA[15:10] — Vector Chip Select Address
Selects the MSB of the vector chip select starting address.

PGSADR — Program Chip Select Starting Address **$0042**

	Bit 7	6	5	4	3	2	1	Bit 0	
	PSA15	PSA14	PSA13	PSA12	PSA11	PSA10	PSTHA	PSTHB	
RESET:	0	0	0	0	0	0	0	1	Expanded Mode
	0	0	0	0	0	0	0	0	Test Mode

PSA[15:10] — Program Chip Select Starting Address

PSTH[A:B] — Program/Vector Chip Select Clock Stretch Select

PSTHA	PSTHB	Clock Stretch
0	0	None
0	1	1 E-Clock Cycle
1	0	2 E-Clock Cycles
1	1	3 E-Clock Cycles

PGEADR — Program Chip Select Ending Address **$0043**

	Bit 7	6	5	4	3	2	1	Bit 0
	PEA15	PEA14	PEA13	PEA12	PEA11	PEA10	—	—
RESET:	0	0	0	0	0	0	0	0

PEA[15:10] — Program Chip Select Ending Address

GP1SADR — General-Purpose Chip Select 1 Starting Address **$0046**

	Bit 7	6	5	4	3	2	1	Bit 0
	GS1A15	GS1A14	GS1A13	GS1A12	GS1A11	GS1A10	GS1THA	GS1THB
RESET:	0	0	0	0	0	0	0	0

GS1A[15:10] — Program Chip Select Starting Address

G1STH[A:B] — Program Chip Select Clock Stretch Select

G1STHA	G1STHB	Clock Stretch
0	0	None
0	1	1 E-Clock Cycle
1	0	2 E-Clock Cycles
1	1	3 E-Clock Cycles

GP1EADR — General-Purpose Chip Select 1 Ending Address **$0047**

	Bit 7	6	5	4	3	2	1	Bit 0
	GE1A15	GE1A14	GE1A13	GE1A12	GE1A11	GE1A10	—	—
RESET:	0	0	0	0	0	0	0	0

GE1A[15:10] — Program Chip Select Ending Address

GP2SADR — General-Purpose Chip Select 2 Starting Address **$0048**

	Bit 7	6	5	4	3	2	1	Bit 0
	GS2A15	GS2A14	GS2A13	GS2A12	GS2A11	GS2A10	GS2THA	GS2THB
RESET:	0	0	0	0	0	0	0	0

GS2A[15:10] — Program Chip Select Starting Address

G2STH[A:B] — Program Chip Select Clock Stretch Select

G2STHA	G2STHB	Clock Stretch
0	0	None
0	1	1 E-Clock Cycle
1	0	2 E-Clock Cycles
1	1	3 E-Clock Cycles

GP2EADR — General-Purpose Chip Select 2 Ending Address **$0049**

	Bit 7	6	5	4	3	2	1	Bit 0
	GE2A15	GE2A14	GE2A13	GE2A12	GE2A11	GE2A10	—	—
RESET:	0	0	0	0	0	0	0	0

GE2A[15:10] — Program Chip Select Ending Address

GP3SADR — General-Purpose Chip Select 3 Starting Address **$004A**

	Bit 7	6	5	4	3	2	1	Bit 0
	GS3A15	GS3A14	GS3A13	GS3A12	GS3A11	GS3A10	GS3THA	GS3THB
RESET:	0	0	0	0	0	0	0	0

GS3A[15:10] — Program Chip Select Starting Address

G3STH[A:B] — Program Chip Select Clock Stretch Select

G3STHA	G3STHB	Clock Stretch
0	0	None
0	1	1 E-Clock Cycle
1	0	2 E-Clock Cycles
1	1	3 E-Clock Cycles

GP3EADR — General-Purpose Chip Select 3 Ending Address **$004B**

	Bit 7	6	5	4	3	2	1	Bit 0
	GE3A15	GE3A14	GE3A13	GE3A12	GE3A11	GE3A10	—	—
RESET:	0	0	0	0	0	0	0	0

GE3A[15:10] — Program Chip Select Ending Address

GP4SADR — General-Purpose Chip Select 4 Starting Address **$004C**

	Bit 7	6	5	4	3	2	1	Bit 0
	GE3A15	GE3A14	GE3A13	GE3A12	GE3A11	GE3A10	—	—
RESET:	0	0	0	0	0	0	0	0

GS4A[15:10] — Program Chip Select Starting Address

G4STH[A:B] — Program Chip Select Clock Stretch Select

G4STHA	G4STHB	Clock Stretch
0	0	None
0	1	1 E-Clock Cycle
1	0	2 E-Clock Cycles
1	1	3 E-Clock Cycles

GP4EADR — General-Purpose Chip Select 4 Ending Address **$004D**

	Bit 7	6	5	4	3	2	1	Bit 0
	GE4A15	GE4A14	GE4A13	GE4A12	GE4A11	GE4A10	—	—
RESET:	0	0	0	0	0	0	0	0

GE4A[15:10] — Program Chip Select Ending Address

GP5SADR — General-Purpose Chip Select 5 Starting Address **$004E**

	Bit 7	6	5	4	3	2	1	Bit 0
	GS5A15	GS5A14	GS5A13	GS5A12	GS5A11	GS5A10	GS5THA	GS5THB
RESET:	0	0	0	0	0	0	0	0

GS5A[15:10] — Program Chip Select Starting Address

G5STH[A:B] — Program Chip Select Clock Stretch Select

G5STHA	G5STHB	Clock Stretch
0	0	None
0	1	1 E-Clock Cycle
1	0	2 E-Clock Cycles
1	1	3 E-Clock Cycles

GP5EADR — General-Purpose Chip Select 5 Ending Address **$004F**

	Bit 7	6	5	4	3	2	1	Bit 0
	GE5A15	GE5A14	GE5A13	GE5A12	GE5A11	GE5A10	—	—
RESET:	0	0	0	0	0	0	0	0

GE5A[15:10] — Program Chip Select Ending Address

5 Parallel Input/Output

The MC68HC11C0 has up to 35 input/output lines, depending on the operating mode. The address and data bus have no associated ports and cannot be used for general-purpose I/O.

Pins on all ports except port E have selectable on-chip pull-up devices. A single bit in the port pull-up assignment register (PPAR) enables the pull-up devices for all pins on the associated port. A pin's pull-up device is active only when the associated data direction bit configures that pin as an input. Pull-ups for $\overline{IRQ}$ and $\overline{XIRQ}$ are active whenever port D pull-ups are enabled. Port G pull-ups are enabled out of reset in order to provide a logic level one on all chip select and memory expansion address lines. Port F pull-ups are also enabled out of reset and are active only when a port F pin is configured as an input. Refer to the PPAR register description.

Port A has eight fully bidirectional I/O pins. Port A shares functions with the timer system. Note that when PA7 is configured as an output (DDA7 = 1) it is still the input to the pulse accumulator (PAI).

Port D shares functions with the serial systems (SCI and SPI). Because $\overline{IRQ}$ and $\overline{XIRQ}$ are now associated functions of port D, a new port D I/O control register (DIOCTL) has been added. Port D functions are controlled by a combination of DIOCTL bits, data direction bits, SCI and SPI enable bits. Refer to the PORTD description. Port D has six bidirectional pins and one output-only pin.

Port E is a four-bit input-only port that shares functions with the A/D converter system. If the A/D system is not being used, port F pins can be used as general-purpose inputs.

Port F has seven bidirectional pins and shares functions with the keyboard interrupt inputs. Port F pins not used for interrupt request inputs can be used for general-purpose I/O.

Port G is an eight-bit fully bidirectional I/O port. Port G shares functions with the memory expansion address lines and the chip selects. Pins not used for memory expansion address or chip select can be used for general-purpose I/O.

Port H is a two-bit bidirectional port. Port H pins also serve as outputs for the two-channel PWM timer.

The following table is a summary of the configuration and features of each port.

Port	Input Pins	Output Pins	Bidirectional Pins	On-Chip Pull-Up Devices	Shared Functions
Port A	—	—	8	Yes	Timer
Port D	—	1	6	Yes	SCI, SPI, $\overline{IRQ}$, and $\overline{XIRQ}$
Port E	4	—	—	No	A/D Converter
Port F	—	—	7	Yes	Keyboard Interrupt Requests
Port G	—	—	8	Yes	Memory Expansion Address and Chip Selects
Port H	—	—	2	Yes	PWM Timer

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	OC5/IC4	IC1	IC2	IC3
And/or:	OC1	OC1	OC1	OC1	OC1	—	—	—

The timer forces the I/O state to output for each port A line associated with an enabled output compare. In these cases the data direction bits will not be changed, but have no effect on these lines. The DDRA will revert to controlling data direction when the associated timer compare is disabled. Input captures do not force the I/O state of the pin or the state of DDRA.

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

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	SDO/ MOSI	SDI/ MISO	TxD	RxD
or:	—	—	$\overline{IRQ}$	—	—	$\overline{XIRQ}$	—	—

After reset PD[5:0] are configured as high impedance inputs. PD[5:0] share functions with the SCI, SPI, and two interrupt request lines ($\overline{IRQ}$ and $\overline{XIRQ}$). The actual function performed by each pin depends on bits in DIOCTL, SCI/SPI enable bits, and bits in the DDRD register. Refer to the tables located in the DIOCTL register description.

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 = Input
1 = Output

DIOCTL — Port D I/O Control **$0007**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	DIO5	DIO4	DIO3	DIO2	—	DIO0
RESET:	0	0	1	1	1	1	0	0

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

DIO[5:2] — Port D I/O Control for Port D Bits [5:2]
Refer to the following tables for description.

Bit 1 — Not implemented
Always reads zero.

DIO0 — Port D I/O Control for Port D Bit 0
Refer to the following tables for description.

Table 3 PD5 Configuration

SPI Enable	DIOCTL Bit 5	DIOCTL Bit 4	PD5	Pull-Up Control	Output Enable
SPE = 0	0	X	I/O	DDD5	DDD5
	1	0	$\overline{IRQ}$	On	Off
	1	1	I/O	DDD5	DDD5
SPE = 1	0	X	I/O	DDD5	DDD5
	1	0	$\overline{IRQ}$	On	Off
	1	1	$\overline{SS}$	Off	MSTR + DDD5

Notes:
1. If a pin is configured for general-purpose I/O, the DDRD bit controls the direction of data.
2. If a pin is configured as $\overline{SS}$, the MSTR and DDRD bits are the output enable.
3. SPE is the SPI enable bit, MSTR is the master/slave select bit. Refer to the SPCR register.
4. $\overline{IRQ}$ or $\overline{SS}$ inputs are internally pulled high if not used. This does not affect the pin.

Table 4 PD[4:3] Configuration

SPI Enable	PD4	PD3	Pull-Up Control	Output Enable
SPE = 0	I/O	I/O	DDD[4:3]	DDD[4:3]
SPE = 1	MOSI	SCK	Off	MSTR + DDD[4:3]

Notes:
1. If a pin is configured for general-purpose I/O, the DDRD bit controls the direction of data.
2. If a pin is configured as MOSI, the MSTR and DDRD bit control the direction of data.
3. SPE is the SPI enable bit, MSTR is the master/slave select bit. Refer to the SPCR register.

Table 5 PD2 Configuration

SPI Enable	DIOCTL Bit 3	DIOCTL Bit 2	PD2	Pull-Up Control	Output Enable
SPE = 0	0	X	I/O	DDD[3:2]	DDD[3:2]
	1	0	$\overline{\text{XIRQ}}$	On	Off
	1	1	I/O	DDD[3:2]	DDD[3:2]
SPE = 1	0	X	I/O	DDD[3:2]	DDD[3:2]
	1	0	$\overline{\text{XIRQ}}$	On	Off
	1	1	MISO	Off	MSTR + DDD[3:2]

Notes:
1. If a pin is configured for general-purpose I/O, the DDRD bit controls the direction of data.
2. If a pin is configured as MISO, the MSTR and DDRD bit control the direction of data.
3. SPE is the SPI enable bit, MSTR is the master/slave select bit. Refer to the SPCR register.
4. $\overline{\text{XIRQ}}$ or MISO inputs are internally pulled high if not used. This does not affect the pin.

Table 6 PD[1:0] Configuration

DIOCTL Bit 0	SCI Enables	PD1	Pull-Up Control	SCI Enables	PD0	Pull-Up Control	SCI Mode
0	TE = 1	TxD	Off	RE = 1	RxD	Off	Two Wire
	TE = 0	I/O	DDD1	RE = 0	I/O	DDD0	
1	TE = 1	TxD	Off	X	I/O	DDD0	Single Wire
	RE = 0						
	TE = 1	RxD	Off				
	RE = 0						
	TE = 1	RxD/TxD	Off				
	RE = 0	Looped					
	TE = 1	I/O	DDD1				
	RE = 0						

Notes:
1. If a pin is configured for general-purpose I/O, the DDRD bit controls the direction of data.
2. If a pin is configured for RxD input, there is no weak pull-up at the pin.
3. If a pin is configured for TxD output, the output can be either an open-drain output or a CMOS driver. This is controlled by the port D driver output mode (DODM) register.
4. TE is the transmitter enable bit. RE is the receiver enable bit. Refer to the SCCR2 register.

NOTE

If any of the pins PD[5:2] are configured as SPI inputs they will not have pull-ups. If any of the pins PD[5:2] are configured as SPI outputs they will be either open-drain outputs or normal CMOS driver outputs depending on the state of the corresponding bit in the DODM register.

DODM — Port D Open Drain Mode **$0075**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	DOD6	DOD5	DOD4	DOD3	DOD2	DOD1	DOD0
RESET:	0	0	1	1	1	1	0	0

Each DODM bit controls an individual port D pin and is valid only if the pin is configured as an output.

DOD[6:0] — Port D Open Drain Bits [6:0]

0 = Corresponding port D output pin configured as normal CMOS driver.

1 = Corresponding port D output pin configured as open-drain output driver.

PORTE — Port E Data **$000A**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	—	—	PE3	PE2	PE1	PE0
RESET:	0	0	0	0	I	I	I	I
Alt. Pin Func.:	—	—	—	—	AN3	AN2	AN1	AN0

Port E has four general-purpose input pins and shares functions with the A/D converter system. When any port E pins are being used as A/D inputs, PORTE should not be read during the sample portion of an A/D conversion. Refer to **11 Analog-to-Digital Converter**.

PORTF — Port F Data Register **$0002**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	PF6	PF5	PF4	PF3	PF2	PF1	PF0
RESET:	0	0	0	0	0	0	0	0
Alt. Pin Func.:	—	$\overline{\text{IRQ7}}$	$\overline{\text{IRQ6}}$	$\overline{\text{IRQ5}}$	$\overline{\text{IRQ4}}$	$\overline{\text{IRQ3}}$	$\overline{\text{IRQ2}}$	$\overline{\text{IRQ1}}$

Port F has seven bidirectional I/O lines. Each line can be either general-purpose I/O or a maskable interrupt source. When corresponding bits in FINTEN are set, port F lines become interrupt request inputs. Writes to PORTF drive pins only if the pins are configured for output and corresponding $\overline{\text{IRQ}}$ is disabled. Refer to FINTEN and FISTAT registers. Refer to **6 Resets and Interrupts**.

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

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

Bit 7 — Not implemented

Always reads zero.

DDF[6:0] — Data Direction for Port F

Overridden if corresponding interrupt request input is enabled.

0 = Input

1 = Output

FISTAT — Port F Interrupt Status **$0004**

	Bit 7	6	5	4	3	2	1	Bit 0
	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
RESET:	0	0	0	0	0	0	0	0

IS7 is the interrupt status bit for $\overline{\text{IRQ}}$ in port D. FISTAT can be read any time but cannot be written. All bits are cleared after CPU has read the register following an interrupt request.

IS7 — $\overline{\text{IRQ}}$ Status
0 = No interrupt pending for the corresponding interrupt line
1 = Interrupt pending for the corresponding interrupt line

IS[6:0] — $\overline{\text{IRQ}}$[6:0] Status
0 = No interrupt pending for the corresponding interrupt line
1 = Interrupt pending for the corresponding interrupt line

FINTEN — Port F Interrupt Enable **$0005**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	IE6	IE5	IE4	IE3	IE2	IE1	IE0
RESET:	0	0	0	0	0	0	0	0

The enable bit for $\overline{\text{IRQ}}$ is located in the DIOCTL register.

IE[6:0] — $\overline{\text{IRQx}}$ Enable
0 = Interrupt request input is disabled and pin is controlled by DDRF bit
1 = Interrupt request input (IRQx) is enabled, overriding state of DDRF bit

PORTG — Port G Data Register **$007E**

	Bit 7	6	5	4	3	2	1	Bit 0
	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
RESET:	0	0	0	0	0	0	0	0
Alt. Pin Func.:	GPCS5	GPCS4	GPCS3	GPCS2	GPCS1	XA17	XA16	CSV/ CSPROG

Port G is a fully bidirectional 8-bit port. Port G shares functions with the memory expansion address lines and the external chip selects. Refer to **4 Memory Expansion and Chip Selects**.

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

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

Bit 7 — Not implemented
Always reads zero.

DDG[6:0] — Data Direction for Port G
Overridden if corresponding interrupt request input is enabled.
0 = Input
1 = Output

PGEN — Port G Enable **$0071**

	Bit 7	6	5	4	3	2	1	Bit 0
	PGEN7	PGEN6	PGEN5	PGEN4	PGEN3	MEM1	MEM0	PGEN0
RESET:	0	0	0	0	0	0	0	1

PGEN[7:3], PGEN0 — Port G Enable Bits [7:3] and 0

PGENx must be set to output the corresponding chip select signal. PGENx overrides DDGx.

0 = Corresponding port G pin configured for output

1 = Corresponding port G pin configured for chip select function

MEM[1:0] — Memory Expansion Mode Select

MEM[1:0] select the memory expansion mode. Refer to the following table.

MEM1	MEM0	Memory Expansion Mode	PG1	PG2
0	0	64 Kbyte CPU Address, No Expansion	I/O	I/O
0	1	64 Kbyte CPU Address, 64 Kbyte Expansion	I/O	I/O
1	0	64 Kbyte CPU Address, 128 Kbyte Expansion	ADDR16	I/O
1	1	64 Kbyte CPU Address, 256 Kbyte Expansion	ADDR16	ADDR17

PORTH — Port H Data Register **$007C**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	—	—	—	—	PH1	PH0
RESET:	0	0	0	0	0	0	0	0
Alt. Pin Func.:	—	—	—	—	—	—	PW2	PW1

Port H has two bidirectional lines and shares functions with the PWM timer system. When a PWM timer channel is enabled the corresponding port H pin becomes an output regardless of the state of the DDHx bit. Refer to **9 Pulse-Width Modulation Timer**.

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

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	—	—	—	—	DDH1	DDH0
RESET:	0	0	0	0	0	0	0	0

Bits [7:2] — Not implemented

Always read zero.

DDH[1:0] — Data Direction for Port H

0 = Input

1 = Output

PPAR — Port Pull-Up Assignment Register **$0070**

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

PPAR can be read and written at any time. Pull-ups for port F interrupt request lines ($\overline{\text{IRQ}}$[6:0]) are enabled out of reset and are active only for port F pins configured as inputs. Port F pull-up devices are **not** automatically activated when the associated interrupt request line is enabled. Pull-up devices for other ports are automatically activated when pull-ups for that port are enabled and the associated pin is configured as an input.

xPPUE — Port x Pull-Up Enable

0 = Pull-up devices for port x disabled
1 = Pull-up devices for port x enabled

Bit 4 and Bits [2:1] — Not implemented

Always reads zero.

6 Resets and Interrupts

The MC68HC11C0 has 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 30 interrupt sources (three non-maskable, 27 maskable). The three non-maskable interrupt vectors 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 the I bit in the CCR is set.

Twenty-seven interrupt sources in the MC68HC11C0 are subject to masking by a global interrupt mask bit (I bit in the CCR). In addition to the global I bit, all of these sources are controlled by local enable bits in control registers. Most interrupt sources in M68HC11 devices have separate interrupt vectors; therefore, it is not usually necessary for software to poll control registers to determine the cause of an interrupt. In the case of the keyboard interrupt inputs, software must poll the port F interrupt status register (FISTAT) immediately following an interrupt to determine its source.

For some interrupt sources, such as the SCI and keyboard interrupts, 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 an automatic clearing mechanism consisting of a read of the SCI status register while RDRF is set, followed by a read of the SCI data register. The normal response to an RDRF interrupt request would be to read the SCI status register to check for receive errors, then to read the received data from the SCI data register. These two steps satisfy the automatic clearing mechanism without requiring any special instructions. Similarly, port F interrupt status register (FISTAT) is cleared when the CPU reads the register to determine which input was the source of the interrupt.

Refer to the following table for 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		23
	• SCI Receive Data Register Full		RIE	
	• SCI Receiver Overrun		RIE	
	• SCI Transmit Data Register Empty		TIE	
	• SCI Transmit Complete		TCIE	
	• SCI Idle Line Detect		ILIE	
FFD8, D9	SPI Serial Transfer Complete	I	SPIE	22
FFDA, DB	Pulse Accumulator Input Edge	I	PAII	21
FFDC, DD	Pulse Accumulator Overflow	I	PAOVI	20
FFDE, DF	Timer Overflow	I	TOI	19
FFE0, E1	Timer Input Capture 4/Output Compare 5	I	I4/O5I	17
FFE2, E3	Timer Output Compare 4	I	OC4I	14
FFE4, E5	Timer Output Compare 3	I	OC3I	13
FFE6, E7	Timer Output Compare 2	I	OC2I	12
FFE8, E9	Timer Output Compare 1	I	OC1I	11
FFEA, EB	Timer Input Capture 3	I	IC3I	10
FFEC, ED	Timer Input Capture 2	I	IC2I	9
FFEE, EF	Timer Input Capture 1	I	IC1I	8
FFF0, F1	Real Time Interrupt	I	RTII	7
FFF2, F3	Parallel I/O Handshake	I	None	6
	$\overline{\text{IRQ}}$	I	None	5
	$\overline{\text{IRQ}}$[6:0]	I	IE[6:0]	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	—	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 mode, or at any time in special modes.

ADPU — Analog-to-Digital Converter Power Up
Refer to **11 Analog-to-Digital Converter**.

CSEL — Clock Select
Refer to **11 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

Bit 2 — Not implemented
Always reads zero

CR[1:0] — COP Timer Rate Select
Refer to the following table of COP timer rates.

Table 7 COP Timer Rate Select

CR[1:0]	Divide E By	XTAL = 4.0 MHz Time-out –0 ms, +32.8 ms	XTAL = 8.0 MHz Time-out –0 ms, +16.4 ms	XTAL = 12.0 MHz Time-out –0 ms, +10.9 ms
0 0	2^{15}	32.768 ms	16.384 ms	10.923 ms
0 1	2^{17}	131.072 ms	65.536 ms	43.691 ms
1 0	2^{19}	524.288 ms	262.140 ms	174.76 ms
1 1	2^{21}	2.097 s	1.049 s	699.05 ms
	E =	1.0 MHz	2.0 MHz	3.0 MHz

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

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

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

NOSEC — COP System Disable
Refer to **3 On-Chip Memory**.

NOCOP — COP System Disable
Resets to programmed value
0 = COP enabled (forces reset on time-out)
1 = COP disabled (does not force reset on time-out)

ROMON — ROM/EPROM Enable
Refer to **3 On-Chip Memory**.

EEON — COP System Disable
Refer to **3 On-Chip Memory**.

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

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

*RBOOT, SMOD, and MDA reset depend on power-up initialization 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**.

Bit 4 — Not implemented
Always reads zero.

PSEL[3:0] — Priority Select Bits [3:0]
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**
3	**2**	**1**	**0**	
0	0	0	0	Timer Overflow
0	0	0	1	Pulse Accumulator Overflow
0	0	1	0	Pulse Accumulator Input Edge
0	0	1	1	SPI Serial Transfer Complete
0	1	0	0	SCI Serial System
0	1	0	1	Reserved (Default to $\overline{\text{IRQ}}$)
0	1	1	0	$\overline{\text{IRQ}}$ and $\overline{\text{IRQ[6:0]}}$
0	1	1	1	Real-Time Interrupt
1	0	0	0	Timer Input Capture 1
1	0	0	1	Timer Input Capture 2
1	0	1	0	Timer Input Capture 3
1	0	1	1	Timer Output Compare 1
1	1	0	0	Timer Output Compare 2
1	1	0	1	Timer Output Compare 3
1	1	1	0	Timer Output Compare 4
1	1	1	1	Timer Output Compare 5/Input Capture 4

6.1 External Interrupt Requests

The MC68HC11C0 has a total of nine external interrupt inputs. In addition to the two external interrupts found on other M68HC11 devices ($\overline{IRQ}$ and $\overline{XIRQ}$), seven more inputs for interrupt requests have been added. The seven additional inputs have been implemented as alternate functions of port F pins. The interrupt request inputs $\overline{IRQ}[6:0]$ are individually enabled by bits in the FINTEN register. The $\overline{IRQ}$ and $\overline{XIRQ}$ inputs have been moved and are now alternate functions of port D.

The $\overline{IRQ}$ and $\overline{IRQ}[6:0]$ interrupt sources are maskable and share the same priority. These interrupt sources can be masked globally by bit I in the condition code register as well as locally by enable bits in control registers. $\overline{IRQ}[6:0]$ are enabled by bits in the FINTEN register. $\overline{IRQ}$ is enabled by bits in DIOCTL register. Since $\overline{IRQ}[6:0]$ have the same priority as $\overline{IRQ}$, software must poll an interrupt status register (FISTAT) to determine the source of the interrupt request. FISTAT is automatically cleared when it is read by the CPU. FISTAT can be read at any time but cannot be written. Refer to the descriptions of port F, FINTEN and FISTAT.

PORTF — Port F Data Register **$0002**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	PF6	PF5	PF4	PF3	PF2	PF1	PF0
RESET:	0	0	0	0	0	0	0	0
Alt. Pin Func.:	—	$\overline{IRQ6}$	$\overline{IRQ5}$	$\overline{IRQ4}$	$\overline{IRQ3}$	$\overline{IRQ2}$	$\overline{IRQ1}$	$\overline{IRQ0}$

When corresponding bits in FINTEN are set, port F lines become interrupt request inputs. Writes to PORTF drive pins only if the pins are configured for output and corresponding $\overline{IRQ}$ is disabled.

FISTAT — Port F Interrupt Status **$0004**

	Bit 7	6	5	4	3	2	1	Bit 0
	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
RESET:	0	0	0	0	0	0	0	0

IS7 is the interrupt status bit for $\overline{IRQ}$ in port D. FISTAT can be read any time but cannot be written. All bits are cleared after CPU has read the register following an interrupt request.

IS7 — $\overline{IRQ}$ Status
0 = No interrupt pending for the corresponding interrupt line
1 = Interrupt pending for the corresponding interrupt line

IS[6:0] — $\overline{IRQ}[6:0]$ Status
0 = No interrupt pending for the corresponding interrupt line
1 = Interrupt pending for the corresponding interrupt line

FINTEN — Port F Interrupt Enable **$0005**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	IE6	IE5	IE4	IE3	IE2	IE1	IE0
RESET:	0	0	0	0	0	0	0	0

The enable bit for $\overline{IRQ}$ is located in the DIOCTL register.

IE[6:0] — $\overline{IRQ}[6:0]$ Enable
0 = Interrupt request input is disabled and pin is controlled by DDRF bit
1 = Interrupt request input $\overline{IRQx}$ is enabled, overriding state of DDRF bit

7 Main Timer

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

The timer has three input capture channels, four output compare channels, and one channel that can be configured as a fourth input capture or a fifth output compare.

Refer to the following table for a summary of the crystal-related frequencies and periods.

Table 8 Timer Summary

Control Bits	XTAL Frequencies			
	4.0 MHz	**8.0 MHz**	**12.0 MHz**	**Other Rates**
	1.0 MHz	**2.0 MHz**	**3.0 MHz**	**(E)**
	1000 ns	**500 ns**	**333 ns**	**(1/E)**
PR[1:0]	**Main Timer Count Rates**			
0 0 1 count — overflow —	1000 ns 65.536 ms	500 ns 32.768 ms	333 ns 21.845 ms	(1/E) (2^{16}/E)
0 1 1 count — overflow —	4.0 µs 262.14 ms	2.0 µs 131.07 ms	1.333 µs 87.381 ms	(4/E) (2^{18}/E)
1 0 1 count — overflow —	8.0 µs 524.28 ms	4.0 µs 262.14 ms	2.667 µs 174.76 ms	(8/E) (2^{19}/E)
1 1 1 count — overflow —	16.0 µs 1.049 ms	8.0 µs 524.29 ms	5.333 µs 349.52 ms	(16/E) (2^{20}/E)
RTR[1:0]	**Periodic (RTI) Interrupt Rates**			
0 0 0 1 1 0 1 1	8.192 ms 16.384 ms 32.768 ms 65.536 ms	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^{13}/E) (2^{14}/E) (2^{15}/E) (2^{16}/E)
CR[1:0]	**COP Watchdog Time-out Rates**			
0 0 0 1 1 0 1 1	32.768 ms 131.072 ms 524.288 ms 2.098 s	16.384 ms 65.536 ms 262.14 ms 1.049 s	10.923 ms 43.691 ms 174.76 ms 699.05 ms	(2^{15}/E) (2^{17}/E) (2^{19}/E) (2^{21}/E)
Time-out Tolerance (–0 ms/+...)	32.8 ms	16.4 ms	10.9 ms	(2^{15}/E)

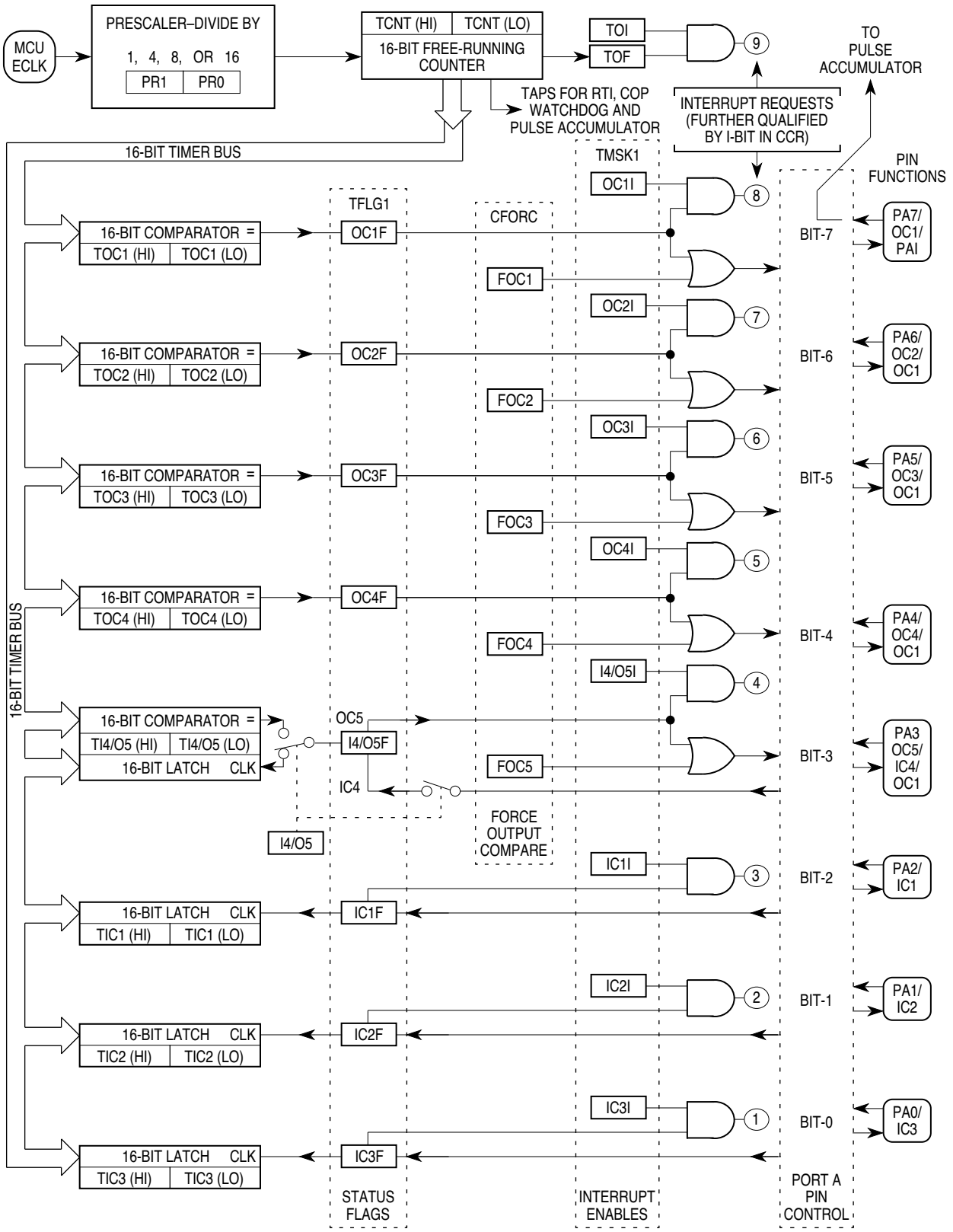


Figure 8 Timer Block Diagram

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
$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 9 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.

NOTE

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

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 –OC4F — 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

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 = TOF interrupts disabled
1 = Interrupt requested when TOF is set

RTII — Real-time Interrupt Enable
0 = RTIF interrupts disabled
1 = Interrupt requested when PAOVF is set

PAOVI — Pulse Accumulator Overflow Interrupt Enable
Refer to **8 Pulse Accumulator**.

PAII — Pulse Accumulator Input Interrupt Enable
Refer to **8 Pulse Accumulator**.

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

PR[1:0] — Timer Prescaler Select
In normal modes, PR0 and PR1 can only be written once, and the write must occur within 64 cycles after reset. The following table shows the prescaler selected with each combination of PR[1:0]. Refer to Table 8 table for specific timing values.

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

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 Flag
Set when TCNT changes from $FFFF to $0000

RTIF — Real-Time Interrupt Flag
Set periodically. Refer to RTR[1:0] in PACTL register.

PAOVF — Pulse Accumulator Overflow Flag
Refer to **8 Pulse Accumulator**.

PAIF — Pulse Accumulator Input Edge Flag
Refer to **8 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 reads zero.

PAEN — Pulse Accumulator System Enable
Refer to **8 Pulse Accumulator**.

PAMOD — Pulse Accumulator Mode
Refer to **8 Pulse Accumulator**.

PEDGE — Pulse Accumulator Edge Control
Refer to **8 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 **8 Pulse Accumulator**.

OPTION — System Configuration Options **$0039**

	Bit 7	6	5	4	3	2	1	Bit 0
	ADPU	CSEL	IRQE*	DLY*	CME	—	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 **11 Analog-to-Digital Converter**.

CSEL — Clock Select
0 = A/D charge pumps use system E clock
1 = A/D charge pumps use internal RC clock

IRQE — IRQ Select Edge-Sensitive Only
Refer to **6 Resets and Interrupts**.

DLY — Enable Oscillator Start-up Delay
Refer to **6 Resets and Interrupts**.

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

Bit 2 — Not implemented
Always reads zero

CR[1:0] — COP Timer Rate Select
Refer to the following table of COP timer rates.

Table 10 COP Timer Rate Select

CR[1:0]	Divide E/ 2^{15} By	XTAL = 4.0 MHz Time-Out –0 ms, +32.8 ms	XTAL = 8.0 MHz Time-Out –0 ms, +16.4 ms	XTAL = 12.0 MHz Time-Out –0 ms, +10.9 ms
0 0	1	32.768 ms	16.384 ms	10.923 ms
0 1	4	131.072 ms	65.536 ms	43.691 ms
1 0	16	524.288 ms	262.140 ms	174.76 ms
1 1	64	2.097 s	1.049 s	699.05 ms
	E =	1.0 MHz	2.0 MHz	3.0 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 to COPRST to arm COP watchdog clearing mechanism. Write $AA (%10101010) to COPRST to reset COP watchdog.

8 Pulse Accumulator

The MC68HC11C0 has an 8-bit counter that can be configured for gated time accumulation or to operate as a simple event counter. The pulse accumulator counter can be read or written at any time.

The PA7 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.

	Selected Crystal	Common XTAL Frequencies		
		4.0 MHz	8.0 MHz	12.0 MHz
CPU Clock	(E)	1.0 MHz	2.0 MHz	3.0 MHz
Cycle Time	(1/E)	1000 ns	500 ns	333 ns
Pulse Accumulator (in Gated Mode)				
(2^6/E)	1 count —	64.0 µs	32.0 µs	21.330 µs
(2^{14}/E)	overflow —	16.384 ms	8.192 ms	5.491 ms

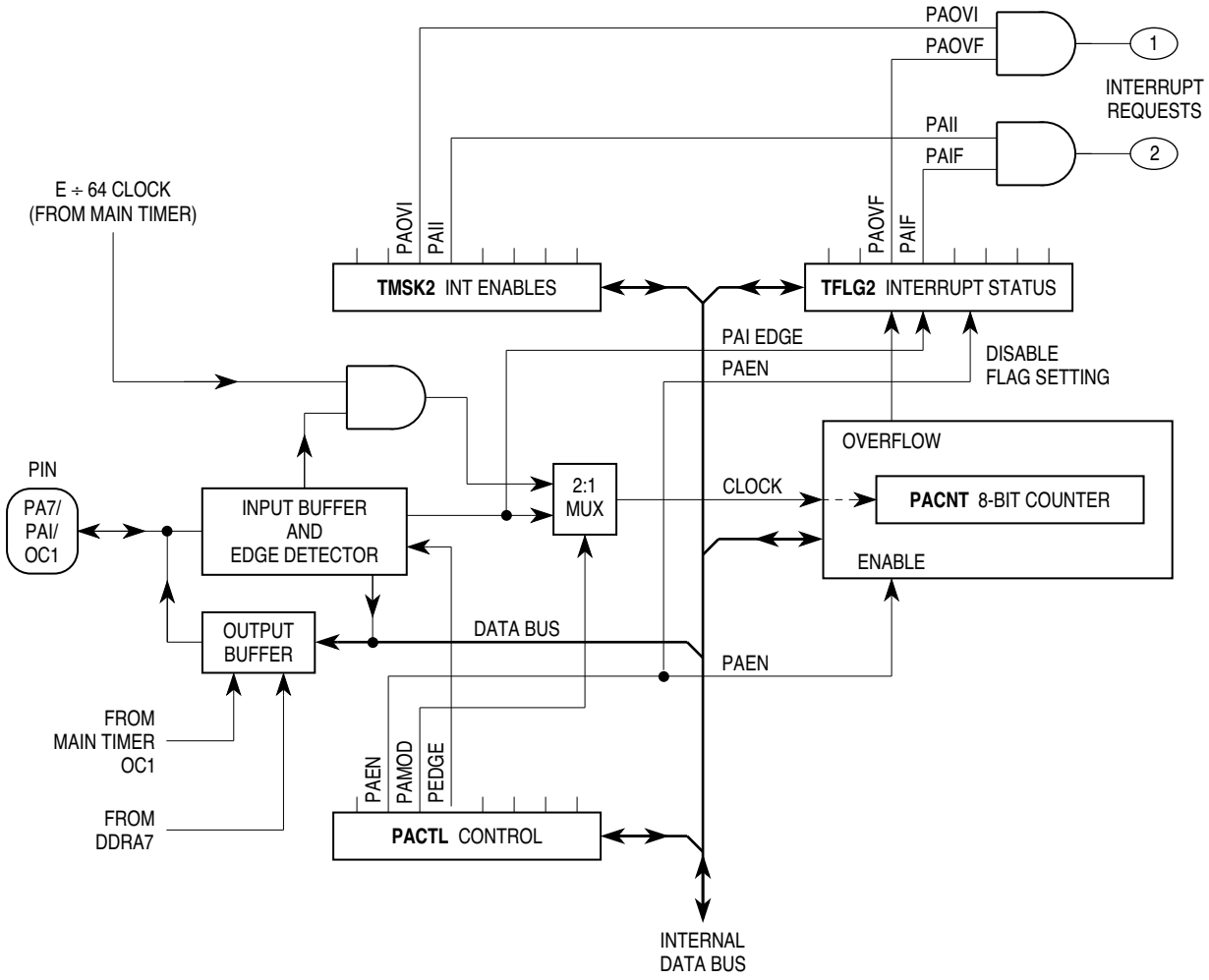


Figure 9 Pulse Accumulator System Block Diagram

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 **7 Main Timer**.

RTII — Real-time Interrupt Enable
Refer to **7 Main Timer**.

PAOVI — Pulse Accumulator Overflow Interrupt Enable
0 = PAOVF interrupts disabled
1 = Interrupt requested when PAOVF is set to one

PAII — Pulse Accumulator Input Edge Interrupt Enable
0 = PAIF interrupts disabled
1 = Interrupt requested when PAIF is set to one

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

PR[1:0] — Timer Prescaler Select
Refer to **7 Main Timer**.

NOTE

Bits in TMSK2 correspond bit for bit with flag bits 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 Flag
Refer to **7 Main Timer**.

RTIF — Real-Time Interrupt Flag
Refer to **7 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

PAMOD	PEDGE	Action on Clock
0	0	PAI Falling Edge Increments the Counter.
0	1	PAI Rising Edge Increments the Counter.
1	0	A Zero on PAI Inhibits Counting.
1	1	A One on PAI Inhibits Counting.

Bit 3 — Not implemented
Always reads zero

I4/O5 — Input Capture 4/Output Compare 5
Refer to **7 Main Timer**.

RTR[1:0] — Real-Time Interrupt Rate
These two bits select the rate for periodic interrupts. Refer to the following table.

Table 11 Real-Time Interrupt Rates

RTR[1:0]	Divide E Into	XTAL = 4.0 MHz	XTAL = 8.0 MHz	XTAL = 12.0 MHz
0 0	2^{13}	8.19 ms	4.096 ms	2.731 ms
0 1	2^{14}	16.38 ms	8.192 ms	5.461 ms
1 0	2^{15}	32.77 ms	16.384 ms	10.923 ms
1 1	2^{16}	65.54 ms	32.768 ms	21.845 ms
	E =	1.0 MHz	2.0 MHz	3.0 MHz

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, unaffected by reset.

9 Pulse-Width Modulation Timer

The MC68HC11C0 MCU contains a PWM timer that is composed of two 8-bit modulators. Each of the modulators can create independent continuous waveforms with software-selectable duty rates from 0% to 100%.

The PWM system provides up to two pulse-width modulated waveforms on port H pins. Each channel has its own counter. The two counters can be concatenated to create a single 16-bit PWM output based on 16-bit counts. Two clock sources (A 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 function. The PWPOL register determines the polarity for each channel polarity and selects the clock source for each channel. The PWSCAL register derives a user-scaled clock based on the A-clock source. 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 a PWM channel configured for 8-bit mode and E equal to 2 MHz, frequencies can be produced from one-half the E clock rate to more than 8 seconds per cycle. By configuring the PWM output for 16-bit mode with E equal to 2 MHz, periods can be produced from one-half the E clock frequency to more than 35 minutes per cycle.

In 16-bit mode, a PWM frequency of 60 Hz corresponds to a duty cycle resolution of only 30 parts per million (0.0003%). In the same system, a PWM frequency of 1 kHz corresponds to a duty cycle resolution of 0.050%.

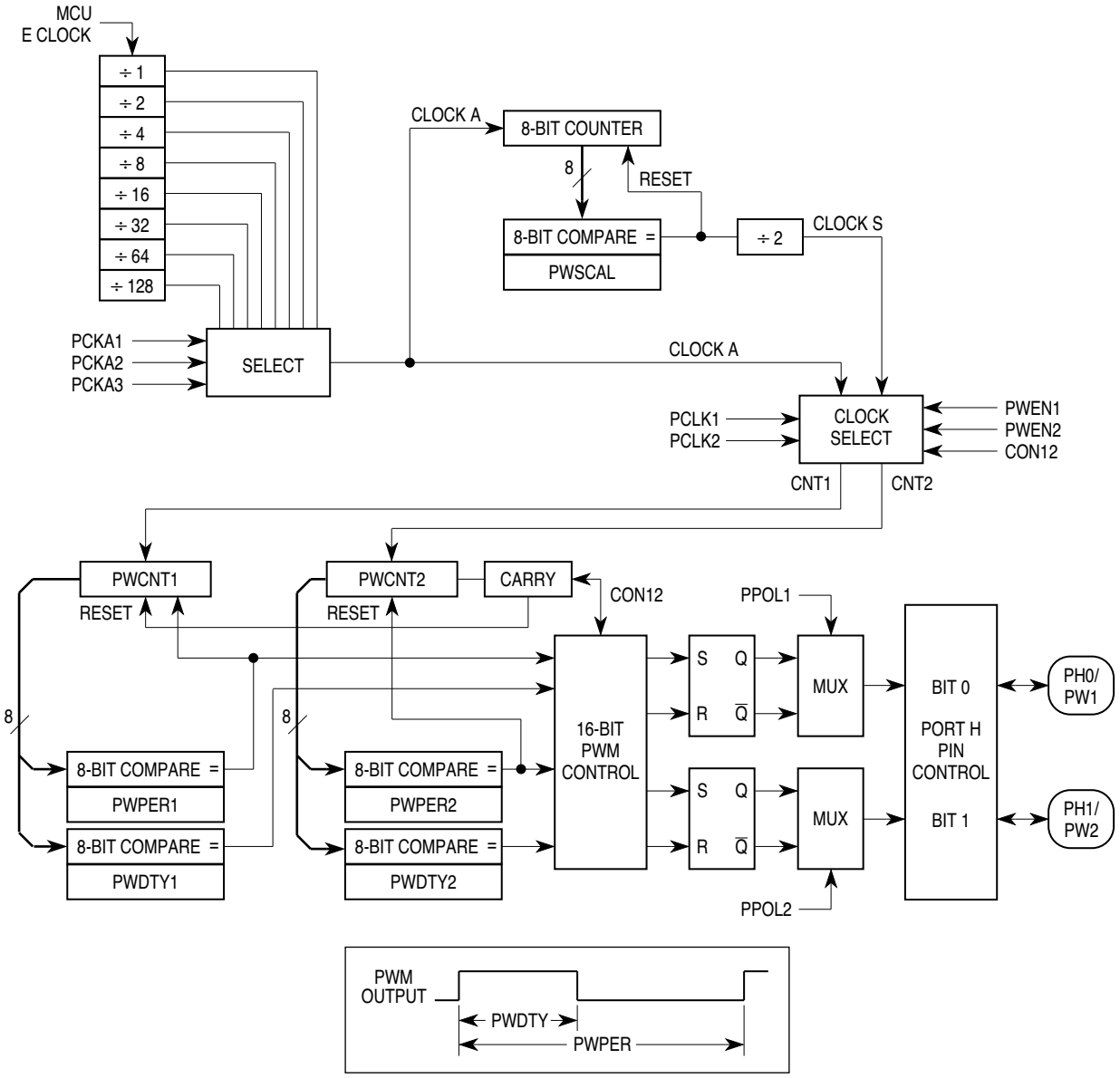


Figure 10 Pulse Width Modulation System Block Diagram

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

	Bit 7	6	5	4	3	2	1	Bit 0
	—	CON12	—	—	—	PCKA3	PCKA2	PCKA1
RESET:	0	0	0	0	0	0	0	0

Bit 7 — Not implemented
Always reads zero

CON12 — Concatenate Channels One and Two
Channel 1 is high order byte, and channel 2 is the low-order byte. The output appears on port H, bit 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.

Bits [5:3] — Not implemented
Always read zero

PCKA[3:1] — Prescaler for Clock A
Determines the rate for clock A

PCKA[3:1]	Value of Clock A
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
	—	—	PCLK2	PCLK1	—	—	PPOL2	PPOL1
RESET:	0	0	0	0	0	0	0	0

PWPOL can be written anytime. If values in PWPOL are changed during a PWM output, a stretched or truncated signal may be generated.

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

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

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

PPOL[2:1] — 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	—	—	—	—	PWEN2	PWEN1
RESET:	0	0	0	0	0	0	0	0

TPWSL — PWM Scaled Clock Test Bit (TEST)

DISCP — Disable Compare Scaled E Clock (TEST)

Bits [5:2] — Not implemented
Always read zero

PWEN[2:1] — Pulse-Width Channel 1–2

0 = Channel disabled

1 = Channel enabled

TPWCNT1–PWCNT2 — Pulse-Width Modulation Timer Counter 1 to 2 **$0066–$0067**

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

PWCNT1–PWCNT2
Begins count using the selected clock. A write to PWCNTx registers causes them to reset to $00.

PWPER1–PWPER2 — Pulse-Width Modulation Timer Period 1 to 2 **$006A–$006B**

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

PWPER1–PWPER2
Determines period of associated PWM channel

Period registers can be written any time. A new value written to a period register does not take effect until the counter resets and the new value is latched. Users may begin the new period immediately by writing the new value to the period register, then writing any value to the counter. The counter will reset and the new period value will be latched. If the value in the period register is equal to or less than the value in the duty register, there will be no change in state. Refer to **9.1 PWM Boundary Cases**.

PWDTY1–PWDTY2 — Pulse-Width Modulation Timer Duty Cycle 1 to 2 **$006E–$006F**

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

PWDTY1–PWDTY2

Determines duty cycle of associated PWM channel

Duty cycle registers can be written any time. A new value written to a duty register does not take effect until the counter resets and the new value is latched. Users may begin the new duty cycle immediately by writing the new value to the duty register, then writing any value to the counter. The counter will reset and the new duty cycle value will be latched. If the value in the duty register is equal to or greater than the value in the period register, there will be no change in state. Refer to **9.1 PWM Boundary Cases**.

9.1 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

10 Serial Subsystems

The MC68HC11C0 contains two serial communication subsystems that allow the MCU to transfer data to and from other devices. One system, the serial communications interface (SCI), is an asynchronous nonreturn to zero (NRZ) serial system that supports single-wire data transmissions. The second system, the serial peripheral interface (SPI), is a synchronous master/slave system that allows data to be both transmitted and received simultaneously.

As in other M68HC11 MCUs, the serial systems are implemented as alternate functions of port D pins. However, the interrupt request lines found elsewhere on other M68HC11 MCUs are now a third function of port D pins. Therefore, two additional registers are associated with port D on the MC68HC11C0. The port D I/O control register (DIOCTL) and the port D output mode register (DODM) have been added.

Briefly, DIOCTL controls the function performed by each port D pin. The DODM register controls the output driver type for each port D pin. Refer to the following descriptions of PORTD, DDRD, DIOCTL, and DODM registers.

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	SDO/ MOSI	SDI/ MISO	TxD	RxD
or:	—	—	IRQ	—	—	XIRQ	—	—

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 = Input
1 = Output

DIOCTL — Port D I/O Control **$0007**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	—	DIO5	DIO4	DIO3	DIO2	—	DIO0
RESET:	0	0	1	1	1	1	0	0

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

DIO[5:2] — Port D I/O Control for Port D Bits [5:2]
Refer to the following tables for description.

Bit 1 — Not implemented
Always reads zero

DIO0 — Port D I/O Control for Port D Bit 0
Refer to the following tables for description.

Table 12 PD5 Configuration

SPI Enable	DIOCTL Bit 5	DIOCTL Bit 4	PD5	Pull-Up Control	Output Enable
SPE = 0	0	X	I/O	DDD5	DDD5
	1	0	$\overline{IRQ}$	On	Off
	1	1	I/O	DDD5	DDD5
SPE = 1	0	X	I/O	DDD5	DDD5
	1	0	$\overline{IRQ}$	On	Off
	1	1	$\overline{SS}$	Off	MSTR + DDD5

Notes:

1. If a pin is configured for general-purpose I/O, the DDRD bit controls the direction of data.
2. If a pin is configured as $\overline{SS}$, the MSTR and DDRD bits are the output enable.
3. SPE is the SPI enable bit, MSTR is the master/slave select bit. Refer to the SPCR register.
4. $\overline{IRQ}$ or $\overline{SS}$ inputs are internally pulled high if not used. This does not affect the pin.

Table 13 PD[4:3] Configuration

SPI Enable	PD4	PD3	Pull-Up Control	Output Enable
SPE = 0	I/O	I/O	DDD[4:3]	DDD[4:3]
SPE = 1	MOSI	SCK	Off	MSTR + DDD[4:3]

Notes:

1. If a pin is configured for general-purpose I/O, the DDRD bit controls the direction of data.
2. If a pin is configured as MOSI, the MSTR and DDRD bit control the direction of data.
3. SPE is the SPI enable bit, MSTR is the master/slave select bit. Refer to the SPCR register.

Table 14 PD2 Configuration

SPI Enable	DIOCTL Bit 3	DIOCTL Bit 2	PD2	Pull-Up Control	Output Enable
SPE = 0	0	X	I/O	DDD[3:2]	DDD[3:2]
	1	0	$\overline{XIRQ}$	On	Off
	1	1	I/O	DDD[3:2]	DDD[3:2]
SPE = 1	0	X	I/O	DDD[3:2]	DDD[3:2]
	1	0	$\overline{XIRQ}$	On	Off
	1	1	MISO	Off	MSTR + DDD[3:2]

Notes:

1. If a pin is configured for general-purpose I/O, the DDRD bit controls the direction of data.
2. If a pin is configured as MISO, the MSTR and DDRD bit control the direction of data.
3. SPE is the SPI enable bit, MSTR is the master/slave select bit. Refer to the SPCR register.
4. $\overline{XIRQ}$ or MISO inputs are internally pulled high if not used. This does not affect the pin.

Table 15 PD[1:0] Configuration

DIOCTL Bit 0	SCI Enables	PD1	Pull-Up Control	SCI Enables	PD0	Pull-Up Control	SCI Mode
0	TE = 1	TxD	Off	RE = 1	RxD	Off	Two Wire
	TE = 0	I/O	DDD1	RE = 0	I/O	DDD0	
1	TE = 1	TxD	Off	X	I/O	DDD0	Single Wire
	RE = 0						
	TE = 1	RxD	Off				
	RE = 0						
	TE = 1	RxD/TxD	Off				
	RE = 0	Looped					
	TE = 1	I/O	DDD1				
	RE = 0						

Notes:
1. If a pin is configured for general-purpose I/O, the DDRD bit controls the direction of data.
2. If a pin is configured for RxD input, there is no weak pull-up at the pin.
3. If a pin is configured for TxD output, the output can be either an open-drain output or a CMOS driver. This is controlled by the port D driver output mode (DODM) register.
4. TE is the transmitter enable bit. RE is the receiver enable bit. Refer to the SCCR2 register.

NOTE

If any of the pins PD[5:2] are configured as SPI inputs they will not have pull-ups. If any of the pins PD[5:2] are configured as SPI outputs they will be either open-drain outputs or normal CMOS driver outputs depending on the state of the corresponding bit in the DODM register.

DODM — Port D Open Drain Mode **$0075**

	Bit 7	6	5	4	3	2	1	Bit 0
	—	DOD6	DOD5	DOD4	DOD3	DOD2	DOD1	DOD0
RESET:	0	0	0	0	0	0	0	0

Each DODM bit controls an individual port D pin and is valid only if the pin is configured as an output.

DOD[6:0] — Port D Open Drain Bits [6:0]
0 = Corresponding port D output pin configured as normal CMOS driver.
1 = Corresponding port D output pin configured as open-drain output driver.

10.1 Serial Communications Interface (SCI)

The SCI is a universal asynchronous receiver transmitter (UART) serial communications interface, an independent serial I/O subsystem in the MC68HC11C0. It has a standard NRZ format (one start bit, eight or nine data bits and one stop bit) and several baud rates available. The SCI transmitter and receiver are independent, but use the same data format and bit rate. Refer to the two tables in the BAUD register description for a summary of the SCI baud rate values.

The MC68HC11C0 SCI system supports single-wire transmit and receive operation. Although SCI systems in other M68HC11 MCUs support external wire-OR operation, two pins are necessary to perform that function. The DIO0 bit in DIOCTL register controls the SCI mode of operation. When single-wire operation is selected (DIO0 = 1), the SCI uses only PD1 for its transmit and receive signals. In this mode, if the receiver is enabled, PD1 is the RxD input; if the transmitter is enabled, PD1 is the TxD output. If both the receiver and the transmitter are enabled, both RxD and TxD are internally tied together

and share the PD1 pin. If an open-drain type driver is used for the TxD output, RxD and TxD can still be externally wire-OR configured. The open-drain control in DIOCTL allows either full CMOS driving or open-drain driving on the TxD output. Refer to the DIOCTL description.

The SCI system external pins are implemented as an alternate function of port D pins. In addition to the port D data direction register (DDRD), the port D I/O control register (DIOCTL) determines which functions are performed by port D pins. The port D open drain mode (DODM) register controls the driver type for each port D pin configured as an output. Refer to the descriptions of PORTD, DDRD, DIOCTL, and DODM registers.

The bits in three control registers and one status register control operation of the SCI. The SCI control registers (SCCR1 and SCCR2) are used to configure the SCI and to enable certain features. The baud rate control register (BAUD) selects the SCI prescaler and baud rate. Bits in the SCI status register (SCSR) flag certain occurrences and control the SCI system accordingly. If enabled by bits in SCCR2, some SCI status bits in SCSR can generate interrupt requests. Interrupt-generating flag bits in SCSR are cleared automatically when the CPU services the SCI request. Refer to the descriptions of SCCR1, SCCR2, BAUD, and SCSR.

The SCI data register (SCDR) is comprised of two separate registers — the receive data register and the transmit data register. When SCDR is read, the receive data register is accessed. When SCDR is written, the transmit data register is accessed. If nine-bit data format is used, R8 and T8 bits in SCCR1 contain the ninth receive data bit and the ninth transmit data bit, respectively. Both the transmit and receive data registers are coupled to serial shift registers. When data to be transmitted is written to SCDR the data is shifted from the transmit data register to the serial shift register in a parallel fashion. The contents of the shift register are then transmitted serially out the TxD pin. When serial data is received on the RxD pin, it enters the serial shift register. When the shift register is full, the entire contents are shifted in parallel to the SCDR register. The size of the shift register changes automatically according to the state of the M bit in SCCR1. However, if nine-bit data is selected it is necessary to read or write the ninth data bit (R8 or T8) in SCCR1 first to ensure that the contents of the shift register are correct. Refer to the block diagrams for the SCI receiver, SCI transmitter, and baud rate generator.

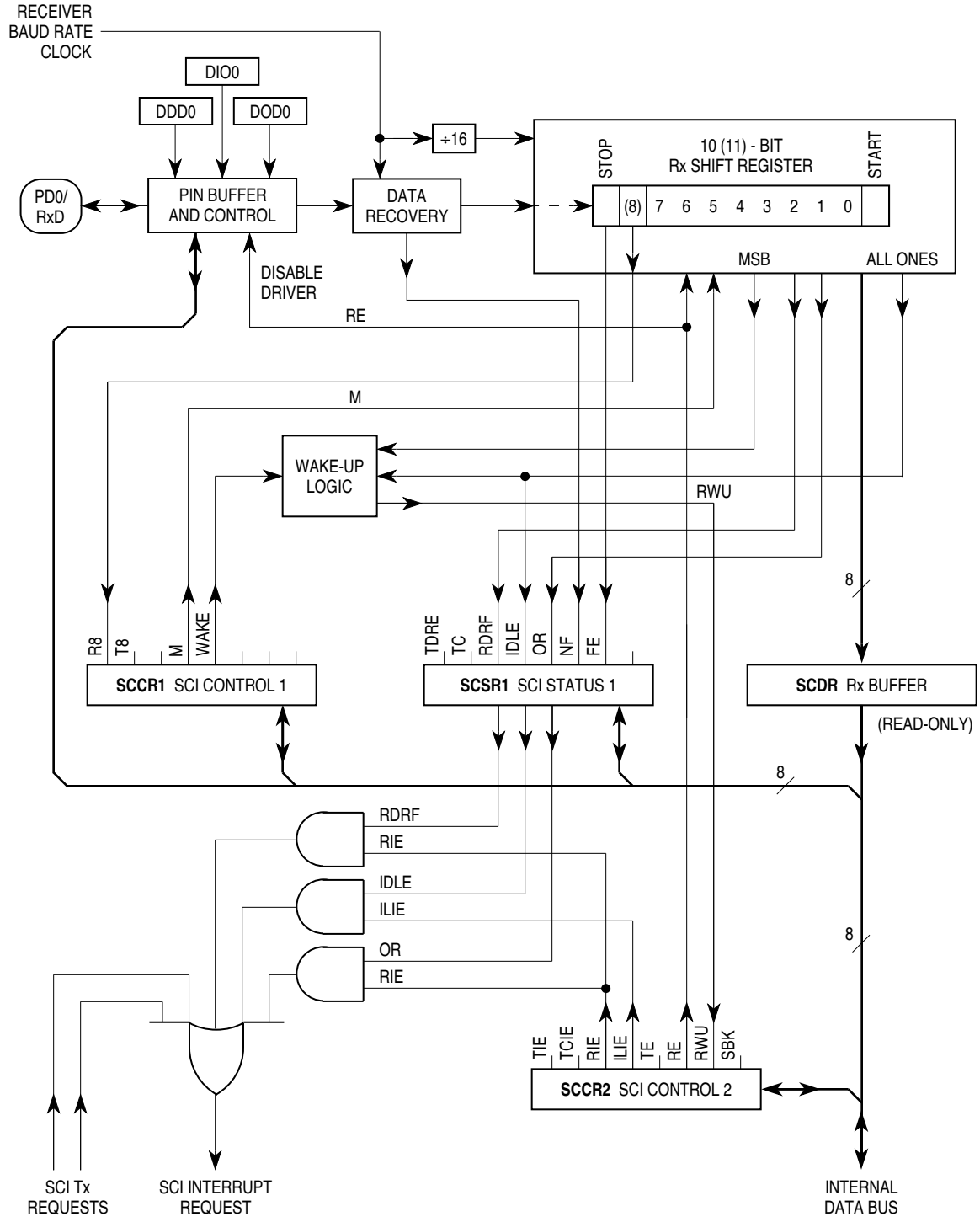


Figure 11 SCI Receiver Block Diagram

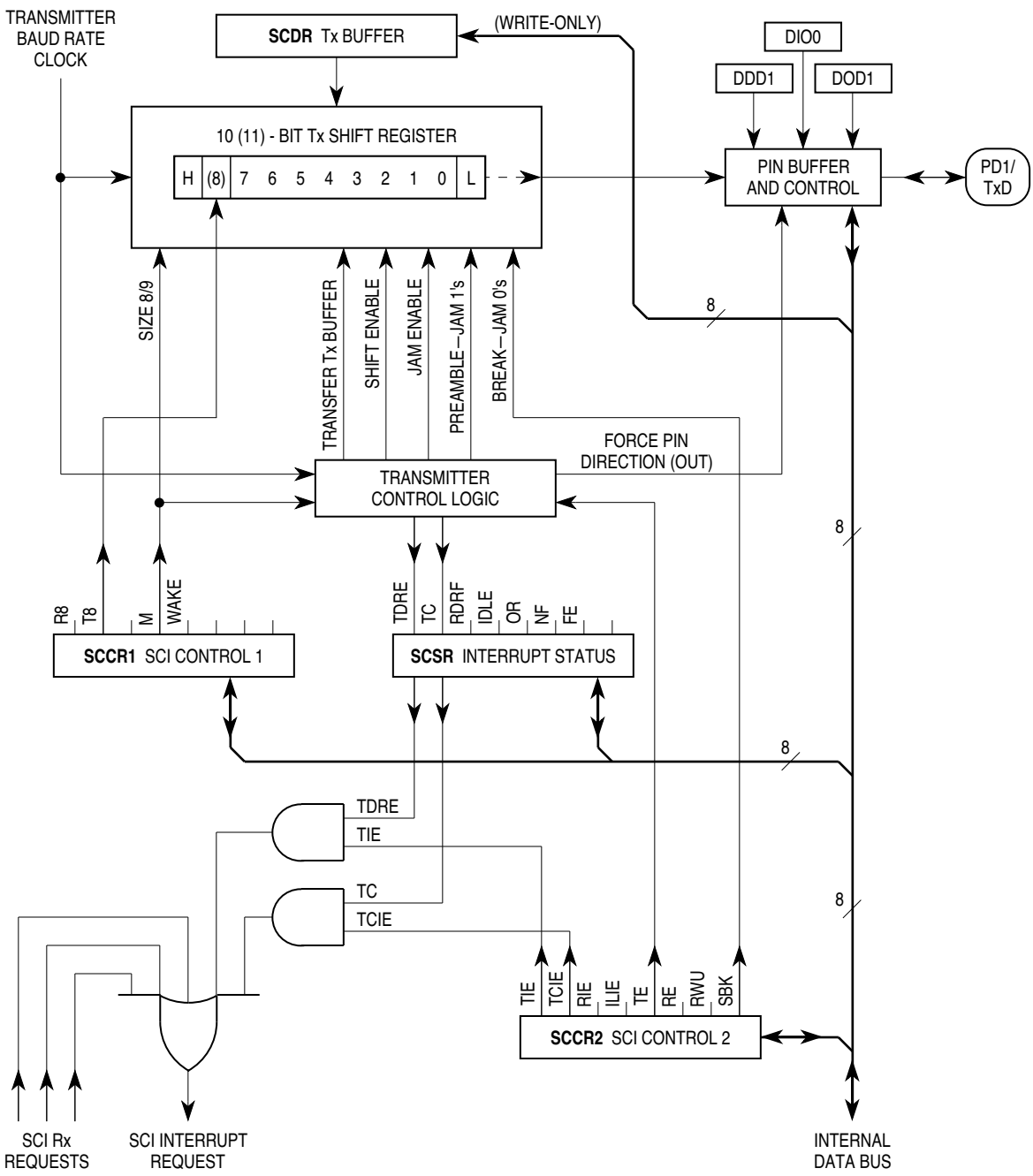


Figure 12 SCI Transmitter Block Diagram

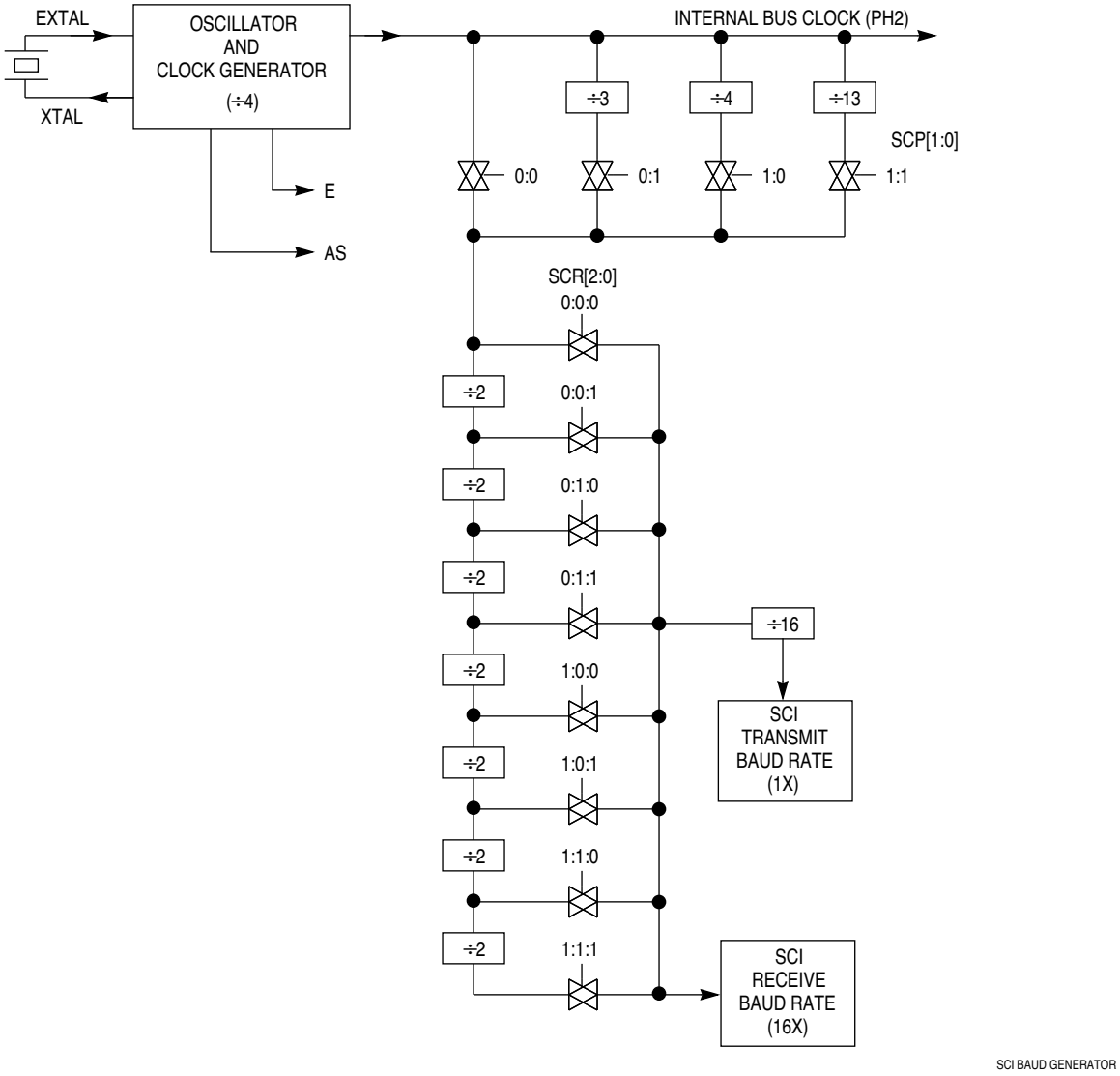


Figure 13 SCI Baud Generator Circuit Diagram

BAUD — Baud Rate Control Register **$002B**

	Bit 7	6	5	4	3	2	1	Bit 0
	TCLR	—	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0
RESET:	0	0	0	0	0	U	U	U

TCLR — Clear Baud Rate Counters

TCLR can only be set in test modes.

1 = Clear baud rate counter chain for testing purposes.

0 = Normal SCI operation

SCP[1:0] — SCI Baud Rate Prescaler Selects

Shaded boxes contain the prescaler rates used in the following table. Refer to the SCI Baud Rate Clock Diagram.

SCP[1:0]	Divide Internal Clock By	Crystal Frequency in MHz		
		4.0 MHz (Baud)	8.0 MHz (Baud)	12.0 MHz (Baud)
0 0	1	62.50 K	125.0 K	187.5 K
0 1	3	20.83 K	41.67 K	62.5 K
1 0	4	15.625 K	31.25 K	46.88 K
1 1	13	4800 K	9600 K	14.4 K

RCKB — SCI Baud Rate Clock Check

RCKB can only be set in test modes.

1 = Exclusive-OR of the RT clock driven out TxD pin for testing purposes.

0 = Normal SCI operation

SCR2, SCR1, and SCR0 — SCI Baud Rate Selects

Selects receiver and transmitter bit rate based on output from baud rate prescaler stage. Shaded boxes in the table below contain the prescaler output rates shown in the preceding table. Refer to the SCI Baud Rate Clock Diagram.

SCR[2:0]	Divide Prescaler By	Baud Rate (Prescaler Output from Previous Table)		
		4800 K	9600 K	14.4 K
0 0 0	1	4800	9600	**14.4**
0 0 1	2	2400	4800	7200
0 1 0	4	1200	2400	3600
0 1 1	8	600	1200	1800
1 0 0	16	300	600	1200
1 0 1	32	150	300	450
1 1 0	64	75	150	225
1 1 1	128	37.5	75	112.5

SCCR1 — SCI Control Register 1 **$002C**

	Bit 7	6	5	4	3	2	1	Bit 0
	R8	T8	—	M	WAKE	—	—	—
RESET:	U	U	0	0	0	0	0	0

R8 — Receive Data Bit 8

When the M-bit is set, R8 stores the ninth data bit in the receive data character. R8 can also be used with 8-bit data to support several special receive data formats. R8 remains unchanged following a transmission and may be used again without rewriting it.

T8 — Transmit Data Bit 8

When the M-bit is set, T8 stores the ninth data bit in the transmit data character. T8 can also be used with 8-bit data to support several special transmit data formats. T8 remains unchanged following a transmission and may be used again without rewriting it.

Bit 5 — Not implemented

Always reads zero

M — Mode (Select Character Format)

M selects either 8- or 9-bit data characters. Format is one start bit, eight or nine data bits, and one stop bit. If 9-bit data is selected R8 and T8 store the ninth receive and transmit data bit, respectively.

0 = 8-bit data characters
1 = 9-bit data characters

WAKE — Wakeup by Address Mark/Idle

0 = Wakeup by IDLE line recognition
1 = Wakeup by address mark (most significant data bit set)

Bits [2:0] — Not implemented

Always read zero

SCCR2 — SCI Control Register 2 **$002D**

	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

SCSR — SCI Status Register **$002E**

	Bit 7	6	5	4	3	2	1	Bit 0
	TDRE	TC	RDRF	IDLE	OR	NF	FE	—
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 SCSR 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 SCSR 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 SCSR 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 SCSR 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 SCSR 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 SCSR 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 SCSR and then reading SCDR.

0 = Stop bit detected
1 = Zero detected

Bit 0 — Not implemented

Always reads zero

SCDR — SCI Data Register **$002F**

	Bit 7	6	5	4	3	2	1	Bit 0
	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0
RESET:	U	U	U	U	U	U	U	U

Receive and transmit are double buffered. Reads access the receive data buffer, and writes access the transmit data buffer. When the M bit in SCCR1 is set, R8 and T8 in SCCR1 store the ninth bit in receive and transmit data characters.

10.2 Serial Peripheral Interface (SPI)

The SPI allows the MCU to communicate synchronously with peripheral devices and other microprocessors. When configured as a master, data transfer rates can be as high as one-half the E clock rate (1 Mbit per second for a 2 MHz bus frequency). When configured as a slave, data transfers can be as fast as the E clock rate (2 Mbit per second for a 2 MHz bus frequency).

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

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

Software can select one of four combinations of serial clock phase and polarity using two bits in the SPI control register (SPCR). The clock polarity is specified by the CPOL control bit, which selects an active high or active low clock, and has no significant effect on the transfer format. The clock phase (CPHA) control bit selects one of two different transfer formats.

The SPI system external pins are implemented as an alternate function of port D pins. In addition to the port D data direction register (DDRD), the port D I/O control register (DIOCTL) determines which functions are performed by port D pins. The port D open drain mode (DODM) register controls the driver type for each port D pin configured as an output. Refer to the descriptions of PORTD, DDRD, DIOCTL, and DODM registers.

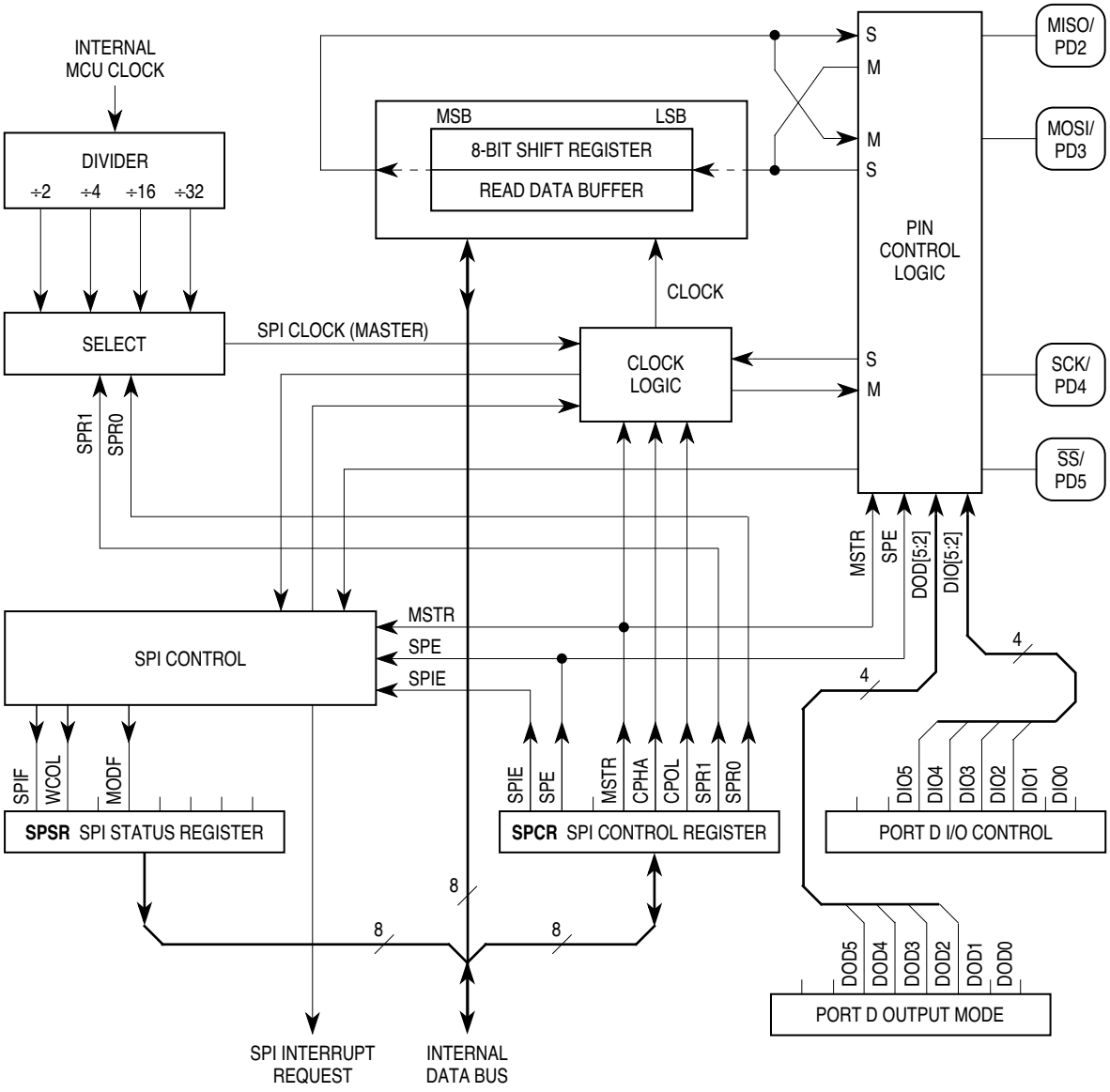


Figure 14 SPI Block Diagram

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

	Bit 7	6	5	4	3	2	1	Bit 0
	SPIE	SPE	—	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

Bit 5 — Not implemented
Always reads zero

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

CPOL, CPHA — Clock Polarity, Clock Phase
Refer to **Figure 15**.

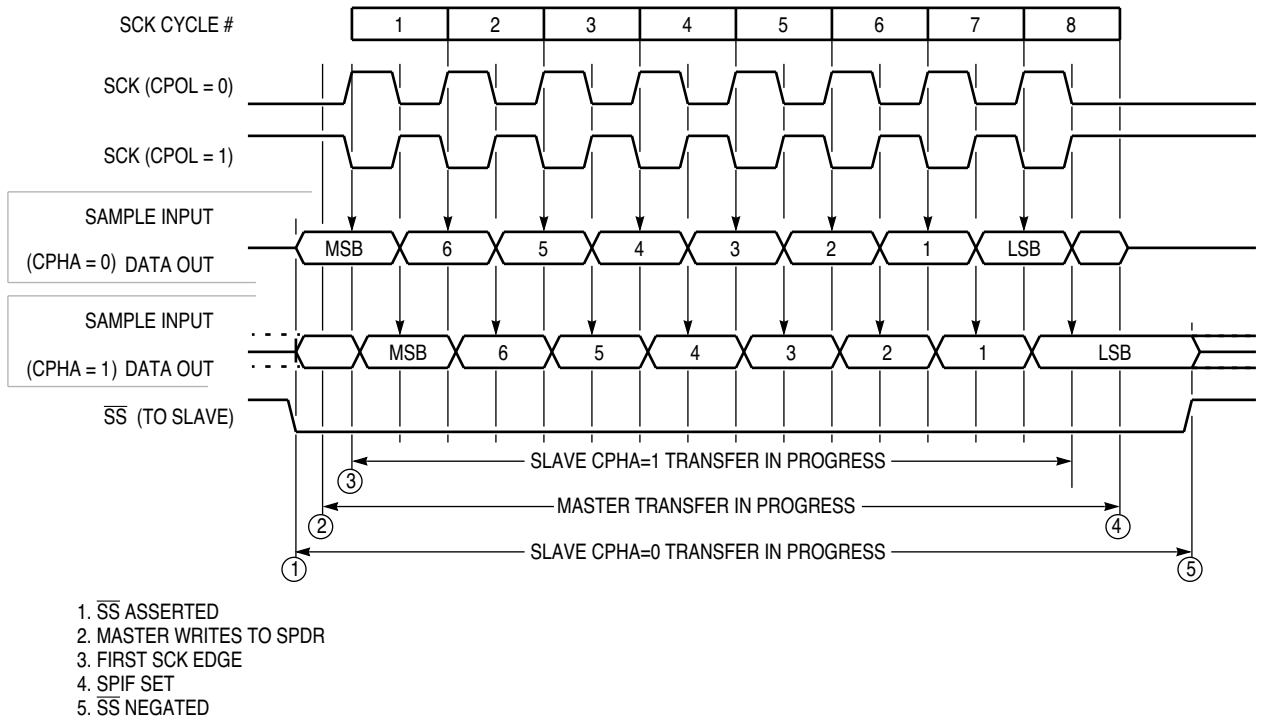


Figure 15 SPI Transfer Format

SPR[1:0] — SPI Clock Rate Selects

Table 16 SPI Clock Rate Selects

SPR[1:0]	Divide E Clock By	SPI Baud Rate at E = 1 MHz	SPI Baud Rate at E = 2 MHz	SPI Baud Rate at E = 3 MHz
0 0	2	500 kHz	1.0 MHz	1.5 MHz
0 1	4	250 kHz	500 kHz	750 kHz
1 0	16	125 kHz	125 kHz	375 kHz
1 1	32	62.5 kHz	62.5 kHz	187.5 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 Register **$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.

11 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 MC68HC11C0 A/D converter system, a four-channel multiplexed-input successive-approximation converter, is accurate to ±1 least significant bit (LSB). It does not require external sample and hold circuits because of the type of charge-redistribution technique used.

Dedicated pins 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 signals.

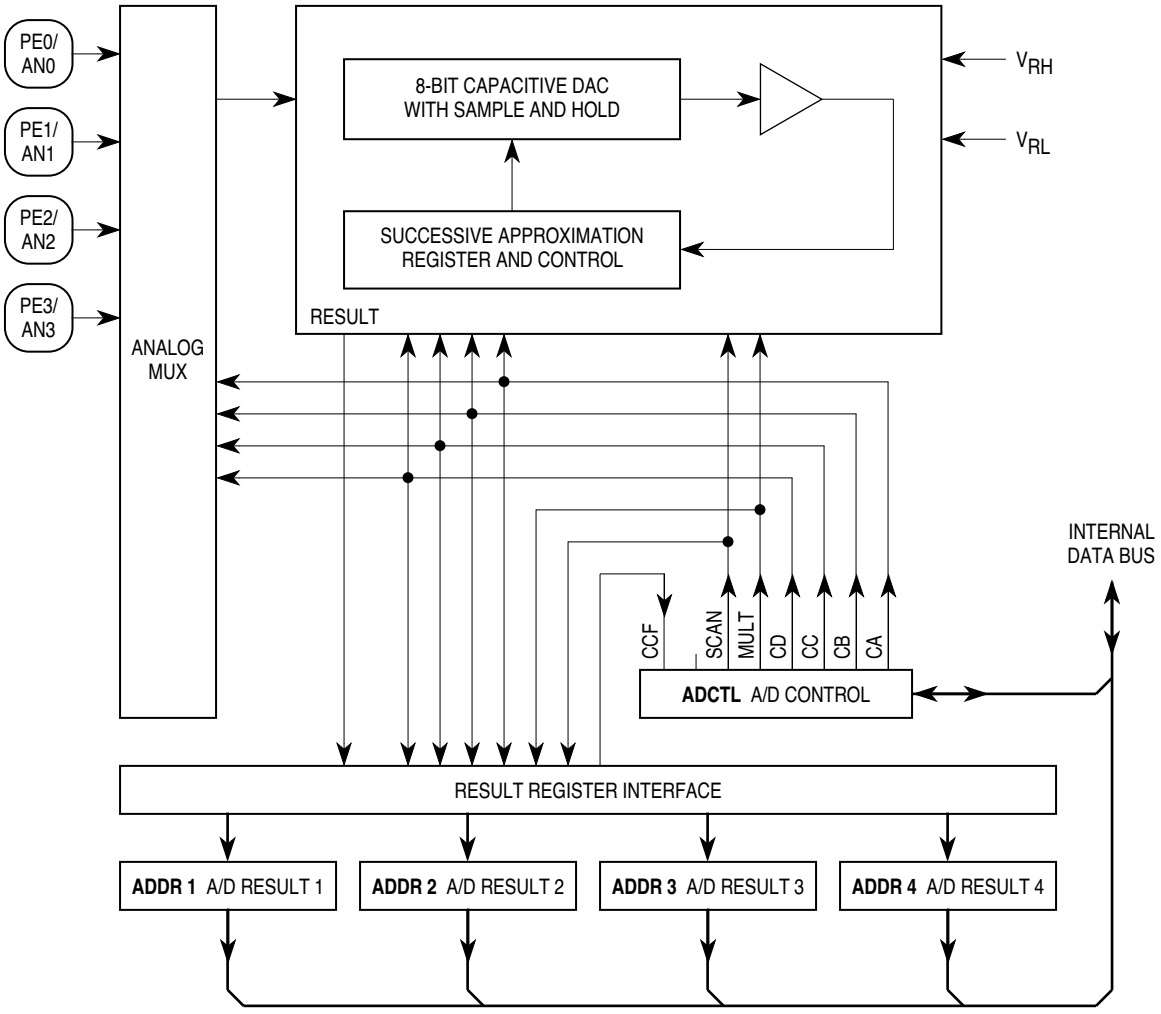


Figure 16 A/D Converter Block Diagram

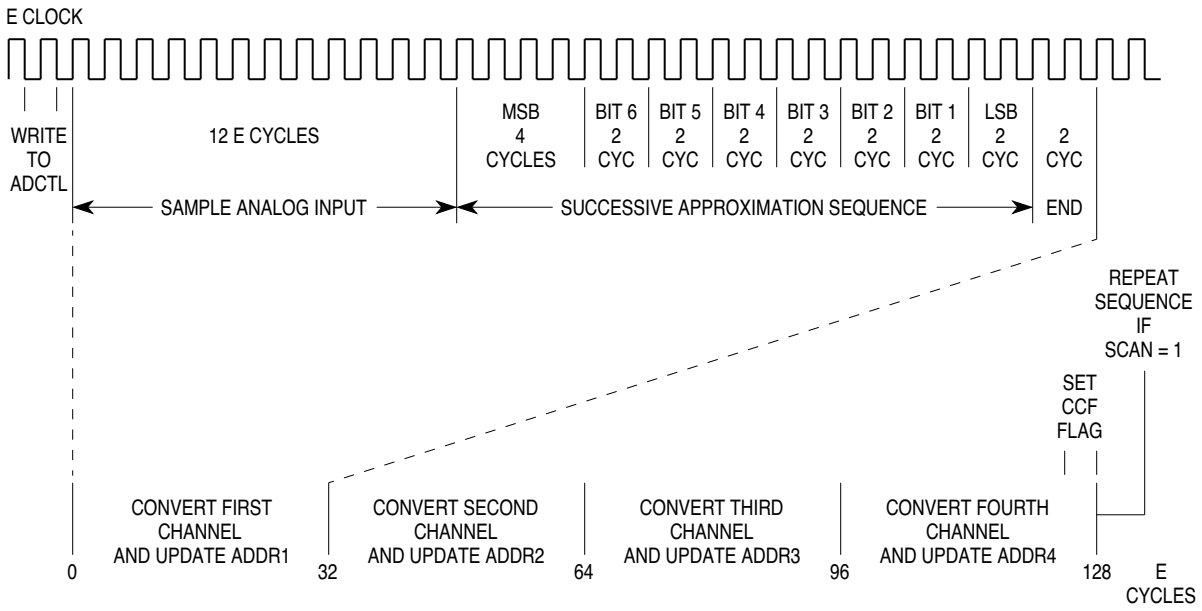
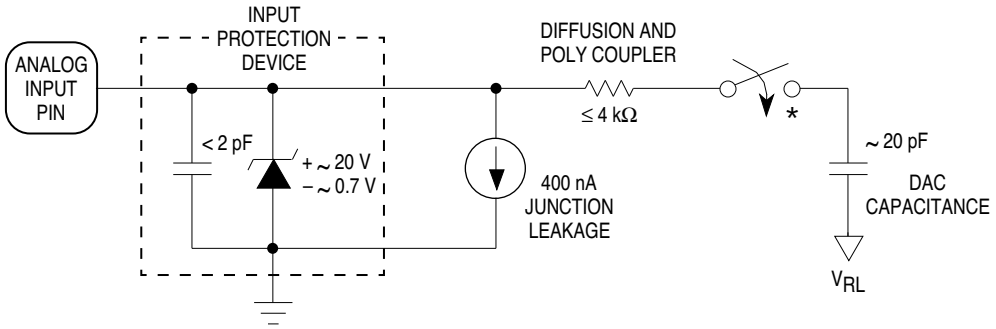


Figure 17 Timing Diagram for a Sequence of Four A/D Conversions



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

Figure 18 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:	I	0	I	I	I	I	I	I

CCF — Conversions Complete Flag
Set after an A/D conversion cycle
Cleared when ADCTL is written

Bit 6 — Not implemented
Always reads zero

SCAN — Continuous Scan Control
0 = Perform four conversions and stop
1 = Convert the four channels and continuously update result registers.

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

CD–CA — Channel Select D through A

Table 17 A/D Converter Channel Assignments

Channel Select Control Bits				Channel Signal	Result in ADRx if MULT = 1	Result in ADRx if MULT = 0
CD	CC	CB	CA			
0	0	0	0	AD0	ADR1	ADR[4:1]
0	0	0	1	AD1	ADR2	ADR[4:1]
0	0	1	0	AD2	ADR3	ADR[4:1]
0	0	1	1	AD3	ADR4	ADR[4:1]
0	1	0	0	Reserved	—	—
0	1	0	1	Reserved	—	—
0	1	1	0	Reserved	—	—
0	1	1	1	Reserved	—	—
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	ADR[4:1]
1	1	0	1	V_{RL}*	ADR2	ADR[4:1]
1	1	1	0	$(V_{RH})/2$*	ADR3	ADR[4:1]
1	1	1	1	Test/Reserved*	ADR4	ADR[4:1]

*Used for factory testing

ADR1–ADR4 — 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

Table 18 Analog Input to 8-Bit Result Translation Table

	Bit 7	6	5	4	3	2	1	Bit 0
% (1)	50%	25%	12.5%	6.25%	3.12%	1.56%	0.78%	0.39%
Volts (2)	2.500	1.250	0.625	0.3125	0.1562	0.0781	0.0391	0.0195

(1) % of $V_{RH} - V_{RL}$ (2) Volts for V_{RL} = 0; V_{RH} = 5.0 V

ADCTL — A/D Control/Status **$0030**

	Bit 7	6	5	4	3	2	1	Bit 0
	ADPU	CSEL	IRQE*	DLY*	CME	—	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, any time in special modes.

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

CSEL — Clock Select
0 = A/D converter uses system E clock
1 = A/D converter use internal RC clock

IRQE — $\overline{\text{IRQ}}$ Select Edge Sensitive Only
Refer to **6 Resets and Interrupts**.

DLY — Enable Oscillator Start-Up Delay on Exit from STOP
Refer to **6 Resets and Interrupts**.

CME — Clock Monitor Enable
Refer to **6 Resets and Interrupts**.

Bit 2 — Not implemented
Always reads zero

CR[1:0] — COP Timer Rate Select
Refer to **7 Main Timer**.

NOTES

How to reach us:
USA/EUROPE: Motorola Literature Distribution;
P.O. Box 20912; Phoenix, Arizona 85036. 1-800-441-2447
MFAX: RMFAX0@email.sps.mot.com - TOUCHTONE (602) 244-6609
INTERNET: http://Design-NET.com
JAPAN: Nippon Motorola Ltd.; Tatsumi-SPD-JLDC, Toshikatsu Otsuki,
6F Seibu-Butsuryu-Center, 3-14-2 Tatsumi Koto-Ku, Tokyo 135, Japan. 03-3521-8315
HONG KONG: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park,
51 Ting Kok Road, Tai Po, N.T., Hong Kong. 852-26629298

MOTOROLA

MPC505TS/D