68HC05PL4
68HC05PL4B
68HC705PL4
68HC705PL4B

SPECIFICATION
(General Release)

April 30, 1998

Consumer Systems Group
Semiconductor Products Sector

freescale™
semiconductor

For More Information On This Product,
Go to: www.freescale.com

TABLE OF CONTENTS

Section **Page**

SECTION 1
GENERAL DESCRIPTION

1.1 FEATURES 1-1
1.2 MCU BLOCK DIAGRAM 1-2
1.3 PIN ASSIGNMENTS 1-3
1.4 PIN DESCRIPTIONS 1-4
1.4.1 VDD, VSS 1-4
1.4.2 OSC1, OSC2 1-4
1.4.3 $\overline{\text{RESET}}$ 1-4
1.4.4 LED/$\overline{\text{IRQ}}$ 1-4
1.4.5 PA0, PA1/DTMF, PA2/TCAP, PA3/TCMP, PA4-PA6 1-5
1.4.6 PB0/KBI0-PB3/KBI3, PB4-PB7 1-5
1.4.7 PC0-PC7 1-6

SECTION 2
MEMORY

2.1 MEMORY MAP 2-1
2.2 I/O REGISTERS 2-2
2.3 RAM 2-2
2.4 ROM 2-2
2.5 COP WATCHDOG REGISTER (COPR) 2-2

SECTION 3
CENTRAL PROCESSING UNIT

3.1 REGISTERS 3-1
3.2 ACCUMULATOR (A) 3-2
3.3 INDEX REGISTER (X) 3-2
3.4 STACK POINTER (SP) 3-2
3.5 PROGRAM COUNTER (PC) 3-2
3.6 CONDITION CODE REGISTER (CCR) 3-3
3.6.1 Half Carry Bit (H-Bit) 3-3
3.6.2 Interrupt Mask (I-Bit) 3-3
3.6.3 Negative Bit (N-Bit) 3-3
3.6.4 Zero Bit (Z-Bit) 3-3
3.6.5 Carry/Borrow Bit (C-Bit) 3-4

SECTION 4
INTERRUPTS

4.1 INTERRUPT VECTORS 4-1
4.2 INTERRUPT PROCESSING 4-2
4.3 SOFTWARE INTERRUPT 4-4
4.4 EXTERNAL INTERRUPT 4-4
4.4.1 LED/$\overline{\text{IRQ}}$ Pin 4-4

TABLE OF CONTENTS

Section **Page**

4.4.2 Miscellaneous Control and Status Register 4-5
4.5 16-BIT TIMER INTERRUPTS 4-6
4.5.1 Input Capture Interrupt 4-6
4.5.2 Output Compare Interrupt 4-6
4.5.3 Timer Overflow Interrupt 4-6
4.6 8-BIT TIMER INTERRUPT 4-6
4.7 KEYBOARD INTERRUPT 4-7

SECTION 5
RESETS

5.1 POWER-ON RESET 5-1
5.2 EXTERNAL RESET 5-2
5.3 INTERNAL RESETS 5-2
5.3.1 Power-On Reset (POR) 5-3
5.3.2 Computer Operating Properly (COP) Reset 5-3
5.3.3 Illegal Address Reset 5-4
5.4 RESET STATES OF SUBSYSTEM IN MCU 5-5
5.4.1 CPU 5-5
5.4.2 I/O Registers 5-5
5.4.3 8-Bit Timer 5-5
5.4.4 16-Bit Programmable Timer 5-5
5.4.5 Keyboard Interrupt Interface 5-6
5.4.6 6-bit DAC Subsystem 5-6
5.4.7 System Clock Option Subsystem 5-6
5.4.8 Miscellaneous Subsystem 5-6
5.5 RESET CHARACTERISTICS 5-7

SECTION 6
OPERATING MODES

6.1 OPERATING MODES 6-1
6.1.1 Single-chip (Normal) Mode 6-1
6.1.2 Self-check Mode 6-1
6.2 LOW POWER MODES 6-2
6.2.1 STOP Mode 6-2
6.2.2 WAIT Mode 6-2

SECTION 7
INPUT/OUTPUT PORTS

7.1 PARALLEL PORTS 7-1
7.1.1 Port Data Registers 7-2
7.1.2 Port Data Direction Registers 7-2
7.2 PORT A 7-2
7.3 PORT B 7-3

TABLE OF CONTENTS

Section **Page**

7.4 PORT C 7-3
7.5 SUMMARY OF PORT A AND PORT B SHARED PINS 7-3

SECTION 8
SYSTEM CLOCKS

8.1 SYSTEM CLOCK SOURCE AND FREQUENCY OPTION 8-1

SECTION 9
16-BIT PROGRAMMABLE TIMER

9.1 TIMER REGISTERS (TMRH, TMRL) 9-2
9.2 ALTERNATE COUNTER REGISTERS (ACRH, ACRL) 9-4
9.3 INPUT CAPTURE REGISTERS 9-5
9.4 OUTPUT COMPARE REGISTERS 9-6
9.5 TIMER CONTROL REGISTER (TCR) 9-8
9.5.1 Miscellaneous Control and Status Register for Timer16 9-9
9.6 TIMER STATUS REGISTER (TSR) 9-10
9.7 16-BIT TIMER OPERATION DURING WAIT MODE 9-11
9.8 16-BIT TIMER OPERATION DURING STOP MODE 9-11

SECTION 10
8-BIT TIMER

10.1 OVERVIEW 10-1
10.2 TIMER8 CONTROL AND STATUS REGISTER (T8CSR) 10-2
10.3 TIMER8 COUNTER REGISTER (T8CNTR) 10-3
10.4 COMPUTER OPERATING PROPERLY (COP) WATCHDOG 10-3
10.5 8-BIT TIMER OPERATION DURING WAIT MODE 10-4
10.6 8-BIT TIMER OPERATION DURING STOP MODE 10-4

SECTION 11
DIGITAL TO ANALOG CONVERTER

11.1 DAC CONTROL AND DATA REGISTER 11-1
11.2 DAC OPERATION DURING WAIT MODE 11-1
11.3 DAC OPERATION DURING STOP MODE 11-1
11.4 DAC CHARACTERISTICS 11-2

SECTION 12
INSTRUCTION SET

12.1 ADDRESSING MODES 12-1
12.1.1 Inherent 12-1
12.1.2 Immediate 12-1
12.1.3 Direct 12-2
12.1.4 Extended 12-2
12.1.5 Indexed, No Offset 12-2
12.1.6 Indexed, 8-Bit Offset 12-2

TABLE OF CONTENTS

Section **Page**

12.1.7 Indexed, 16-Bit Offset 12-3
12.1.8 Relative 12-3
12.1.9 Instruction Types 12-3
12.1.10 Register/Memory Instructions 12-4
12.1.11 Read-Modify-Write Instructions 12-5
12.1.12 Jump/Branch Instructions 12-5
12.1.13 Bit Manipulation Instructions 12-7
12.1.14 Control Instructions 12-7
12.1.15 Instruction Set Summary 12-8

SECTION 13
ELECTRICAL SPECIFICATIONS

13.1 MAXIMUM RATINGS 13-1
13.2 OPERATING TEMPERATURE RANGE 13-1
13.3 THERMAL CHARACTERISTICS 13-1
13.4 SUPPLY CURRENT CHARACTERISTICS 13-2
13.5 DC ELECTRICAL CHARACTERISTICS (4V) 13-3
13.6 DC ELECTRICAL CHARACTERISTICS (2V) 13-4
13.7 CONTROL TIMING (4V) 13-5
13.8 CONTROL TIMING (2V) 13-5

SECTION 14
MECHANICAL SPECIFICATIONS

14.1 28-PIN PDIP (CASE 710) 14-1
14.2 28-PIN SOIC (CASE 751F) 14-1
14.3 28-PIN SSOP 14-2

APPENDIX A
MC68HC705PL4

A.1 INTRODUCTION A-1
A.2 MEMORY A-1
A.3 BOOTLOADER MODE A-1
A.4 EPROM PROGRAMMING A-1
A.4.1 EPROM Program Control Register (PCN) A-2
A.4.2 Programming Sequence A-3
A.5 EPROM PROGRAMMING SPECIFICATIONS A-3
A.6 SUPPLY CURRENT CHARACTERISTICS A-6

LIST OF FIGURES

Figure	Title	Page
1-1	MC68HC05PL4 Block Diagram	1-2
1-2	MC68HC05PL4 Pin Assignment	1-3
1-3	MC68HC05PL4B Pin Assignment	1-3
1-4	Oscillator Connections	1-4
1-5	Miscellaneous Control and Status Register (MICSR)	1-5
2-1	MC68HC05PL4 Memory Map	2-1
2-2	COP Watchdog Register (COPR)	2-2
2-3	I/O Registers $0000-$000F	2-3
2-4	I/O Registers $0010-$001F	2-4
3-1	MC68HC05 Programming Model	3-1
4-1	Interrupt Stacking Order	4-2
4-2	Interrupt Flowchart	4-3
4-3	External Interrupt Logic	4-5
4-4	Miscellaneous Control and Status Register (MICSR)	4-5
4-5	Pull-Up Enable Register (PUER)	4-7
4-6	Keyboard Interrupt Enable Register (KIER)	4-7
4-7	Keyboard Interrupt Flag Register (KIFR)	4-7
5-1	Reset Sources	5-1
5-2	Miscellaneous Control and Status Register (MICSR)	5-2
5-3	COP Watchdog Block Diagram	5-3
5-4	COP Watchdog Register (COPR)	5-3
5-5	Miscellaneous Control and Status Register (MICSR)	5-4
5-6	Stop Recovery Timing Diagram	5-7
5-7	Internal Reset Timing Diagram	5-8
6-1	STOP/WAIT Flowchart	6-3
7-1	Port Input/Output Circuitry	7-1
8-1	System Clock Control Register (SYSCR)	8-1
9-1	Programmable Timer Block Diagram	9-1
9-2	Timer Counter and Register Block Diagram	9-2
9-3	Programmable Timer Registers (TMRH, TMRL)	9-3
9-4	Alternate Counter Block Diagram	9-4
9-5	Alternate Counter Registers (ACRH, ACRL)	9-4
9-6	Timer Input Capture Block Diagram	9-5
9-7	Input Capture Registers (ICRH, ICRL)	9-6
9-8	Timer Output Compare Block Diagram	9-7
9-9	Output Compare Registers (OCRH, OCRL)	9-7
9-10	Timer Control Register (TCR)	9-8
9-11	Miscellaneous Control and Status Register (MISCR)	9-9
9-12	Timer Status Registers (TSR)	9-10
10-1	Timer8 Block Diagram	10-1
10-2	Timer8 Control and Status Register	10-2
10-3	Timer8 Counter Register	10-3
11-1	DAC Control and Data Register	11-1

LIST OF FIGURES

Figure	Title	Page
A-1	MC68HC705PL4B Memory Map	A-2
A-2	EPROM Programming Sequence	A-4
A-3	MC68HC705PL4 Pin Assignment	A-5
A-4	MC68HC705PL4B Pin Assignment	A-5

LIST OF TABLES

Table	Title	Page
1-1	MC68HC05PL4 and MC68HC05PL4B Differences	1-1
4-1	Vector Address for Interrupts and Reset	4-1
5-1	Reset Characteristics	5-7
6-1	Operation Mode Condition After Reset	6-1
7-1	I/O Pin Functions	7-2
7-2	Port A and Port B Shared Pins	7-3
8-1	System Clock Divider Select	8-1
8-2	System Clock Source Select	8-1
9-1	Output Compare Initialization Example	9-8
12-1	Register/Memory Instructions	12-4
12-2	Read-Modify-Write Instructions	12-5
12-3	Jump and Branch Instructions	12-6
12-4	Bit Manipulation Instructions	12-7
12-5	Control Instructions	12-7
12-6	Instruction Set Summary	12-8
12-7	Opcode Map	12-14
A-1	MC68HC705PL4 and MC68HC705PL4B Differences	A-1
A-2	EPROM Programming Electrical Characteristics	A-3

LIST OF TABLES

Table	Title	Page

SECTION 1
GENERAL DESCRIPTION

The MC68HC05PL4 HCMOS microcontroller is a member of the M68HC05 Family of low-cost single-chip microcontroller units (MCUs). This MCU is designed specically for the handset and base set of cost-sensitive CT0/1 analog cordless phones.

References to MC68HC05PL4 apply to both MC68HC05PL4 and MC68HC05PL4B, unless otherwise stated.

Table 1-1. MC68HC05PL4 and MC68HC05PL4B Differences

Device	Pin 27
MC68HC05PL4	PA0
MC68HC05PL4B	OSC2

1.1 FEATURES

- Industry standard 8-bit M68HC05 CPU core
- Bus frequency: 2.56MHz @ 4V and 1MHz @ 2V
- Built-in low-frequency RC oscillator (500kHz and 20kHz)
- OSC input pin (OSC output pin on MC68HC05PL4B)
- 256 bytes of user RAM
- 4k-bytes of user ROM
- ROM security
- 23 (22 for MC68HC05PL4B) bidirectional I/O lines with:
  - 4 keyboard interrupts with pull-up resistor
  - 6 high current sink pins
- Open-drain output for LED drive
- Multiplexed DTMF output with built-in 6-bit D/A
- 16-bit programmable timer with input capture and output compare functions
- Reloadable 8-bit event timer
- COP watchdog reset
- Power saving STOP and WAIT modes
- Available in 28-pin PDIP, SOIC, and SSOP packages

1.2 MCU BLOCK DIAGRAM

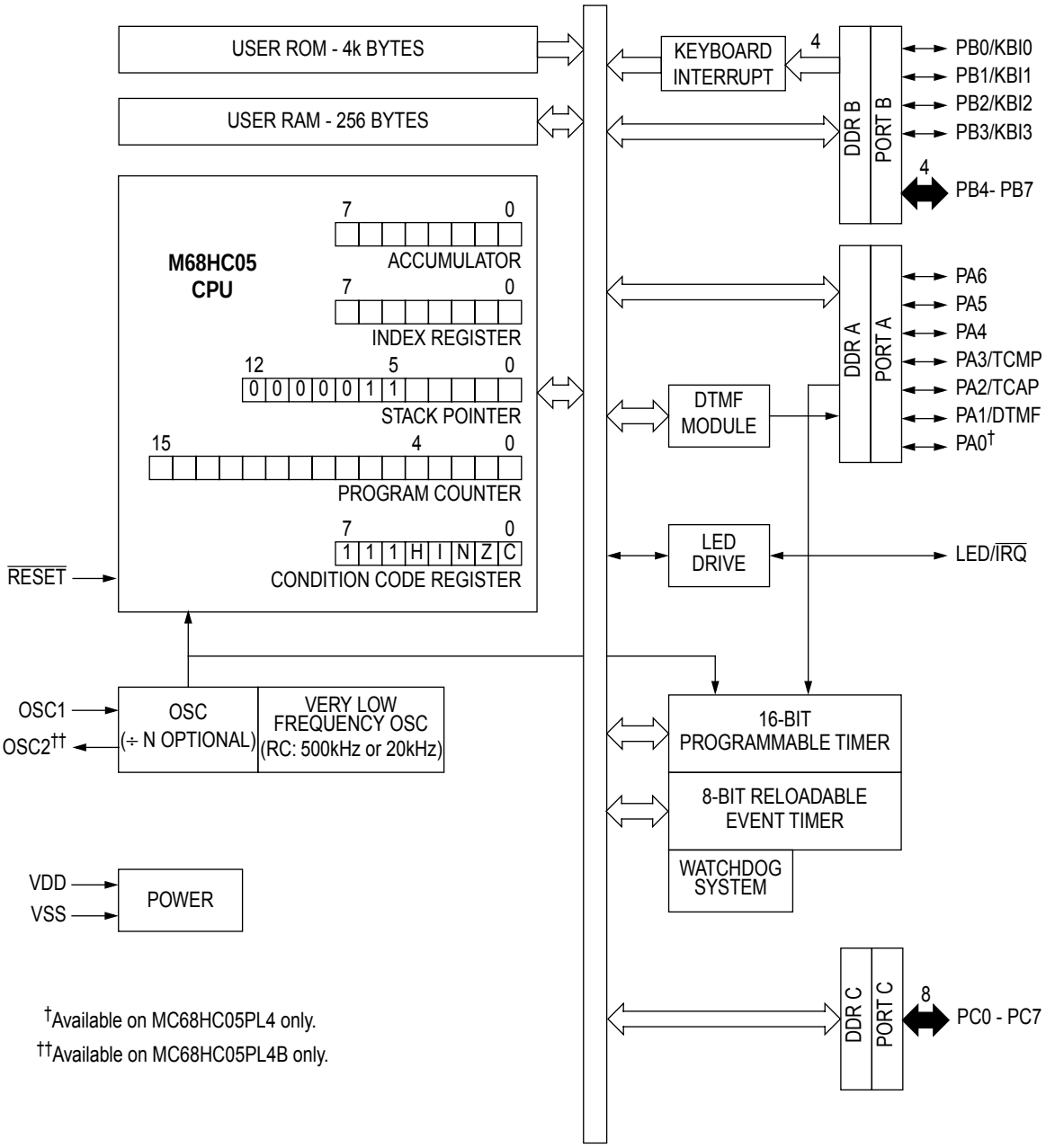


Figure 1-1. MC68HC05PL4 Block Diagram

NOTE

A line over a signal name indicates an active low signal. Any reference to voltage, current, or frequency speci ed in the following sections will refer to the nominal values. The exact values and their tolerance or limits are speci ed in Electrical Speci cations section.

1.3 PIN ASSIGNMENTS

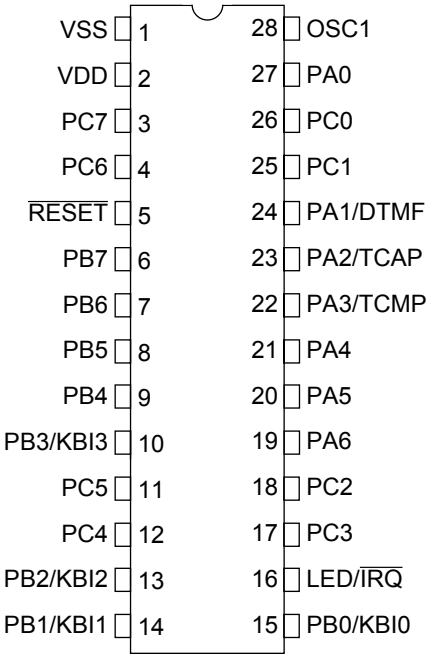


Figure 1-2. MC68HC05PL4 Pin Assignment

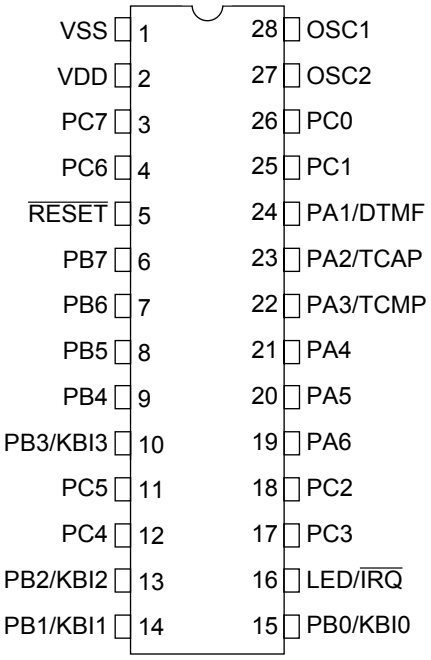


Figure 1-3. MC68HC05PL4B Pin Assignment

1.4 PIN DESCRIPTIONS

The following paragraphs give a description of each functional pin.

1.4.1 VDD, VSS

Power is supplied to the MCU using these pins. VDD is the positive supply and VSS is the ground pin.

1.4.2 OSC1, OSC2

OSC2 is only available on MC68HC05PL4B.

The OSC1 and OSC2 pins are the connections for the on-chip oscillator — the following con gur ations are available:

1. A crystal or ceramic resonator as shown in Figure 1-4(a).
2. An external clock signal as shown in Figure 1-4(b).

The external oscillator clock frequency, f_{OSC}, is divided by two to produce the internal operating frequency, f_{OP}.

(a) Crystal or Ceramic Resonator Connections

(c) External Clock Source Connection

Figure 1-4. Oscillator Connections

1.4.3 $\overline{\text{RESET}}$

This active low input-only pin is used to reset the MCU to a known start-up state. The $\overline{\text{RESET}}$ pin has an Schmitt trigger circuit as part of its input to improve noise immunity.

1.4.4 LED/$\overline{\text{IRQ}}$

This pin has two functions, con gured b y the IRQEN bit in the Miscellaneous Control and Status Register, at $1C (MISCR).

When this pin is $\overline{\text{IRQ}}$, it drives the asynchronous IRQ interrupt function of the CPU. The IRQ interrupt function uses the IRQS bit in the MISCR to provide either only negative edge-sensitive triggering or both negative edge-sensitive and low level-sensitive triggering. If the MISCR bit is set to enable level-sensitive

triggering, the LED/$\overline{IRQ}$ pin requires an external resistor to VDD for “wired-OR” operation. If the LED/$\overline{IRQ}$ is not used, it must be tied to the VDD supply. The contains an internal Schmitt trigger as part of its input to improve noise immunity.

When this pin is LED, the LED bit in the MISCR controls the on/off function of the connected LED. This LED pin sinks current via an internal pulldown resistor.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
MICSR	R	IRQEN	IRQS	TCMPEN	TCAPEN		LED	COPON	POR
$001C	W								
RESET		0	0	0	0	0	0	0	0

Figure 1-5. Miscellaneous Control and Status Register (MICSR)

IRQEN — External Interrupt Request Enable

0 = LED/$\overline{IRQ}$ pin con gured as LED dr ive pin.
1 = LED/$\overline{IRQ}$ pin con gured as $\overline{IRQ}$ input pin, for external interrupts.

LED — LED Drive Output Control

1 = Enable internal pulldown resistor, pin is logic low.
0 = Disable internal pulldown resistor, pin is in high impedance state.

1.4.5 PA0, PA1/DTMF, PA2/TCAP, PA3/TCMP, PA4-PA6

These eight I/O lines comprise port A, a general purpose bidirectional I/O port. The state of any pin is software programmable and all port B lines are con gured as inputs during power-on or reset.

PA0 is only available on MC68HC05PL4.

PA1 is shared with DTMF output of the DAC subsystem. This pin is con gured as an output pin for DTMF.

PA2 is shared with TCAP input of the 16-bit timer. This pin is con gured as an input pin for TCAP.

PA3 is shared with TCMP output of the 16-bit timer. This pin is con gured as an output pin for TCMP.

PA5 and PA6 have high current sinking capability; see Electrical Speci cations section for values.

1.4.6 PB0/KBI0-PB3/KBI3, PB4-PB7

These eight I/O lines comprise port B, a general purpose bidirectional I/O port. The state of any pin is software programmable and all port B lines are con gured as inputs during power-on or reset.

All port B pins have internal pullups which can be individually enabled by software.

PB0-PB3 also have keyboard interrupt capability, which can be individually enabled.

1.4.7 PC0-PC7

These eight I/O lines comprise port C, a general purpose bidirectional I/O port. The state of any pin is software programmable and all port C lines are con gured as inputs during power-on or reset.

PC4-PC7 have high current sinking capability; see Electrical Speci cations section for values.

SECTION 2
MEMORY

This section describes the organization of the memory on the MC68HC05PL4.

2.1 MEMORY MAP

The CPU can address 8k-bytes of memory space as shown in **Figure 2-1**. The ROM portion of the memory holds the program instructions, fixed data, user defined vectors, and interrupt service routines. The RAM portion of memory holds variable data. I/O registers are memory mapped so that the CPU can access their locations in the same way that it accesses all other memory locations.

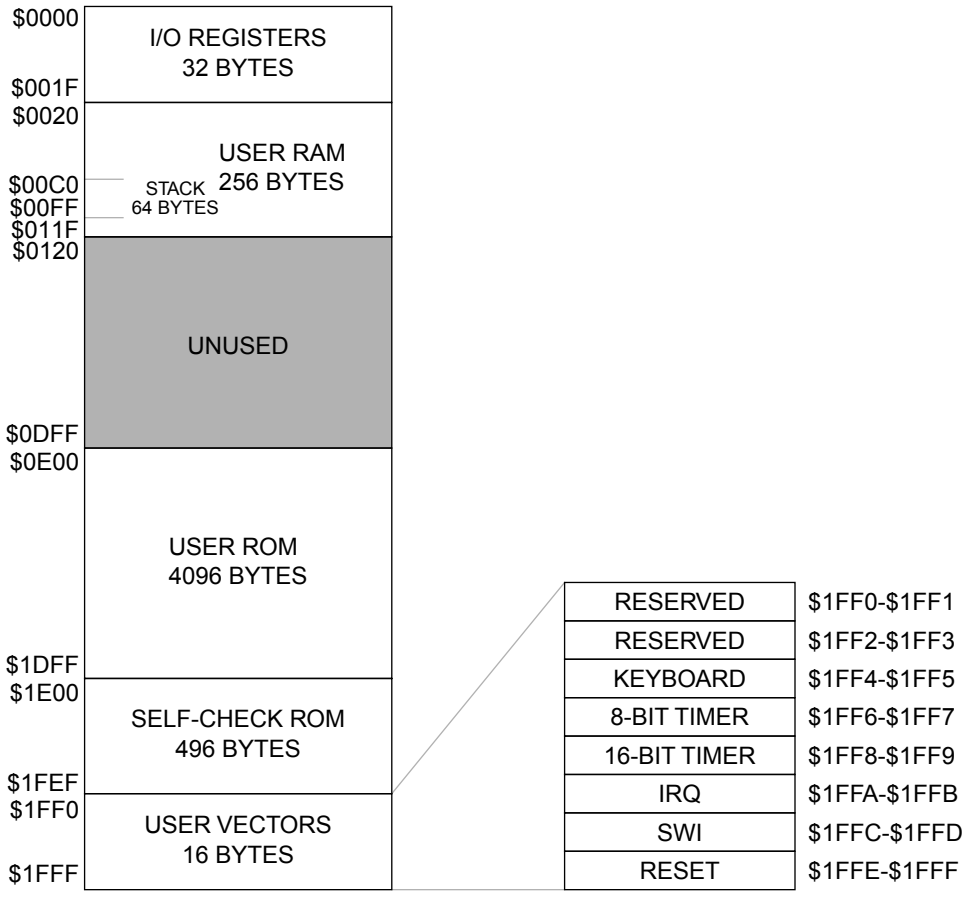


Figure 2-1. MC68HC05PL4 Memory Map

2.2 I/O REGISTERS

The first 32 addresses of the memory space, $0000-$001F, are the I/O section.

One I/O register is located outside the 32-byte I/O section, which is the Computer Operating Properly (COP) register mapped at $1FF0.

The bit assignment of each I/O register is described in the respective sections and summarized in **Figure 2-3** and **Figure 2-4**.

2.3 RAM

The 256 addresses from $0020 to $01FF serve as both user RAM and the stack RAM. The CPU uses five RAM bytes to save all CPU register contents before processing an interrupt. During a subroutine call, the CPU uses two bytes to store the return address. The stack pointer decrements during pushes and increments during pulls.

NOTE

Be careful when using nested subroutines or multiple interrupt levels. The CPU may overwrite data in the RAM during a subroutine or during the interrupt stacking operation.

2.4 ROM

The 4096 bytes of user ROM is located from address $0E00 to $1DFF.

Addresses $1FF0 to $1FFF contain 16 bytes of ROM reserved for user vectors.

2.5 COP WATCHDOG REGISTER (COPR)

Writing “0” to the COPC bit in the COP watchdog register ($1FF0) resets the COP watchdog timer. This is a write only register; writing a “1” to COPC has no effect.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
COPR	R								
$1FF0	W								COPC
RESET		U	U	U	U	U	U	U	U

Figure 2-2. COP Watchdog Register (COPR)

ADDR	REGISTER	ACCESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
$0000	Port A Data PORTA	R		PA6	PA5	PA4	PA3	PA2	PA1	PA0
		W								
$0001	Port B Data PORTB	R	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
		W								
$0002	Port C Data PORTC	R	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
		W								
$0003	RESERVED	R								
		W								
$0004	RESERVED	R								
		W								
$0005	Port A Data Direction DDRA	R		DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		W								
$0006	Port B Data Direction DDRB	R	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		W								
$0007	Port C Data Direction DDRC	R	DDRC7	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		W								
$0008	RESERVED	R								
		W								
$0009	RESERVED	R								
		W								
$000A	Pull-up Enable PUER	R	PUL7	PUL6	PUL5	PUL4	PUL3	PUL2	PUL1	PUL0
		W								
$000B	Keyboard Int. Enable KIER	R					KIE3	KIE2	KIE1	KIE0
		W								
$000C	Keyboard Int. Flag KIFR	R					KIF3	KIF2	KIF1	KIF0
		W								
$000D	Timer 8 Ctrl/Status T8CSR	R	T8IF	0	T8IE		T8EN	PS2	PS1	PS0
		W		T8IFR						
$000E	Timer 8 Counter T8CNTR	R	T8CNT7	T8CNT6	T8CNT5	T8CNT4	T8CNT3	T8CNT2	T8CNT1	T8CNT0
		W								
$000F	DAC Ctrl and Data DACDR	R	DACEN		DA5	DA4	DA3	DA2	DA1	DA0
		W								

Figure 2-3. I/O Registers $0000-$000F

ADDR	REGISTER	ACCESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
$0010	RESERVED	R								
		W								
$0011	RESERVED	R								
		W								
$0012	Timer Control TCR	R	ICIE	OCIE	TOIE				IEDG	OLVL
		W								
$0013	Timer Status TSR	R	ICF	OCF	TOF					
		W								
$0014	Input Capture High ICRH	R	ICRH7	ICRH6	ICRH5	ICRH4	ICRH3	ICRH2	ICRH1	ICRH0
		W								
$0015	Input Capture Low ICRL	R	ICRL7	ICRL6	ICRL5	ICRL4	ICRL3	ICRL2	ICRL1	ICRL0
		W								
$0016	Output Compare High OCRH	R	OCRH7	OCRH6	OCRH5	OCRH4	OCRH3	OCRH2	OCRH1	OCRH0
		W								
$0017	Output Compare Low OCRL	R	OCRL7	OCRL6	OCRL5	OCRL4	OCRL3	OCRL2	OCRL1	OCRL0
		W								
$0018	Timer Counter High TMRH	R	TMRH7	TMRH6	TMRH5	TMRH4	TMRH3	TMRH2	TMRH1	TMRH0
		W								
$0019	Timer Counter Low TMRL	R	TMRL7	TMRL6	TMRL5	TMRL4	TMRL3	TMRL2	TMRL1	TMRL0
		W								
$001A	Alt. Counter High ACRH	R	ACRH7	ACRH6	ACRH5	ACRH4	ACRH3	ACRH2	ACRH1	ACRH0
		W								
$001B	Alt. Counter Low ACRL	R	ACRL7	ACRL6	ACRL5	ACRL4	ACRL3	ACRL2	ACRL1	ACRL0
		W								
$001C	Misc. Control/Status MICSR	R	IRQEN	IRQS	TCMPEN	TCAPEN		LED	COPON	POR
		W								
$001D	System Clock Control SYSCR	R	SYSDIV1	SYSDIV2	CKSEL1	CKSEL2	FMODE	OSCF	RCF	CKOSC
		W								
$001E	RESERVED	R								
		W								
$001F	RESERVED	R								
		W								

Figure 2-4. I/O Registers $0010-$001F

SECTION 3
CENTRAL PROCESSING UNIT

The MC68HC05PL4 has an 8k-bytes memory map. The stack has only 64 bytes. Therefore, the stack pointer has been reduced to only 6 bits and will only decrement down to $00C0 and then wrap-around to $00FF. All other instructions and registers behave as described in this chapter.

3.1 REGISTERS

The MCU contains five registers which are hard-wired within the CPU and are not part of the memory map. These five registers are shown in **Figure 3-1** and are described in the following paragraphs.

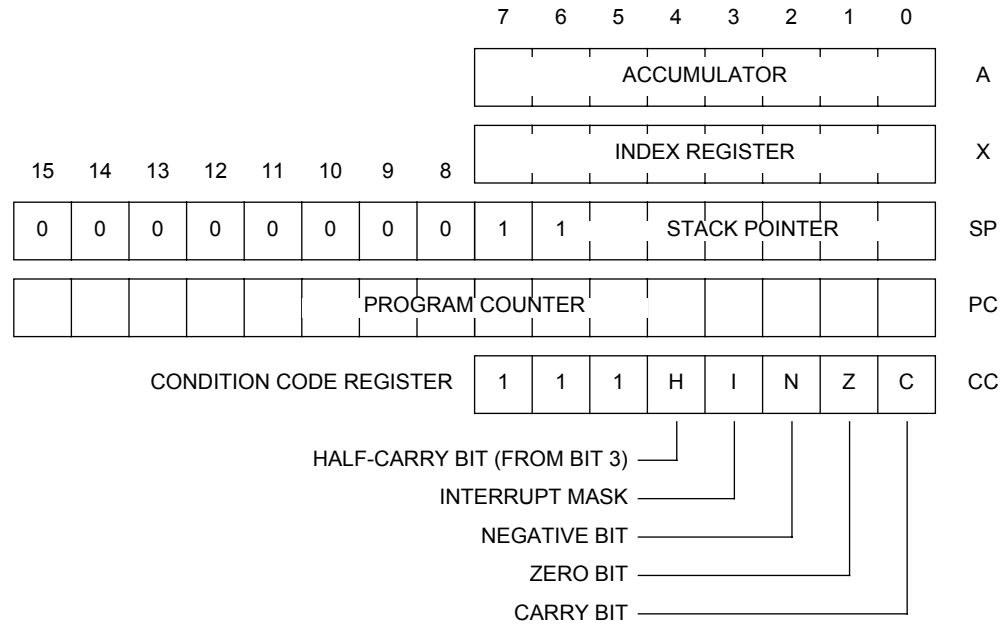


Figure 3-1. MC68HC05 Programming Model

3.2 ACCUMULATOR (A)

The accumulator is a general purpose 8-bit register as shown in **Figure 3-1**. The CPU uses the accumulator to hold operands and results of arithmetic calculations or non-arithmetic operations. The accumulator is not affected by a reset of the device.

3.3 INDEX REGISTER (X)

The index register shown in **Figure 3-1** is an 8-bit register that can perform two functions:

- Indexed addressing
- Temporary storage

In indexed addressing with no offset, the index register contains the low byte of the operand address, and the high byte is assumed to be $00. In indexed addressing with an 8-bit offset, the CPU nds the oper and address by adding the index register content to an 8-bit immediate value. In indexed addressing with a 16-bit offset, the CPU nds the operand address by adding the index register content to a 16-bit immediate value.

The index register can also serve as an auxiliary accumulator for temporary storage. The index register is not affected by a reset of the device.

3.4 STACK POINTER (SP)

The stack pointer shown in **Figure 3-1** is a 16-bit register. In MCU devices with memory space less than 64k-bytes the unimplemented upper address lines are ignored. The stack pointer contains the address of the next free location on the stack. During a reset or the reset stack pointer (RSP) instruction, the stack pointer is set to $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled off the stack.

When accessing memory, the ten most signi cant bits are permanently set to 0000000011. The six least signi cant register bits are appended to these ten x ed bits to produce an address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64($C0) locations. If 64 locations are exceeded, the stack pointer wraps around and overwrites the previously stored information. A subroutine call occupies two locations on the stack and an interrupt uses ve locations.

3.5 PROGRAM COUNTER (PC)

The program counter shown in **Figure 3-1** is a 16-bit register. In MCU devices with memory space less than 64k-bytes the unimplemented upper address lines are ignored. The program counter contains the address of the next instruction or operand to be fetched.

Normally, the address in the program counter increments to the next sequential memory location every time an instruction or operand is fetched. Jump, branch, and interrupt operations load the program counter with an address other than that of the next sequential location.

3.6 CONDITION CODE REGISTER (CCR)

The CCR shown in **Figure 3-1** is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. The fth bit is the interrupt mask. These bits can be individually tested by a program, and speci c actions can be taken as a result of their states. The condition code register should be thought of as having three additional upper bits that are always ones. Only the interrupt mask is affected by a reset of the device. The following paragraphs explain the functions of the lower v e bits of the condition code register.

3.6.1 Half Carry Bit (H-Bit)

When the half-carry bit is set, it means that a carry occurred between bits 3 and 4 of the accumulator during the last ADD or ADC (add with carry) operation. The half-carry bit is required for binary-coded decimal (BCD) arithmetic operations.

3.6.2 Interrupt Mask (I-Bit)

When the interrupt mask is set, the internal and external interrupts are disabled. Interrupts are enabled when the interrupt mask is cleared. When an interrupt occurs, the interrupt mask is automatically set after the CPU registers are saved on the stack, but before the interrupt vector is fetched. If an interrupt request occurs while the interrupt mask is set, the interrupt request is latched. Normally, the interrupt is processed as soon as the interrupt mask is cleared.

A return from interrupt (RTI) instruction pulls the CPU registers from the stack, restoring the interrupt mask to its state before the interrupt was encountered. After any reset, the interrupt mask is set and can only be cleared by the Clear I-Bit (CLI), or WAIT instructions.

3.6.3 Negative Bit (N-Bit)

The negative bit is set when the result of the last arithmetic operation, logical operation, or data manipulation was negative. (Bit 7 of the result was a logical one.)

The negative bit can also be used to check an often tested ag b y assigning the ag to bit 7 of a register or memor y location. Loading the accumulator with the contents of that register or location then sets or clears the negative bit according to the state of the ag.

3.6.4 Zero Bit (Z-Bit)

The zero bit is set when the result of the last arithmetic operation, logical operation, data manipulation, or data load operation was zero.

3.6.5 Carry/Borrow Bit (C-Bit)

The carry/borrow bit is set when a carry out of bit 7 of the accumulator occurred during the last arithmetic operation, logical operation, or data manipulation. The carry/borrow bit is also set or cleared during bit test and branch instructions and during shifts and rotates. This bit is neither set by an INC nor by a DEC instruction.

SECTION 4
INTERRUPTS

The CPU can be interrupted by ve different sources – one software and four hardware:

- Non-maskable Software Interrupt Instruction (SWI)
- External Asynchronous Interrupt ($\overline{IRQ}$)
- 16-Bit Timer
- 8-Bit Timer
- Keyboard Interrupt

4.1 INTERRUPT VECTORS

Table 4-1 summarizes the reset and interrupt sources and vector assignments

Table 4-1. Vector Address for Interrupts and Reset

Function	Source	Local Mask	Global Mask	Priority (1=Highest)	Vector Address
Reset	Power-On Logic	None	None	1	$1FFE-$1FFF
	$\overline{RESET}$ Pin	None			
	COP Watchdog	COPON[1]			
SWI	User Code	None	None	Same Priority As Instruction	$1FFC-$1FFD
External $\overline{IRQ}$	$\overline{IRQ}$ Pin	IRQEN	I Bit	2	$1FFA-$1FFB
16-Bit Timer	ICF Bit	ICIE	I Bit	3	$1FF8-$1FF9
	TCF Bit	TCIE			
	OCF Bit	OCIE			
8-Bit Timer	T8IF Bit	T8IE	I Bit	4	$1FF6-$1FF7
Keyboard	KIF3 Bit	KIE3	I Bit	5	$1FF4-$1FF5
	KIF2 Bit	KIE2			
	KIF1 Bit	KIE1			
	KIF0 Bit	KIE0			
Reserved	—	—	—	—	$1FF2-$1FF3
Reserved	—	—	—	—	$1FF0-$1FF1

NOTES:
1. COPON enables/disables the COP watchdog timer.

NOTE

If more than one interrupt request is pending, the CPU fetches the vector of the higher priority interrupt rst. A higher priority interrupt does not actually interrupt a lower priority interrupt service routine unless the lower priority interrupt service routine clears the I bit.

4.2 INTERRUPT PROCESSING

The CPU does the following actions to begin servicing an interrupt:

- Stores the CPU registers on the stack in the order shown in **Figure 4-1**
- Sets the I bit in the condition code register to prevent further interrupts
- Loads the program counter with the contents of the appropriate interrupt vector locations as shown in **Table 4-1**

The return from interrupt (RTI) instruction causes the CPU to recover its register contents from the stack as shown in **Figure 4-1**. The sequence of events caused by an interrupt is shown in the o w chart in **Figure 4-2**

$0020	(Bottom of RAM)		
$0021			
$00BE			
$00BF			
$00C0	(Bottom of Stack)		
$00C1			
$00C2			Unstacking
			Order
			⇓
n	Condition Code Register	5	1
n+1	Accumulator	4	2
n+2	Index Register	3	3
n+3	Program Counter (High Byte)	2	4
n+4	Program Counter (Low Byte)	1	5
		⇑	
		Stacking	
$00FD		Order	
$00FE			
$00FF	Top of Stack (RAM)		

Figure 4-1. Interrupt Stacking Order

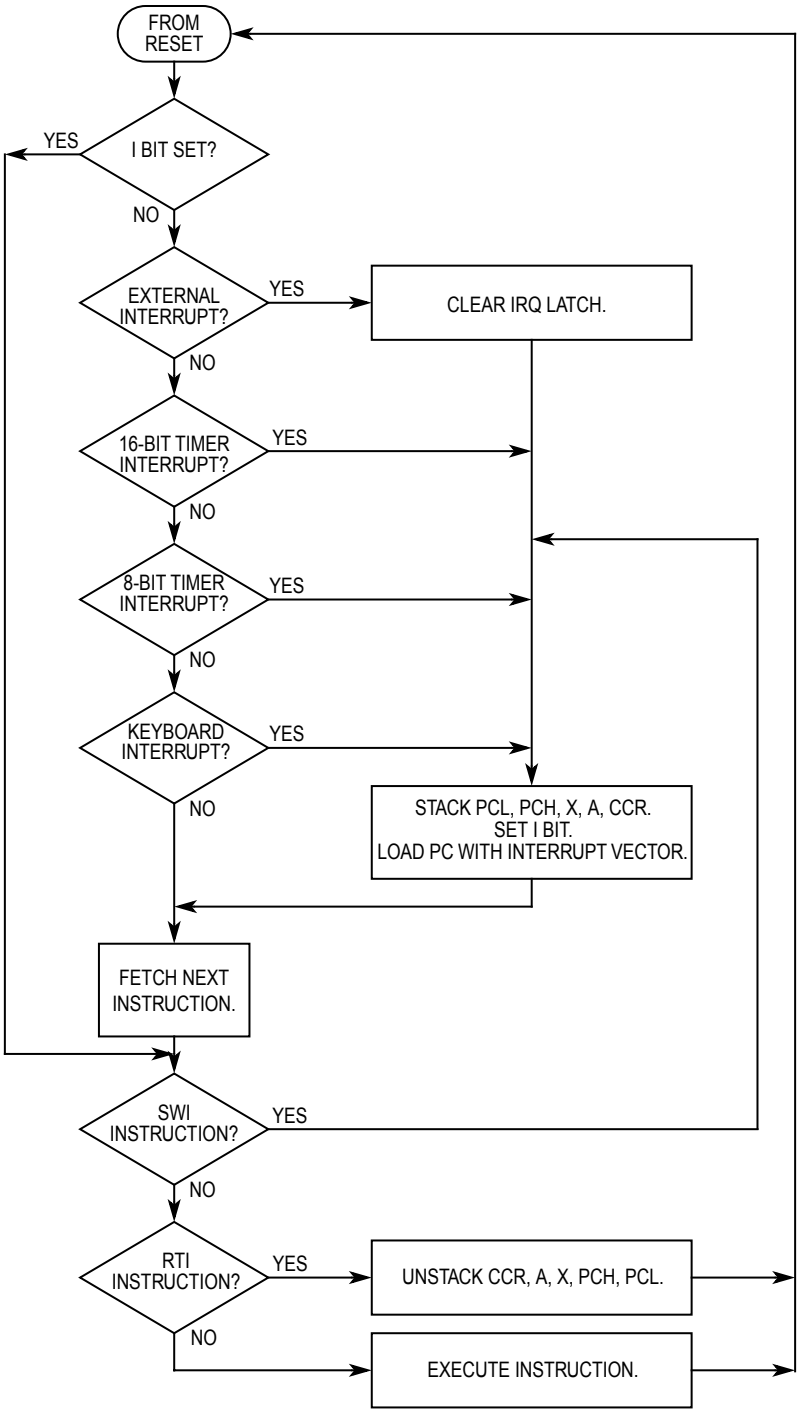


Figure 4-2. Interrupt Flowchart

4.3 SOFTWARE INTERRUPT

The software interrupt (SWI) instruction causes a non-maskable interrupt.

4.4 EXTERNAL INTERRUPT

The LED/$\overline{IRQ}$ pin is the source that generates external interrupt. Setting the I bit in the condition code register or clearing the IRQEN bit in the miscellaneous control/ status register disables this external interrupt.

4.4.1 LED/$\overline{IRQ}$ Pin

This pin is an open drain pin and setting the IRQEN bit in Miscellaneous Control/ Status Register (MICSR) will set this pin for external interrupt input pin.

An interrupt signal on the LED/$\overline{IRQ}$ pin latches an external interrupt request. To help clean up slow edges, the input from the LED/$\overline{IRQ}$ pin is processed by a Schmitt trigger gate. When the CPU completes its current instruction, it tests the IRQ latch. If the IRQ latch is set, the CPU then tests the I bit in the condition code register and the IRQEN bit in the MICSR. If the I bit is clear and the IRQEN bit is set, then the CPU begins the interrupt sequence. The CPU clears the IRQ latch while it fetches the interrupt vector, so that another external interrupt request can be latched during the interrupt service routine. As soon as the I bit is cleared during the return from interrupt, the CPU can recognize the new interrupt request. **Figure 4-3** shows the logic for external interrupts.

The LED/$\overline{IRQ}$ pin can be negative edge-triggered only or negative edge- and low-level-triggered. External interrupt sensitivity is programmed with the IRQS bit.

With the edge- and level-sensitive trigger option, a falling edge or a low level on the LED/$\overline{IRQ}$ pin latches an external interrupt request. The edge- and level-sensitive trigger option allows connection to the LED/$\overline{IRQ}$ pin of multiple wired-OR interrupt sources. As long as any source is holding the LED/$\overline{IRQ}$ low, an external interrupt request is present, and the CPU continues to execute the interrupt service routine.

With the edge-sensitive-only trigger option, a falling edge on the LED/$\overline{IRQ}$ pin latches an external interrupt request. A subsequent interrupt request can be latched only after the voltage level on the LED/$\overline{IRQ}$ pin returns to a logic one and then falls again to logic zero.

NOTE

To use the external interrupt function to exit from WAIT or STOP, it must be enabled prior entering either of the power saving modes.

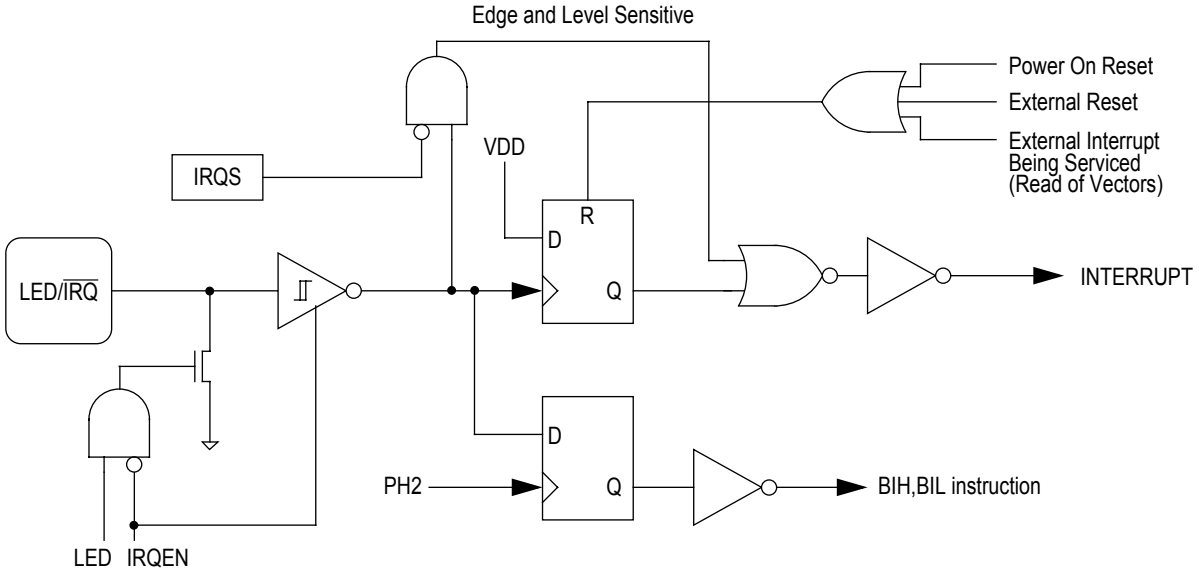


Figure 4-3. External Interrupt Logic

4.4.2 Miscellaneous Control and Status Register

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
MICSR	R	IRQEN	IRQS	TCMPEN	TCAPEN		LED	COPON	POR
$001C	W								
RESET		0	0	0	0	0	0	0	0

Figure 4-4. Miscellaneous Control and Status Register (MICSR)

IRQEN — External Interrupt Request Enable

This read/write bit enables external interrupts. Reset clears the IRQEN bit.

0 = External interrupt processing disabled. LED/$\overline{IRQ}$ pin return to normal LED function

1 = External interrupt processing enabled. LED/$\overline{IRQ}$ pin set to IRQ function

IRQS— External Interrupt Sensitivity

This bit makes the external interrupt inputs level-triggered as well as edge-triggered.

0 = IRQ negative edge-triggered and low level-triggered.

1 = IRQ negative edge-triggered only.

4.5 16-BIT TIMER INTERRUPTS

The 16-bit programmable Timer can generate an interrupt whenever the following events occur:

- Input capture
- Output compare
- Timer counter over o w

Setting the I bit in the condition code register disables Timer interrupts. The controls for these interrupts are in the Timer control register (TCR) located at $0012 and in the status bits are in the Timer status register (TSR) located at $0013.

The 16-bit programmable Timer interrupts can wake up MCU from WAIT Mode.

4.5.1 Input Capture Interrupt

An input capture interrupt occurs if the input capture ag (ICF) becomes set while the input capture interrupt enable bit (ICIE) is also set. The ICF ag bit is in the TSR; and the ICIE enable bit is located in the MICSR. The ICF ag bit is cleared by a read of the TSR with the ICF ag bit is set; and then followed by a read of the LSB of the input capture register (ICRL) or by reset. The ICIE enable bit is unaffected by reset.

4.5.2 Output Compare Interrupt

An output compare interrupt occurs if the output compare ag (OCF) becomes set while the output compare interrupt enable bit (OCIE) is also set. The OCF ag bit is in the TSR and the OCIE enable bit is in the MICSR. The OCF ag bit is cleared by a read of the TSR with the OCF ag bit set; and then followed by an access to the LSB of the output compare register (OCRL) or by reset. The OCIE enable bit is unaffected by reset.

4.5.3 Timer Overflow Interrupt

A Timer over ow interrupt occurs if the Timer over ow ag (TOF) becomes set while the Timer over o w interrupt enable bit (TOIE) is also set. The TOF ag bit is in the TSR and the TOIE enable bit is in the TCR. The TOF ag bit is cleared b y a read of the TSR with the TOF ag bit set; and then followed by an access to the LSB of the timer registers (TMRL) or by reset. The TOIE enable bit is unaffected by reset.

4.6 8-BIT TIMER INTERRUPT

The 8-bit Timer can generate an interrupt when the Timer8 Counter Register (T8CNTR) decrements from preset value to zero and the interrupt enable bit is set. Setting the I bit in the condition code register disables this Timer interrupts. The control bit for this interrupt and status bit are in the Timer 8 control register (T8CSR) located at $000D.

The 8-Bit timer interrupt can wake up MCU from WAIT Mode.

4.7 KEYBOARD INTERRUPT

Port B has internal pull-up resistors (typically 100KΩ) and are enabled individually by setting the corresponding bit in the Pull-Up Enable Register (PUER).

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
PUER	R	PUL7	PUL6	PUL5	PUL4	PUL3	PUL2	PUL1	PUL0
$000A	W								
RESET		0	0	0	0	0	0	0	0

Figure 4-5. Pull-Up Enable Register (PUER)

PB0 to PB3 have keyboard interrupt functions, with individual enable and flag bits in registers $000B and $000C.

A falling edge on any one of the keyboard interrupt pins sets the corresponding KIF flag in the Keyboard Interrupt Flag Register (KIFR) located at $000C. If the associated KIE bit in the Keyboard Interrupt Enable Register (KIER) located at $000B is also set, a keyboard interrupt is generated to the processor.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
KIER	R					KIE3	KIE2	KIE1	KIE0
$000B	W								
RESET		0	0	0	0	0	0	0	0

Figure 4-6. Keyboard Interrupt Enable Register (KIER)

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
KIFR	R					KIF3	KIF2	KIF1	KIF0
$000C	W								
RESET		0	0	0	0	0	0	0	0

Figure 4-7. Keyboard Interrupt Flag Register (KIFR)

KIFx can be cleared by writing “1” to the bit. Resets clear both KIFR and KIER.

Keyboard Interrupt can wake up the MCU from WAIT mode or STOP mode.

NOTE

Since the Keyboard Interrupt function is associated with PB0-PB3, any falling edge on these pins sets the corresponding KIF flag in the Keyboard Interrupt Flag Register. Therefore, PB0-PB3 should be connected to internal or external pull-ups, and KIFR cleared before these port pins switch from I/O to keyboard application.
To use the keyboard interrupt function to exit from WAIT or STOP, it must be enabled prior entering either of the power saving modes.

SECTION 5
RESETS

This section describes the four reset sources and how they initialize the MCU. A reset immediately stops the operation of the instruction being executed, initializes certain control bits, and loads the program counter with a user de ned reset v ector address. The following conditions produce a reset:

- Initial power-up of device (power-on reset)
- A logic zero applied to the $\overline{\text{RESET}}$ pin (external reset)
- Time-out of the COP watchdog (COP reset)
- Fetch of an opcode from an address not in the memory map (illegal address reset)

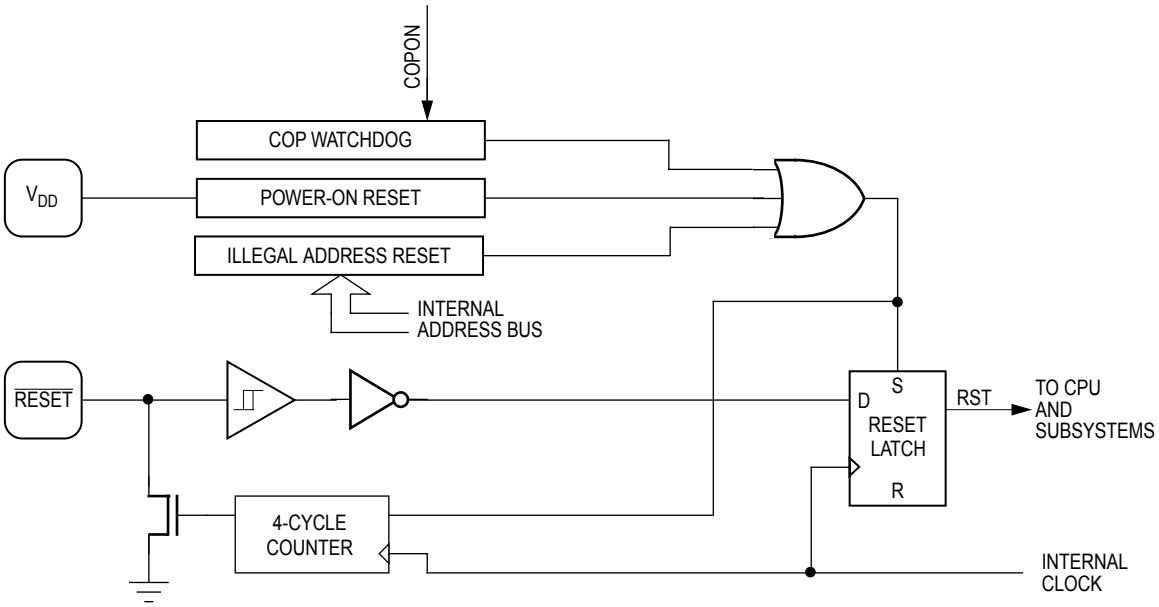


Figure 5-1. Reset Sources

5.1 POWER-ON RESET

A positive transition on the V_{DD} pin generates a power on reset. The power-on reset is strictly for conditions during powering up and cannot be used to detect drops in power supply voltage.

A 4064 t_{CYC} (internal clock cycle) delay after the oscillator becomes active allows the clock generator to stabilize. If the $\overline{\text{RESET}}$ pin is at logic zero at the end of the multiple t_{CYC} time, the MCU remains in the reset condition until the signal on the $\overline{\text{RESET}}$ pin goes to a logic one.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
MICSR	R	IRQEN	IRQS	TCMPEN	TCAPEN		LED	COPON	POR
$001C	W								
RESET		0	0	0	0	0	0	0	0

Figure 5-2. Miscellaneous Control and Status Register (MICSR)

POR - Power on Reset Flag

The POR bit is set each time the device is powered on. It allows the user to make a software distinction between a power-on and an external reset. POR can be cleared by software by writing a '0' to the bit. It cannot be set by software.

5.2 EXTERNAL RESET

A logic zero applied to the $\overline{\text{RESET}}$ pin for $1.5t_{CYC}$ generates an external reset. This pin is connected to a Schmitt trigger input gate to provide an upper and lower threshold voltage separated by a minimum amount of hysteresis. The external reset occurs whenever the $\overline{\text{RESET}}$ pin is pulled below the lower threshold and remains in reset until the $\overline{\text{RESET}}$ pin rises above the upper threshold. This active low input will generate the internal RST signal that resets the CPU and peripherals.

The $\overline{\text{RESET}}$ pin can also act as an open drain output. It will be pulled to a low state by an internal pulldown device that is activated by three internal reset sources. This RESET pulldown device will only be asserted for 3-4 cycles of the internal clock, f_{OP}, or as long as the internal reset source is asserted. When the external $\overline{\text{RESET}}$ pin is asserted, the pulldown device will not be turned on.

NOTE

Do not connect the $\overline{\text{RESET}}$ pin directly to V_{DD}, as this may overload some power supply designs when the internal pulldown on the $\overline{\text{RESET}}$ pin activates.

5.3 INTERNAL RESETS

The four internally generated resets are the initial power-on reset function, the COP Watchdog timer reset, the low voltage reset, and the illegal address detector. Only the COP Watchdog timer reset, low voltage reset and illegal address detector will also assert the pulldown device on the $\overline{\text{RESET}}$ pin for the duration of the reset function or 3-4 internal clock cycles, whichever is longer.

5.3.1 Power-On Reset (POR)

The internal POR is generated on power-up to allow the clock oscillator to stabilize. The POR is strictly for power turn-on conditions and is not able to detect a drop in the power supply voltage (brown-out). There is an oscillator stabilization delay of 4064 internal processor bus clock cycles after the oscillator becomes active.

The POR will generate the RST signal which will reset the CPU. If any other reset function is active at the end of the 4096 cycle delay, the RST signal will remain in the reset condition until the other reset condition(s) end.

POR will not activate the pulldown device on the $\overline{\text{RESET}}$ pin. V_{DD} must drop below V_{POR} in order for the internal POR circuit to detect the next rise of V_{DD}.

5.3.2 Computer Operating Properly (COP) Reset

The COP watchdog system consist of a divide by 8 counter with clock source from the 8-bit Timer (Timer8). Hence, a COP watchdog time-out occurs on the 8th Timer8 clock pulse. A COP watchdog time-out generates a COP reset to the CPU. **Figure 5-3** shows a block diagram of the COP watchdog logic.

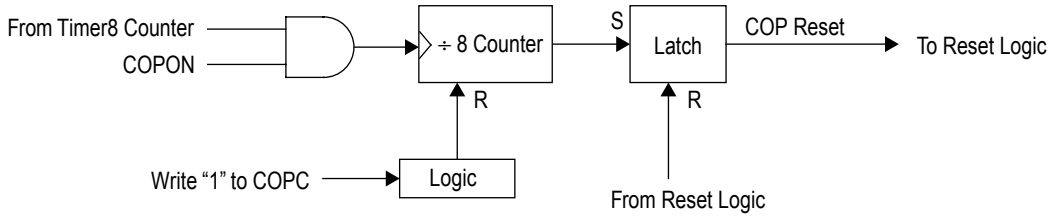


Figure 5-3. COP Watchdog Block Diagram

The COP watchdog is part of a software error detection system and must be cleared periodically to start a new time-out period. To clear the COP watchdog and prevent a COP reset, write a logic "1" to the COPC bit in the COP register at location $1FF0. The COP register, shown in **Figure 5-4**, is a write-only register that returns the content of a ROM location when read.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
COPR	R								
$1FF0	W								COPC
RESET		U	U	U	U	U	U	U	U

Figure 5-4. COP Watchdog Register (COPR)

COPC — COP Clear

COPC is a write-only bit. Periodically writing a logic one to COPC prevents the COP watchdog from resetting the MCU. Reset clears the COPC bit.

1 = Reset COP watchdog timer.
0 = No effect on COP watchdog timer.

Use the following formula to calculate the COP time-out period:

$$\text{COP Time-out Period} = (\text{prescaler} \times 256 \times 8) \div f_{BUS}$$

where prescaler is the Timer8 prescaler value

The clock input to the watchdog system is derived from the output of the Timer8, therefore a reset or preset of Timer8 may affect the COP watchdog time-out period.

The COP Watchdog reset will assert the pulldown device to pull the $\overline{\text{RESET}}$ pin low for 3-4 cycles of the internal bus clock.

The COP reset can be enable or disable by the COPON bit in MISCR. The MISCR is in **Figure 5-5**.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
MICSR	R	IRQEN	IRQS	TCMPEN	TCAPEN		LED	COPON	POR
$001C	W								
RESET		0	0	0	0	0	0	0	0

Figure 5-5. Miscellaneous Control and Status Register (MICSR)

COPON — COP On

Since the COP Watchdog system is derived from the 8-bit Timer system, the T8EN bit in the Timer8 Control and Status register (bit3 of $0D) must be set for COPON bit to have any affect.

COPON can be set to enable the COP watchdog system. Once set, the watchdog system cannot be disabled other than by a power-on reset or external reset. After a reset the COPON bit is cleared and the COP watchdog system is disabled.

1 = COP Watchdog enabled.
0 = COP Watchdog disabled.

NOTE

The COP Watchdog system is not designed to operate in STOP mode, therefore it should be disabled before entering STOP mode by clearing the COPON bit. Entering STOP mode with COP watchdog enabled will cause an internal reset of the MCU.

5.3.3 Illegal Address Reset

An opcode fetch from an address that is not in the ROM (locations $0E00–$1DFF and $1FF0-$1FFF) or the RAM (locations $0020–$011F) generates an illegal address reset. The illegal address reset will assert the pulldown device to pull the $\overline{\text{RESET}}$ pin low for 3-4 cycles of the internal bus clock.

5.4 RESET STATES OF SUBSYSTEM IN MCU

The following paragraphs describe how a reset initializes various sub-systems.

5.4.1 CPU

A reset has the following effects on the CPU:

- Loads the stack pointer with $FF.
- Sets the I bit in the condition code register, inhibiting interrupts.
- Loads the program counter with the user de ned reset vector from locations $1FFE and $1FFF.
- Clears the stop latch, enabling the CPU clock.
- Clears the wait latch, bringing the CPU out of the wait mode.

5.4.2 I/O Registers

A reset has the following effects on I/O registers:

- Clears bits in data direction registers con gur ing pins as inputs:
  - DDRA6–DDRA0 in DDRA for port A.
  - DDRB7–DDRB0 in DDRB for port B.
  - DDRC7–DDRC0 in DDRC for port C.
- Has no effect on port A, B, C data registers.

5.4.3 8-Bit Timer

A reset has the following effects on the 8-Bit Timer:

- Timer 8 system disabled (T8EN bit cleared)
- Timer 8 interrupt request disabled
- Timer 8 Pre-scalar preset to divide the internal bus clock by ratio 16
- Timer 8 Counter register preset to $FF

Therefore disables the timer 8 interrupt and preset the counter for POR cycle delay.

5.4.4 16-Bit Programmable Timer

A reset has the following effects on the 16-bit programmable Timer:

- Initializes the timer counter registers (TMRH, TMRL) to a value of $FFFC.
- Initializes the alternate timer counter registers (ACRH, ACRL) to a value of $FFFC.
- Clears all the interrupt enables and the output level bit (OLVL) in the timer control register (TCR).

- Does not affect the input capture edge bit (IEDG) in the TCR.
- Does not affect the interrupt ags in the timer status register (TSR).
- Does not affect the input capture registers (ICRH, ICRL).
- Does not affect the output compare registers (OCRH, OCRL).

Therefore con gure the por t A pins PA2,PA3 as general I/O function. However the timer is free running for interrupt process.

5.4.5 Keyboard Interrupt Interface

A reset has the following effects on the Keyboard Interrupt interface:

- Clears all bits in Keyboard interrupt enable register (KIER) and Keyboard interrupt disable
- Clears all bits in Keyboard interrupt ag register (KIFR)
- Clears all bits in Pull-Up enable register (PUER)

Therefore disables the Keyboard interrupt and leaves the shared port B pins as general I/O. Any pending interrupt ag is cleared and the K eyboard interrupt is disabled.

5.4.6 6-bit DAC Subsystem

A reset has the following effects on the DAC subsystem:

- Clears all bits in DAC control Register, hence DAC subsystem is disabled.

Therefore con gure the por t A pin PA1 as general I/O function.

5.4.7 System Clock Option Subsystem

At reset has the following effects on OSC clock subsystem

- The internal RC is enabled and oscillating at around 500kHz
- Internal clock divider selected to divide by 2 for bus frequency

5.4.8 Miscellaneous Subsystem

A P reset has the following effects on IRQ subsystem

- IRQ is disabled and reset the IRQ selection as negative edge-triggered and low level-triggered, hence the LED/IRQ pin function as LED output pin

Therefore also disable the LED driver output, hence the LED/IRQ pin is in high impedance state.

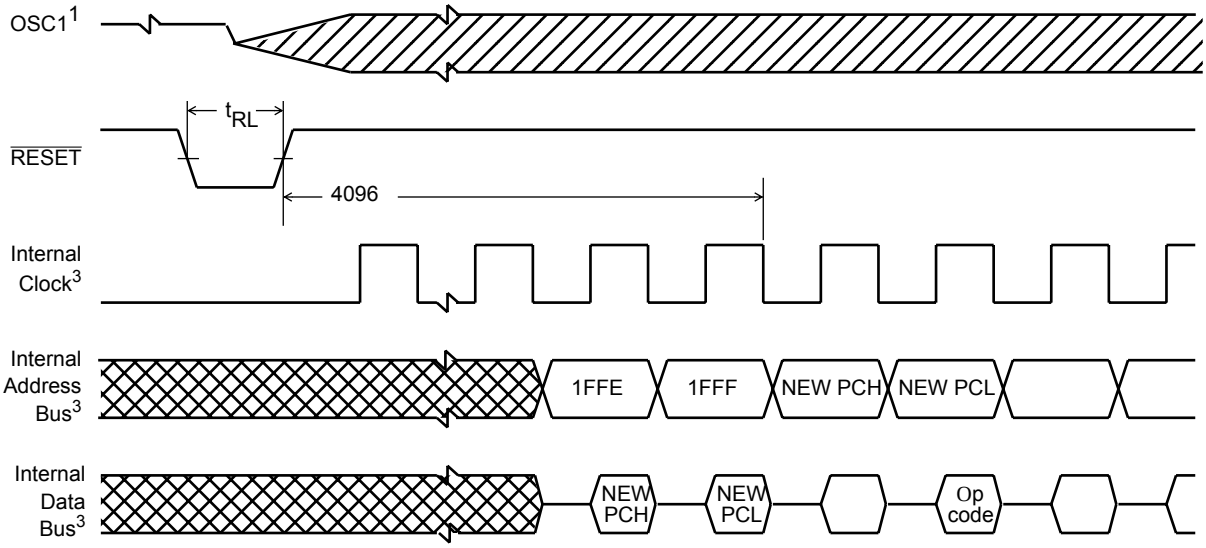
5.5 RESET CHARACTERISTICS

Table 5-1. Reset Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
POR Recovery Voltage[2]	V_{POR}	0	—	100	mV
POR V_{DD} Slew Rate[2] Rising[2] Falling[2]	 S_{VDDR} S_{VDDF}	 — —	 — —	 0.1 0.05	 V/ms V/ms
$\overline{RESET}$ Pulse Width (when bus clock active)	t_{RL}	1.5	—	—	t_{CYC}
$\overline{RESET}$ Pulldown Pulse Width (from internal reset)	t_{RPD}	3	—	4	t_{CYC}

Note:

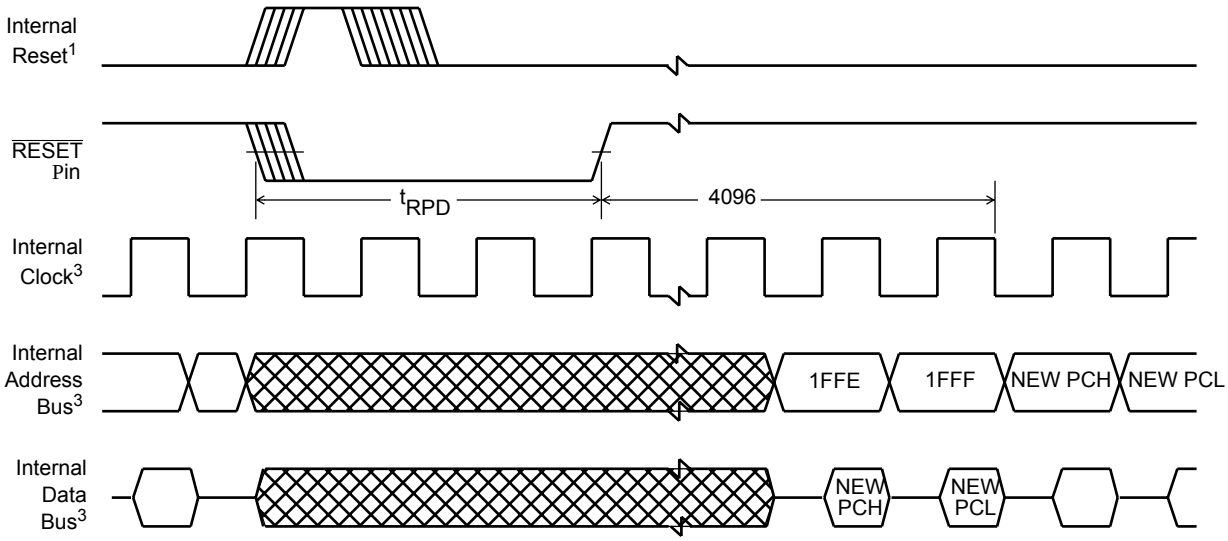
1. $+2.0 \le V_{DD} \le +4.0$ V, $V_{SS} = 0$ V, $T_L \le T_A \le T_H$, unless otherwise noted
2. By design, not tested.



NOTES:

1. Represents the internal gating of the OSC1 pin
2. Normal delay of 4064 t_{CYC}
3. Internal timing signal and data information not available externally.

Figure 5-6. Stop Recovery Timing Diagram



NOTES:

1. Represents the internal reset from low voltage reset, illegal opcode fetch or COP Watchdog timeout.
2. Normal delay of 4064 t_{CYC}
3. Internal timing signal and data information not available externally.

Figure 5-7. Internal Reset Timing Diagram

SECTION 6
OPERATING MODES

This section describes the various operating modes of the MC68HC05PL4.

6.1 OPERATING MODES

The MC68HC05PL4 has two operating modes: Single-Chip (Normal) Mode and Self-Check Mode.

At the rising edge of the $\overline{\text{RESET}}$, the device latches the states of LED/$\overline{\text{IRQ}}$ and PB0/KBI0 pins and places itself in the speci ed mode. $\overline{\text{RESET}}$ must be held low for the prede ned po wer-on reset cycles of the internal PH2 clock after POR, or for a time t_{RL} for any other reset.

The conditions required to enter each mode are shown in **Table 6-1**. The mode of operation is determined by the voltages on the LED/$\overline{\text{IRQ}}$ and PB0/KBI0 pins on the rising edge of the external $\overline{\text{RESET}}$ pin.

Table 6-1. Operation Mode Condition After Reset

$\overline{\text{RESET}}$ Pin	LED/$\overline{\text{IRQ}}$	PB0/KBI0	MODE
↑	V_{SS} to V_{DD}	V_{SS} to V_{DD}	Single-Chip (Normal)
	V_{TST}	V_{DD}	Self-Check

$V_{TST} = 2 \times V_{DD}$

6.1.1 Single-chip (Normal) Mode

The Single-Chip Mode is the normal operating mode, and it allows the device to function as a self-contained microcontroller, with maximum use of the pins for on-chip peripheral functions.

In the Single-Chip Mode all address and data activity occurs within the MCU and is not available externally. Single-Chip Mode is entered if the LED/$\overline{\text{IRQ}}$ pin is within the normal operating voltage range when the rising edge of a $\overline{\text{RESET}}$ occurs. In Single-Chip Mode, all I/O port pins are available.

6.1.2 Self-check Mode

The self-check program is mask at location $1E00 to $1FEF, and is used for checking device functionality under minimum hardware support.

6.2 LOW POWER MODES

In each of its con gur ation modes the MC68HC05PL4 is capable of running in one of two low-power operating modes. The WAIT and STOP instructions provide two modes that reduce the power required for the MCU by stopping various internal clocks and/or the oscillator. The o w of the STOP, and WAIT modes are shown in **Figure 6-1**.

6.2.1 STOP Mode

Execution of the STOP instruction places the MCU in its lowest power consumption mode.

The MCU can exit from the STOP by an $\overline{\text{IRQ}}$ or Keyboard interrupt (KBIx), or an externally generated RESET. When exiting the STOP mode the internal oscillator will resume after 4064 internal processor clock cycles oscillator stabilization delay.

6.2.2 WAIT Mode

The WAIT instruction places the MCU in a low-power mode, which consumes more power than the STOP Mode.

The WAIT mode may be exited by an external $\overline{\text{IRQ}}$, a keyboard interrupt, 16-bit timer interrupt, 8-bit timer interrupt, or by an external $\overline{\text{RESET}}$.

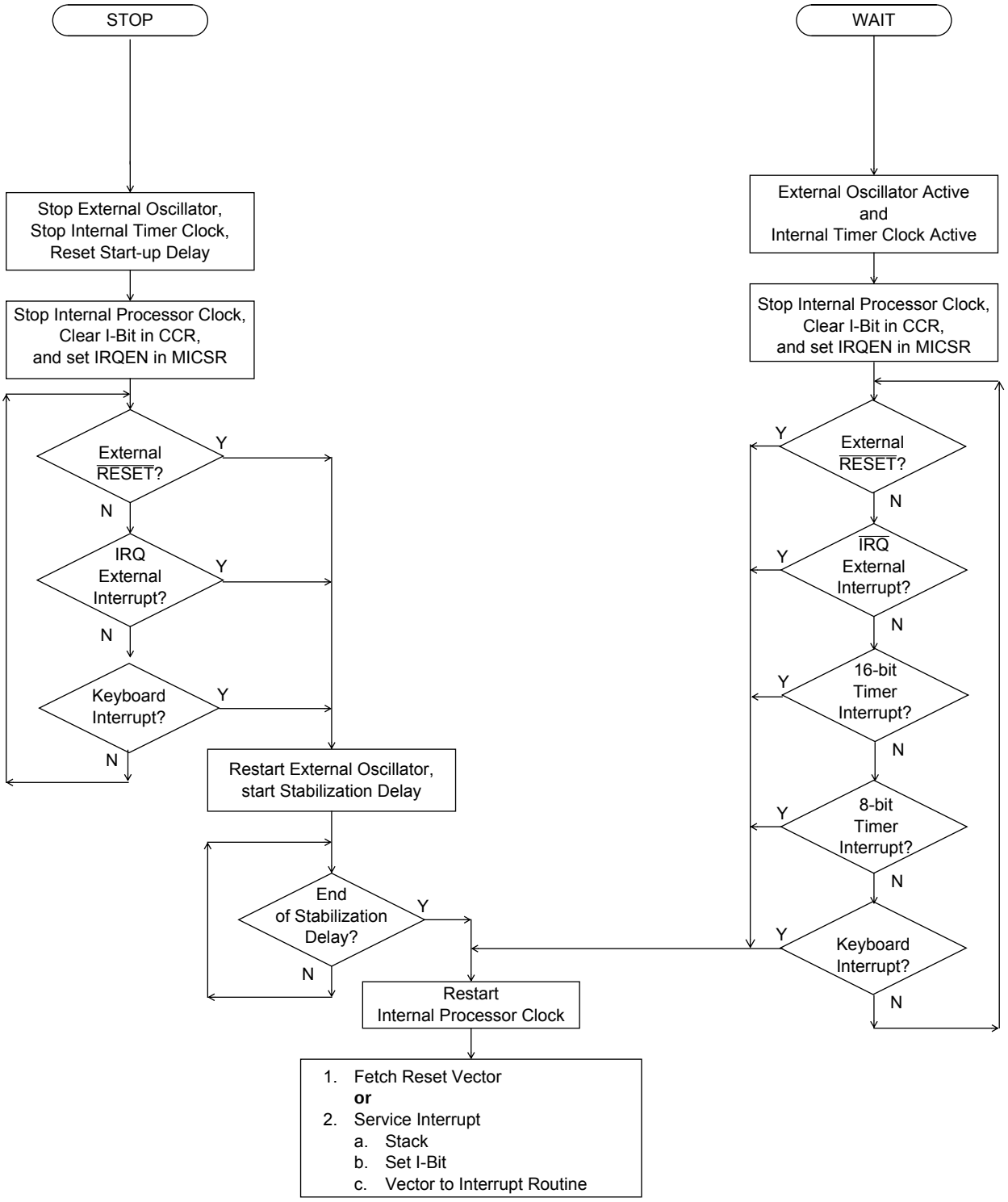


Figure 6-1. STOP/WAIT Flowchart

SECTION 7
INPUT/OUTPUT PORTS

This section describes the general purpose I/O ports on the MC68HC05PL4 and MC68HC05PL4B MCUs.

In the MC68HC05PL4, 23 bidirectional I/O lines are available, arranged as one 7-bit I/O port (Port A), one 8-bit I/O port (Port B), and one 8-bit I/O port (Port C).

In the MC68HC05PL4B, 22 bidirectional I/O lines are available, arranged as one 6-bit I/O port (Port A), one 8-bit I/O port (Port B), and one 8-bit I/O port (Port C).

NOTE

To avoid generating a glitch on an I/O port pin, data should be written to the I/O port data register before writing a “1” (for output) to the corresponding data direction register.

7.1 PARALLEL PORTS

Port A, B, and C are bidirectional ports. Each port pin is controlled by the corresponding bits in a data direction register and a data register as shown in **Figure 7-1**. The functions of the I/O pins are summarized in **Table 7-1**.

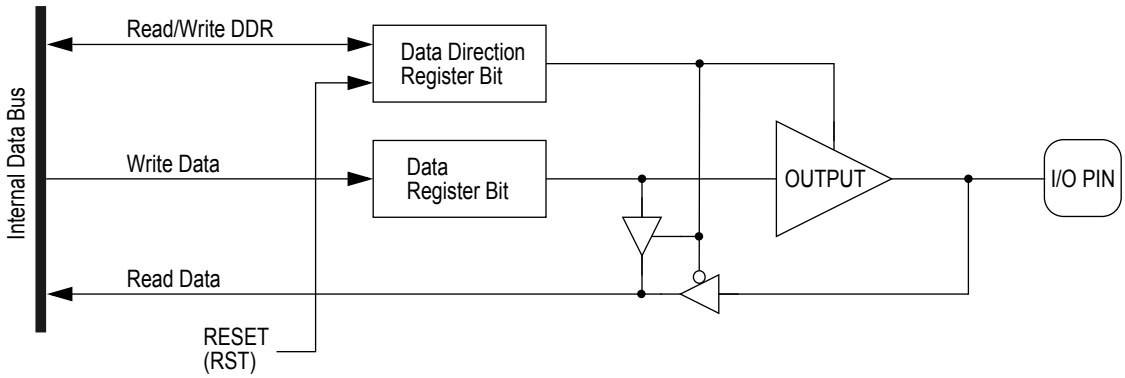


Figure 7-1. Port Input/Output Circuitry

Table 7-1. I/O Pin Functions

R/$\overline{W}$	DDR	I/O Pin Functions
0	0	The I/O pin is in input mode. Data is written into the output data latch.
0	1	Data is written into the output data latch and output to the I/O pin.
1	0	The state of the I/O pin is read.
1	1	The I/O pin is in an output mode. The output data latch is read.

7.1.1 Port Data Registers

Each port I/O pin has a corresponding bit in the Port Data Register. When a port I/O pin is programmed as an output the state of the corresponding data register bit determines the state of the output pin.

When a port pin is programmed as an input, any read of the Port Data Register will return the logic state of the corresponding I/O pin. The locations of the Data Registers for Port A, B, and C are at $0000, $0001 and $0002. The Port Data Registers are unaffected by reset.

7.1.2 Port Data Direction Registers

Each port I/O pin may be programmed as an input by clearing the corresponding bit in the DDR, or programmed as an output by setting the corresponding bit in the DDR. The DDR for Port A, B, and C are located at $0005, $0006 and $0007. The DDRs are cleared by reset.

NOTE

A “glitch” can be generated on an I/O pin when changing it from an input to an output unless the data register is rst preconditioned to the desired state before changing the corresponding DDR bit from a zero to a one.

7.2 PORT A

Port A is an 7-bit bidirectional port, with pins shared with other modules. The Port A Data Register is at address $0000 and the Data Direction Register is at address $0005. Port pins PA5 and PA6 are high current sink pins; see Electrical Speci cations section f or values.

Pin PA0 is only available on MC68HC05PL4. OSC2 replaces PA0 on MC68HC05PL4B.

Pin PA1 becomes the DTMF output from the DAC when the DACEN bit is set in the DAC Control and Data Register ($000F).

Pins PA2 and PA3 become the 16-bit timer TCAP and TCMP respectively, when TCAPEN and TCMPEN are set in the Miscellaneous Control/Status Register ($001C).

7.3 PORT B

Port B is an 8-bit bidirectional port, with pins PB0-PB3 shared with keyboard interrupt functions. The Port B Data Register is at address $0001 and the Data Direction Register is at address $0006.

Pins PB0 to PB3 keyboard interrupt functions have individual enable and flag bits in registers $000B and $000C.

7.4 PORT C

Port C is an 8-bit bidirectional port. The Port C Data Register is at address $0002 and the Data Direction Register is at address $0007. Port pins PC0 to PC3 are high current sink pins; see Electrical Specifications section for values.

7.5 SUMMARY OF PORT A AND PORT B SHARED PINS

Table 7-2 below shows a summary of port pins shared with other on-chip modules.

Table 7-2. Port A and Port B Shared Pins

Port	Port Pin	Control	Pin Name	Shared Functions
Port A	PA0	—	PA0 or OSC2	PA0 on MC68HC05PL4 OSC2 on MC68HC05PL4B
	PA1	DACEN	PA1/DTMF	DAC DTMF Output
	PA2	TCAPEN	PA2/TCAP	16-bit Timer Input Capture
	PA3	TCMPEN	PA3/TCMP	16-bit Timer Output Compare
Port B	PB3-PB0	KBIE3-KBIE0 PUL3-PUL0	PB3/KBI3-PB0/KBI0	Keyboard Interrupt

SECTION 8
SYSTEM CLOCKS

This section describes the system clock options for the MC68HC05PL4.

8.1 SYSTEM CLOCK SOURCE AND FREQUENCY OPTION

The operating bus frequency of the MCU is dependent on the clock source (OSC1 or internal RC) and the clock divider ratio. These are selected in the System Clock Control Register (SYSCR).

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SYSCR	R	SYSDIV1	SYSDIV2	CKSEL1	CKSEL2	FMODE	OSCF	RCF	CKOSC
$001D	W								
POR		0	0	1	0	1	0	1	0

Figure 8-1. System Clock Control Register (SYSCR)

SYSDIV1,SYSDIV2 — System Clock Divider Select

The SYSDIV1 and SYSDIV2 bits select the divide ratio for the clock source. After power-on-reset, the default setting is divide by 2. **Table 8-1** shows the divide ratios.

Table 8-1. System Clock Divider Select

SYSDIV1	SYSDIV2	DIV
0	0	2
0	1	4
1	0	8
1	1	16

CKSEL1,CKSEL2 — System Clock Source Select

The CKSEL1 and CKSEL2 bits select the system clock source for the MCU. After power-on-reset, the default setting is internal RC. **Table 8-2** shows the system clock source options.

Table 8-2. System Clock Source Select

CKSEL1	CKSEL2	Select Option
0	0	External from OSC1
0	1	External from OSC1
1	0	Internal RC
1	1	External from OSC1 (with RC enabled)

FMODE — Fast Mode RC select

FMODE selects the oscillating frequency of the internal RC. After power-on-reset, the default setting is 500kHz.

1 = Internal RC oscillates at 500kHz
0 = Internal RC oscillates at 20kHz

OSCF — OSC running Flag

This bit is set when the external clock (External/crystal) from OSC1 is on. See also CKOSC bit below.

RCF — RC Running Flag

This bit is set when the internal RC clock is on.

CKOSC — CheckK OSC

The CKOSC bit enables the internal logic for external clock selection. The procedure below should be followed when switching from RC to external clock.

1. Set the CKSEL1 and CKSEL2 bits for external clock source.
2. If crystal option is used — set the 8-bit timer for counting crystal stabilization delay (typically 4064 clock cycles).
3. Write a “1” to the CKOSC bit and check for OSCF bit set.
4. If the OSCF bit is not set, no external clock is available, the internal RC clock will be used as the system clock, irrespective of the setting for CKSEL1 and CKSEL2.

SECTION 9
16-BIT PROGRAMMABLE TIMER

The MC68HC05PL4 MCU contains a 16-bit programmable Timer with an Input Capture function and an Output Compare function as shown by the block diagram in **Figure 9-1**.

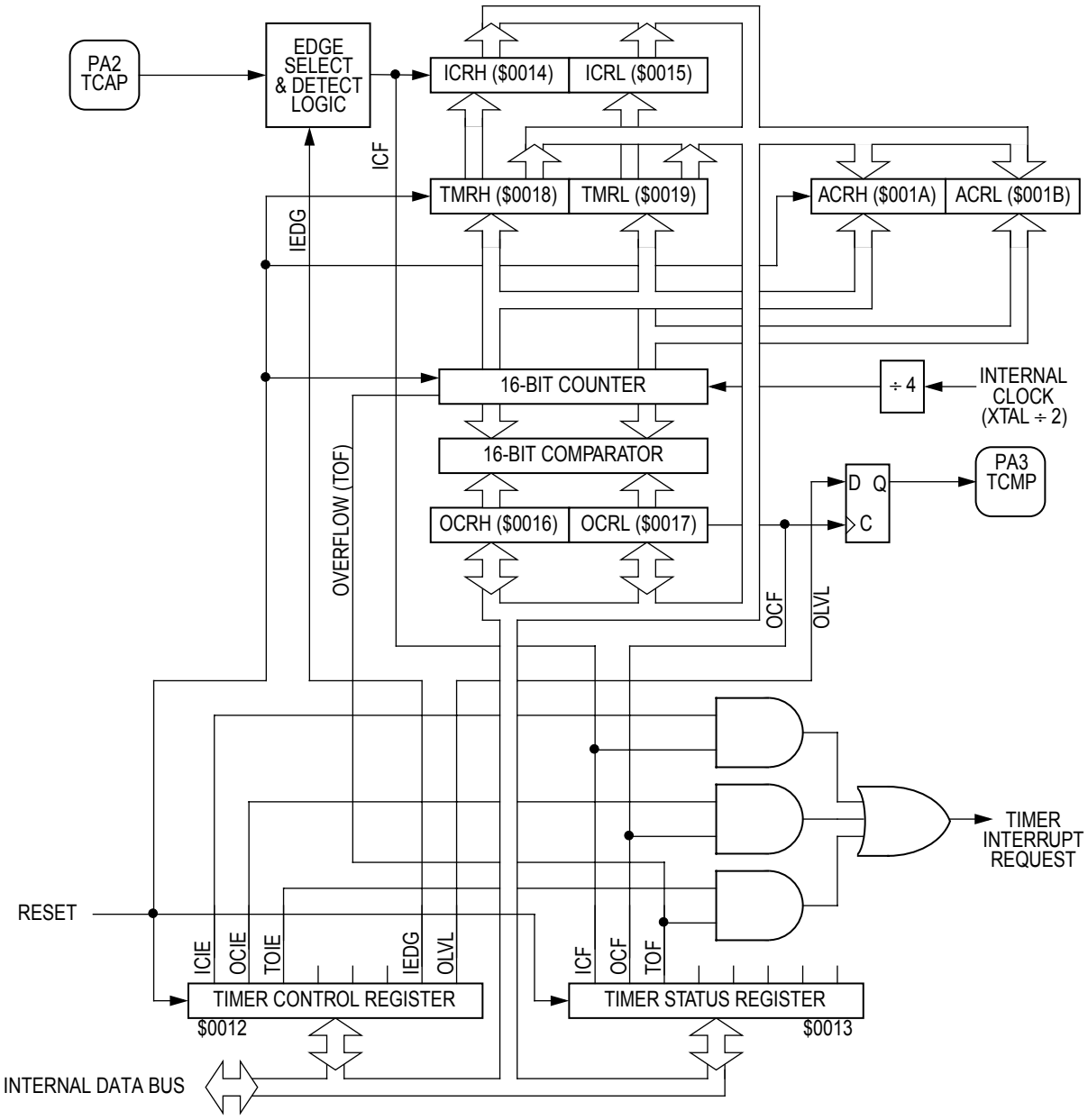


Figure 9-1. Programmable Timer Block Diagram

The basis of the capture/compare Timer is a 16-bit free-running counter which increases in count with each internal bus clock cycle. The counter is the timing reference for the input capture and output compare functions. The input capture and output compare functions provide a means to latch the times at which external events occur, to measure input waveforms, and to generate output waveforms and timing delays. Software can read the value in the 16-bit free-running counter at any time without affect the counter sequence.

Because of the 16-bit timer architecture, the I/O registers for the input capture and output compare functions are pairs of 8-bit registers. Each register pair contains the high and low byte of that function. Generally, accessing the low byte of a specific timer function allows full control of that function; however, an access of the high byte inhibits that specific timer function until the low byte is also accessed.

Because the counter is 16 bits long and preceded by a fixed divide-by-four prescaler, the counter rolls over every 262,144 internal clock cycles. Timer resolution with a 4 MHz crystal oscillator is 2 microsecond/count.

The interrupt capability, the input capture edge, and the output compare state are controlled by the timer control register (TCR) located at $0012 and the status of the interrupt flags can be read from the timer status register (TSR) located at $0013.

9.1 TIMER REGISTERS (TMRH, TMRL)

The functional block diagram of the 16-bit free-running timer counter and timer registers is shown in **Figure 9-2**. The timer registers include a transparent buffer latch on the LSB of the 16-bit timer counter.

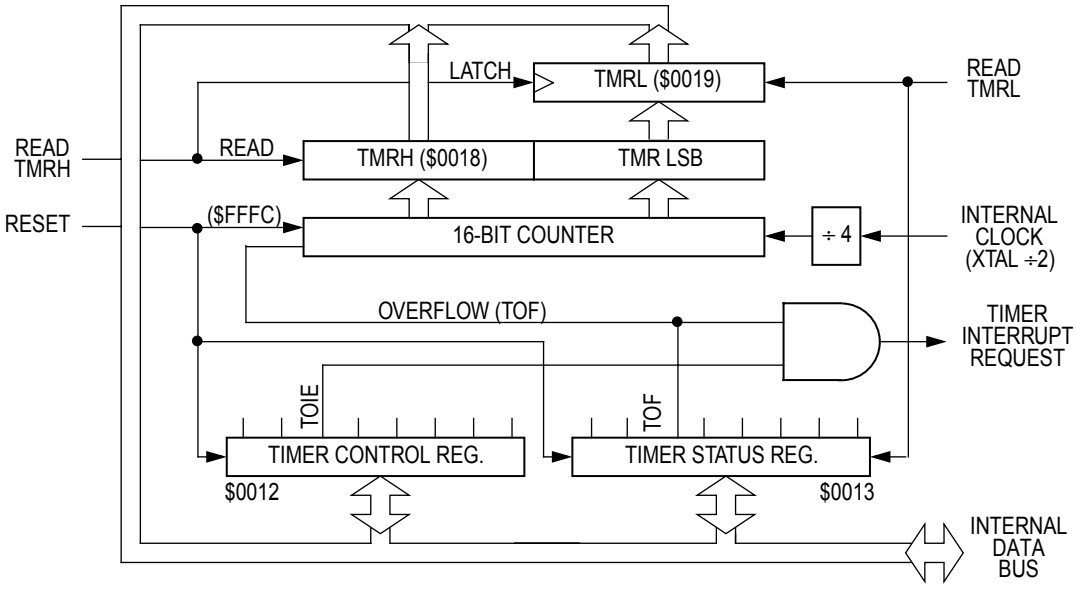


Figure 9-2. Timer Counter and Register Block Diagram

The timer registers (TMRH, TMRL) shown in **Figure 9-3** are read-only locations which contain the current high and low bytes of the 16-bit free-running counter. Writing to the timer registers has no effect. Reset of the device presets the timer counter to $FFFC.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TMRH	R	TMRH7	TMRH6	TMRH5	TMRH4	TMRH3	TMRH2	TMRH1	TMRH0
$0018	W								
reset:		1	1	1	1	1	1	1	1
TMRL	R	TMRL7	TMRL6	TMRL5	TMRL4	TMRL3	TMRL2	TMRL1	TMRL0
$0019	W								
reset:		1	1	1	1	1	1	0	0

Figure 9-3. Programmable Timer Registers (TMRH, TMRL)

The TMRL latch is a transparent read of the LSB until the a read of the TMRH takes place. A read of the TMRH latches the LSB into the TMRL location until the TMRL is again read. The latched value remains fixed even if multiple reads of the TMRH take place before the next read of the TMRL. Therefore, when reading the MSB of the timer at TMRH the LSB of the timer at TMRL must also be read to complete the read sequence.

During power-on-reset (POR), the counter is initialized to $FFFC and begins counting after the oscillator start-up delay. Because the counter is sixteen bits and preceded by a fixed divide-by-four prescaler, the value in the counter repeats every 262, 144 internal bus clock cycles (524, 288 oscillator cycles).

When the free-running counter rolls over from $FFFF to $0000, the timer overflow flag bit (TOF) is set in the TSR. When the TOF is set, it can generate an interrupt if the timer overflow interrupt enable bit (TOIE) is also set in the TCR. The TOF flag bit can only be reset by reading the TMRL after reading the TSR.

Other than clearing any possible TOF flags, reading the TMRH and TMRL in any order or any number of times does not have any effect on the 16-bit free-running counter.

NOTE

To prevent interrupts from occurring between readings of the TMRH and TMRL, set the I bit in the condition code register (CCR) before reading TMRH and clear the I bit after reading TMRL.

9.2 ALTERNATE COUNTER REGISTERS (ACRH, ACRL)

The functional block diagram of the 16-bit free-running timer counter and alternate counter registers is shown in **Figure 9-4**. The alternate counter registers behave the same as the timer registers, except that any reads of the alternate counter will not have any effect on the TOF ag bit and Timer interrupts. The alternate counter registers include a transparent buffer latch on the LSB of the 16-bit timer counter.

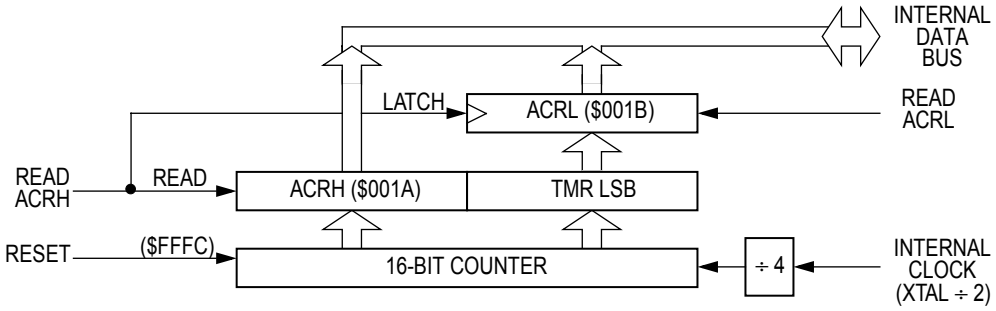


Figure 9-4. Alternate Counter Block Diagram

The alternate counter registers (ACRH, ACRL) shown in **Figure 9-5** are read-only locations which contain the current high and low bytes of the 16-bit free-running counter. Writing to the alternate counter registers has no effect. Reset of the device presets the timer counter to $FFFC.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ACRH	R	ACRH7	ACRH6	ACRH5	ACRH4	ACRH3	ACRH2	ACRH1	ACRH0
$001A	W								
reset:		1	1	1	1	1	1	1	1

ACRL	R	ACRL7	ACRL6	ACRL5	ACRL4	ACRL3	ACRL2	ACRL1	ACRL0
$001B	W								
reset:		1	1	1	1	1	1	0	0

Figure 9-5. Alternate Counter Registers (ACRH, ACRL)

The ACRL latch is a transparent read of the LSB until the a read of the ACRH takes place. A read of the ACRH latches the LSB into the ACRL location until the ACRL is again read. The latched value remains x ed even if multiple reads of the ACRH take place before the next read of the ACRL. Therefore, when reading the MSB of the timer at ACRH the LSB of the timer at ACRL must also be read to complete the read sequence.

During power-on-reset (POR), the counter is initialized to $FFFC and begins counting after the oscillator start-up delay. Because the counter is sixteen bits and preceded by a xed divide-by-four prescaler, the value in the counter repeats every 262,144 internal bus clock cycles (524,288 oscillator cycles).

Reading the ACRH and ACRL in any order or any number of times does not have any effect on the 16-bit free-running counter or the TOF ag bit.

NOTE

To prevent interrupts from occurring between readings of the ACRH and ACRL, set the I bit in the condition code register (CCR) before reading ACRH and clear the I bit after reading ACRL.

9.3 INPUT CAPTURE REGISTERS

The input capture function is a technique whereby an external signal (connected to PA2/TCAP pin) is used to trigger the 16-bit timer counter. In this way it is possible to relate the timing of an external signal to the internal counter value, and hence to elapsed time.

When the input capture circuitry detects an active edge on the selected source, it latches the contents of the free-running timer counter registers into the input capture registers as shown in **Figure 9-6**.

Latching values into the input capture registers at successive edges of the same polarity measures the period of the selected input signal. Latching the counter values at successive edges of opposite polarity measures the pulse width of the signal.

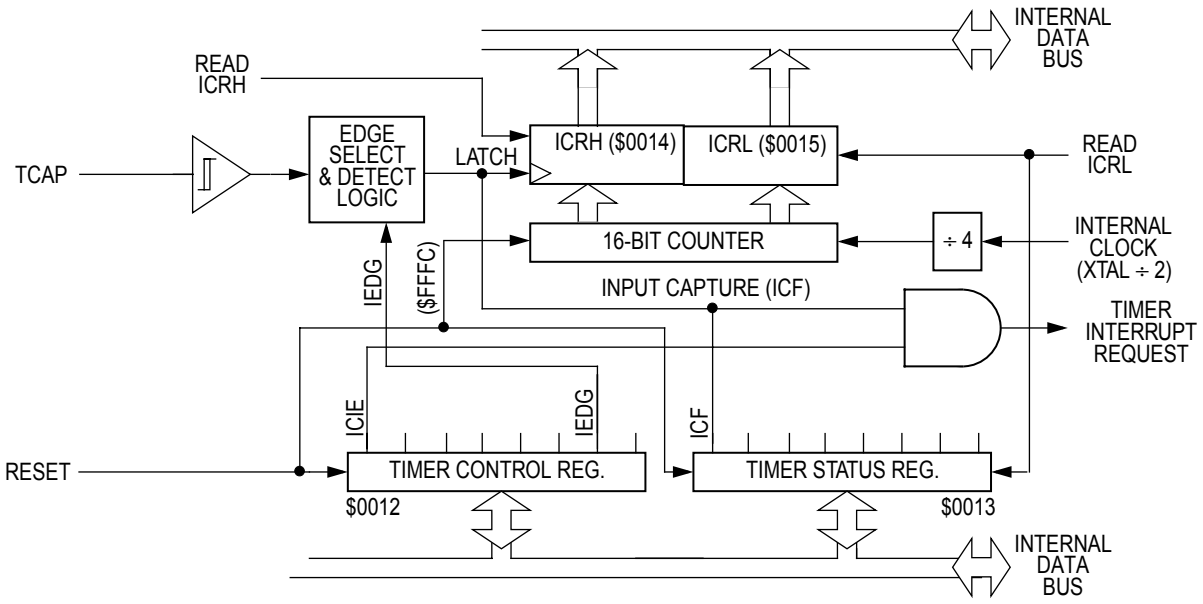


Figure 9-6. Timer Input Capture Block Diagram

The input capture registers are made up of two 8-bit read-only registers (ICRH, ICRL) as shown in **Figure 9-7**. The input capture edge detector contains a Schmitt trigger to improve noise immunity. The edge that triggers the counter transfer is de ned by the input edge bit (IEDG) in the TCR. Reset does not affect the contents of the input capture registers.

The result obtained by an input capture will be one count higher than the value of the free-running timer counter preceding the external transition. This delay is required for internal synchronization. Resolution is affected by the prescaler, allowing the free-running timer counter to increment once every four internal clock cycles (eight oscillator clock cycles).

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ICRH	R	ICRH7	ICRH6	ICRH5	ICRH4	ICRH3	ICRH2	ICRH1	ICRH0
$0014	W								
reset:		U	U	U	U	U	U	U	U

ICRL	R	ICRL7	ICRL6	ICRL5	ICRL4	ICRL3	ICRL2	ICRL1	ICRL0
$0015	W								
reset:		U	U	U	U	U	U	U	U

U = UNAFFECTED BY RESET

Figure 9-7. Input Capture Registers (ICRH, ICRL)

Reading the ICRH inhibits further captures until the ICRL is also read. Reading the ICRL after reading the timer status register (TSR) clears the ICF ag bit. does not inhibit transfer of the free-running counter. There is no con ict betw een reading the ICRL and transfers from the free-running timer counters. The input capture registers always contain the free-running timer counter value which corresponds to the most recent input capture.

NOTE

To prevent interrupts from occurring between readings of the ICRH and ICRL, set the I bit in the condition code register (CCR) before reading ICRH and clear the I bit after reading ICRL.

9.4 OUTPUT COMPARE REGISTERS

The Output Compare function is a means of generating an output signal when the 16-bit timer counter reaches a selected value as shown in **Figure 9-8**. Software writes the selected value into the output compare registers. On every fourth internal clock cycle (every eight oscillator clock cycle) the output compare circuitry compares the value of the free-running timer counter to the value written in the output compare registers. When a match occurs, the timer transfers the output level (OLVL) from the timer control register (TCR) to the TCMP.

Software can use the output compare register to measure time periods, to generate timing delays, or to generate a pulse of speci c dur ation or a pulse train of speci c frequency and duty cycle on the TCMP.

The planned action on the TCMP depends on the value stored in the OLVL bit in the TCR, and it occurs when the value of the 16-bit free-running timer counter matches the value in the output compare registers shown in **Figure 9-3**. These registers are read/write bits and are unaffected by reset.

Writing to the OCRH before writing to the OCRL inhibits timer compares until the OCRL is written. Reading or writing to the OCRL after reading the TCR will clear the output compare ag bit (OCF). The output compare OLVL state will be clocked to its output latch regardless of the state of the OCF.

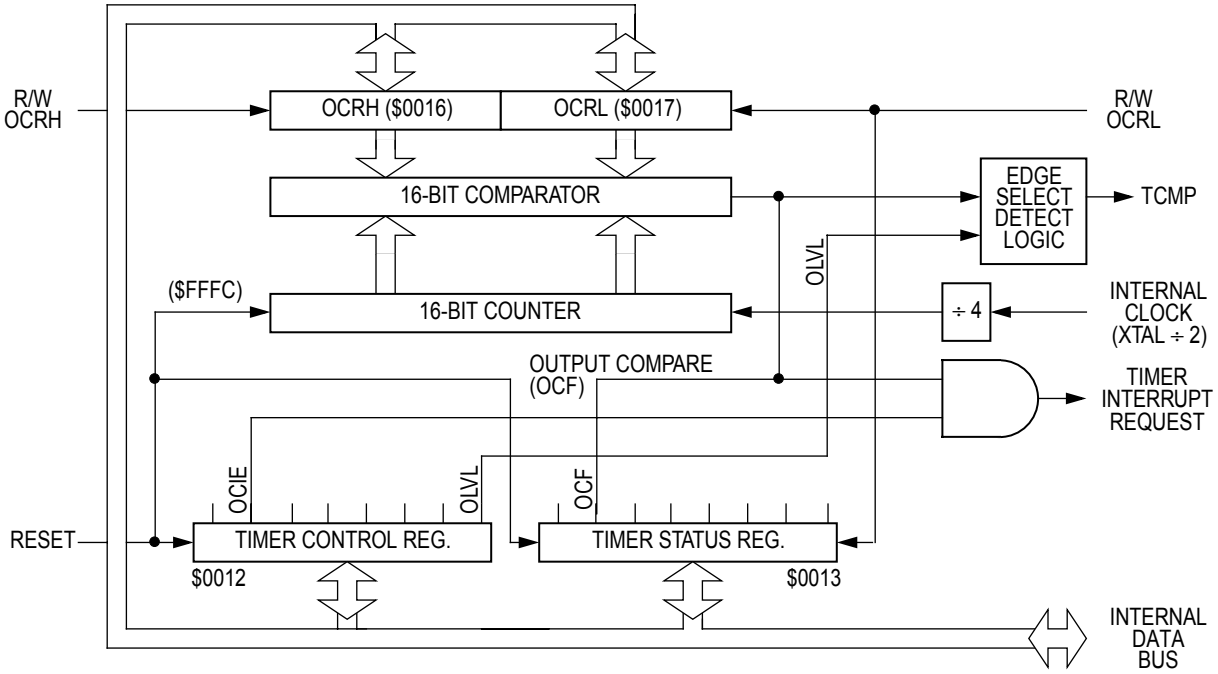


Figure 9-8. Timer Output Compare Block Diagram

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OCRH	R	OCRH7	OCRH6	OCRH5	OCRH4	OCRH3	OCRH2	OCRH1	OCRH0
$0016	W								
reset:		U	U	U	U	U	U	U	U

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OCRL	R	OCRL7	OCRL6	OCRL5	OCRL4	OCRL3	OCRL2	OCRL1	OCRL0
$0017	W								
reset:		U	U	U	U	U	U	U	U

U = UNAFFECTED BY RESET

Figure 9-9. Output Compare Registers (OCRH, OCRL)

To prevent OCF from being set between the time it is read and the time the output compare registers are updated, use the following procedure:

1. Disable interrupts by setting the I bit in the condition code register.
2. Write to the OCRH. Compares are now inhibited until OCRL is written.

3. Read the TSR to arm the OCF for clearing.
4. Enable the output compare registers by writing to the OCRL. This also clears the OCF ag bit in the TSR.
5. Enable interrupts by clearing the I bit in the condition code register.

A software example of this procedure is shown in **Table 9-1**.

Table 9-1. Output Compare Initialization Example

```
9B              SEI             DISABLE INTERRUPTS
...             ...             .....
...             ...             .....
B7      16      STA     OCRH    INHIBIT OUTPUT COMPARE
B6      13      LDA     TSR     ARM OCF FLAG FOR CLEARING
BF      17      STX     OCRL    READY FOR NEXT COMPARE, OCF CLEARED
...             ...             .....
...             ...             .....
9A              CLI             ENABLE INTERRUPTS
```

9.5 TIMER CONTROL REGISTER (TCR)

The timer control register shown in **Figure 9-10** performs the following functions:

- Enables input capture interrupts.
- Enables output compare interrupts.
- Enables timer over o w interrupts.
- Con gure the I/O P ort Pin PA2 as input pin for TCAP signal
- Con gure the I/O P ort Pin PA3 as output pin for TCMP signal
- Control the active edge polarity of the TCAP signal.
- Controls the active level of the TCMP output.

Reset clears all the bits in the TCR with the exception of the IEDG bit which is unaffected.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TCR	R	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
$0012	W								
reset:		0	0	0	0	0	0	U	0

Figure 9-10. Timer Control Register (TCR)

ICIE - INPUT CAPTURE INTERRUPT ENABLE

This read/write bit enables interrupts caused by an active signal on the PB1/ TCAP pin or from CPF2 ag bit of the analog subsystem v oltage comparator 2. Reset clears the ICIE bit.

1 = Input capture interrupts enabled.
0 = Input capture interrupts disabled.

OCIE - OUTPUT COMPARE INTERRUPT ENABLE

This read/write bit enables interrupts caused by an active signal on the TCMP pin. Reset clears the OCIE bit.

1 = Output compare interrupts enabled.
0 = Output compare interrupts disabled.

TOIE - TIMER OVERFLOW INTERRUPT ENABLE

This read/write bit enables interrupts caused by a timer over o w. Reset clears the TOIE bit.

1 = Timer over o w interrupts enabled.
0 = Timer over o w interrupts disabled.

IEDG - INPUT CAPTURE EDGE SELECT

The state of this read/write bit determines whether a positive or negative transition on the TCAP pin triggers a transfer of the contents of the timer register to the input capture register. Resets have no effect on the IEDG bit.

1 = Positive edge (low to high transition) triggers input capture.
0 = Negative edge (high to low transition) triggers input capture.

OLVL - OUTPUT COMPARE OUTPUT LEVEL SELECT

The state of this read/write bit determines whether a logic one or a logic zero appears on the TCMP when a successful output compare occurs. Resets clear the OLVL bit.

1 = TCMP goes high on output compare.
0 = TCMP goes low on output compare.

9.5.1 Miscellaneous Control and Status Register for Timer16

The Miscellaneous Control and Status Register shown in **Figure 9-11** performs the following functions:

- Con gure the I/O por t pin PA2 as input pin for TCAP signal
- Con gure the I/O por t pin PA3 as output pin for TCMP signal

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
MICSR	R	IRQEN	IRQS	TCMPEN	TCAPEN	0	LED	COPON	POR
$001C	W								
reset:		0	0	0	0	0	0	0	1

Figure 9-11. Miscellaneous Control and Status Register (MISCR)

TCAPEN - TIMER INPUT CAPTURE ENABLE

The bit con gures por t pin PA2 for Timer16 input capture function (TCAP). At power-on-reset, this bit is cleared, PA2 is a standard I/O port pin, TCAP to the Timer16 is pulled high.

1 = PA2 pin con gured as TCAP for timer input capture
0 = PA2 pin as standard I/O port pin

TCMPEN - TIMER OUTPUT COMPARE ENABLE

The bit con gures por t pin PA3 for Timer16 output compare function (TCMP). At power-on-reset, this bit is cleared, PA3 is a standard I/O port pin, TCMP signal to PA3 is disabled from Timer16.

1 = PA3 pin con gured as TCMP for timer output compare
0 = PA3 pin as standard I/O port pin

9.6 TIMER STATUS REGISTER (TSR)

The timer status register (TSR) shown in **Figure 9-12** contains ags f or the following events:

- An active signal on the PA2/TCAP pin transferring the contents of the timer registers to the input capture registers.
- A match between the 16-bit counter and the output compare registers, transferring the OLVL bit to the TCMP.
- An over o w of the timer registers from $FFFF to $0000.

Writing to any of the bits in the TSR has no effect. Reset does not change the state of any of the ag bits in the TSR.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
TSR	R	ICF	OCF	TOF	0	0	0	0	0
$0013	W								
reset:		U	U	U	0	0	0	0	0

U = UNAFFECTED BY RESET

Figure 9-12. Timer Status Registers (TSR)

ICF - INPUT CAPTURE FLAG

The ICF bit is automatically set when an edge of the selected polarity occurs on the PA2/TCAP pin. Clear the ICF bit by reading the timer status register with the ICF set, and then reading the low byte (ICRL, $0015) of the input capture registers. Resets have no effect on ICF.

OCF - OUTPUT COMPARE FLAG

The OCF bit is automatically set when the value of the timer registers matches the contents of the output compare registers. Clear the OCF bit by reading the timer status register with the OCF set, and then accessing the low byte (OCRL, $0017) of the output compare registers. Resets have no effect on OCF.

TOF - TIMER OVERFLOW FLAG

The TOF bit is automatically set when the 16-bit timer counter rolls over from $FFFF to $0000. Clear the TOF bit by reading the timer status register with the TOF set, and then accessing the low byte (TMRL, $0019) of the timer registers. Resets have no effect on TOF.

9.7 16-BIT TIMER OPERATION DURING WAIT MODE

During WAIT mode the 16-bit timer continues to operate normally and may generate an interrupt to trigger the MCU out of the WAIT mode.

9.8 16-BIT TIMER OPERATION DURING STOP MODE

When the MCU enters the STOP mode the free-running counter stops counting (the internal processor clock is stopped). It remains at that particular count value until the STOP mode is exited by applying a low signal to the $\overline{\text{IRQ}}$ pin, at which time the counter resumes from its stopped value as if nothing had happened. If STOP mode is exited via an external reset (logic low applied to the $\overline{\text{RESET}}$ pin) the counter is forced to timer interrupt vector.

If a valid input capture edge occurs at the PA2/TCAP pin during the STOP mode the input capture detect circuitry will be armed. This action does not set any ags or “wake up” the MCU, but when the MCU does “wake up” there will be an active input capture ag (and data) from the rst v alid edge. If the STOP mode is exited by an external reset, no input capture ag or data will be present even if a valid input capture edge was detected during the STOP mode.

SECTION 10
8-BIT TIMER

This section describes the 8-bit count down timer module.

10.1 OVERVIEW

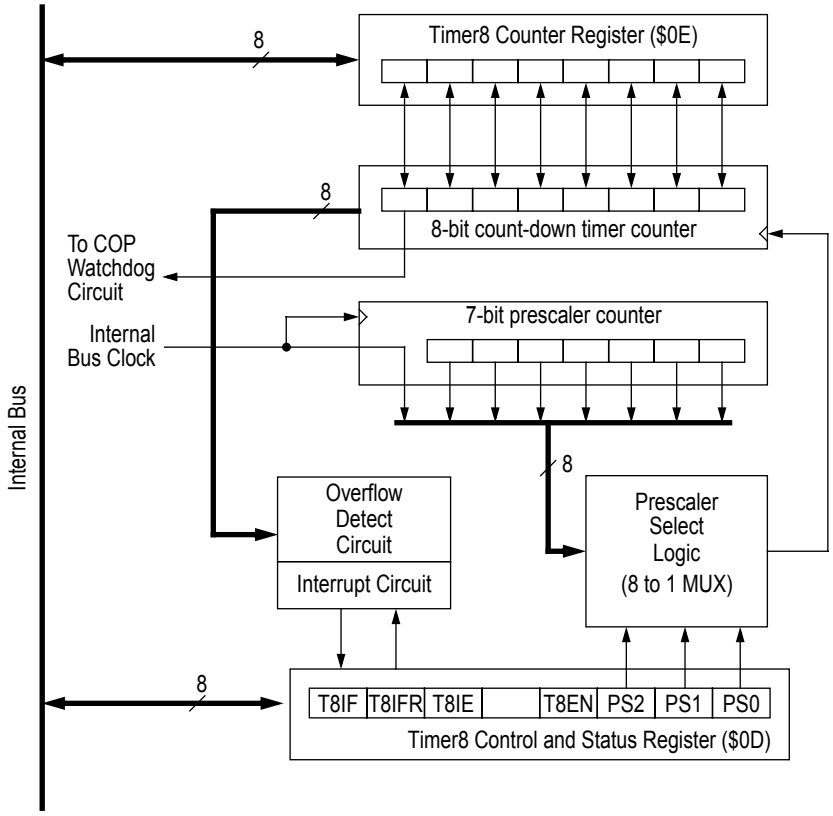


Figure 10-1. Timer8 Block Diagram

As shown in **Figure 10-1** this timer contains a single 8-bit software programmable countdown timer counter with a 3-bit software control prescaler. The counter's value may be preset under software control and counts down to zero. When the counter decrements to zero, the timer8 interrupt request bit (T8IF in T8CR) is set. Then if the timer interrupt is enabled (T8IE in T8CR is set) and the I-bit of the condition code register are is cleared, the processor receives an interrupt. After completion of the current instruction, the processor proceeds to store the appropriate registers on the stack and then fetches the timer8 interrupt vector in order to begin serving the interrupt.

The counter continues to count after it reaches zero, allowing the software to determine the number of internal or external clocks since the timer interrupt request bit (T8IF) was set. The counter may be read at any time by the processor without disturbing the count. The contents of the counter become stable prior to the read portion of a cycle and do not change during the read. The timer interrupt request bit (T8IF) remains set until cleared by writing a “1” to the T8IFR bit in the T8CR. If writing to the timer 8 counter register (T8CNTR) occurs before the timer interrupt is served, the interrupt is lost. The T8IF bit may also be used as a scanned status bit in a non-interrupt mode of operation.

The 3-bit control prescaler is a 7-bit divider which is used to extend the maximum length of the timer. Bit 0, bit 1 and bit 2 (PS0, PS1 and PS2) of T8CR are programmed to choose the appropriate prescaler output which is used as the counter input.

10.2 TIMER8 CONTROL AND STATUS REGISTER (T8CSR)

The T8CSR at address $000D enables the software to control the operation of the 8-bit timer.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
T8CSR	R	T8IF	0	T8IE	0	T8EN	PS2	PS1	PS0
$000D	W		T8IFR						
reset:		0	0	0	0	0	1	0	0

Figure 10-2. Timer8 Control and Status Register

T8IF - Timer8 Interrupt Flag

T8IF is set when Timer8 Counter Register counts down to zero. A CPU interrupt request will be generated if T8IE is set. Writing a "1" to the T8IFR bit clears the T8IF bit. Writing a "0" to this bit has no effect. Reset clears T8IF.

1 = Timer8 has count down to zero
0 = Timer8 has not count down to zero

T8IFR - Timer8 Interrupt Flag Reset

The T8IFR bit is a write-only bit, which clears the T8IF ag b y writing “1” to this bit when the T8IF bit is set. Writing a "0" has no effect. Reset does not affect this bit

1 = Clear T8IF ag bit
0 = No effect on T8IF ag bit

T8IE - Interval Timer Interrupt Enable

When this bit is set, a CPU interrupt request is generated when the T8IF bit is set. Reset clears this bit.

1 = 8-Bit Timer Interrupt enabled
0 = 8-Bit Timer Interrupt disabled

T8EN - Timer8 Enable

This read/write bit enables the Timer8. Reset clears this bit.

1 = Timer8 enabled
0 = Timer8 disabled

PS2-PS0 - Prescaler select

These read/write bits is used to select the clock frequency to drive the 8-bit timer counter. The counter will be driven by a internal bus clock (E-clock) through this prescaler ratio. Upon reset and power on reset, the value of prescaler is set to a default value of divided by 16.

PS2	PS1	PS0	DIVIDE RATIO
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16 (default after reset)
1	0	1	32
1	1	0	64
1	1	1	128

10.3 TIMER8 COUNTER REGISTER (T8CNTR)

The T8CNTR is a read/write register which contains the current value of the 8-bit timer counter. Reading this register enables the software to calculate the number of internal and external clocks since the timer interrupt request ag (T8IF) was set. Reading this address does not disturb the counter operation.

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
T8CNTR	R	Timer 8 Counter Register							
$000E	W								
reset:		U	U	U	U	U	U	U	U

Figure 10-3. Timer8 Counter Register

NOTE

This timer is used during the power-on sequence to time out the POR signal. The timer is con gured at po wer-on, with a prescaler division ratio of 16 and set to $FF in Timer counter register. Also the clock source for the COP watchdog system is derived from the output of this timer, hence a reset or preset of the prescaler and timer counter register may affect the frequency of the watchdog timeout.

10.4 COMPUTER OPERATING PROPERLY (COP) WATCHDOG

Please refer to section on RESETS for details.

10.5 8-BIT TIMER OPERATION DURING WAIT MODE

The CPU clock halts during the WAIT mode, but the timer remains active. If the interrupts are enabled, the timer interrupt will cause the processor to exit the WAIT mode.

10.6 8-BIT TIMER OPERATION DURING STOP MODE

The timer ceases counting in STOP mode. When STOP is exited by an external interrupt or an external reset, the internal oscillator will resume its operation, followed by internal processor stabilization delay. The timer is then cleared to zero and resumes its operation.

NOTE

The T8IF bit in T8CSR will be set after MCU exit from STOP mode. To avoid generation of the timer 8 interrupt when exiting STOP mode, it is recommended to clear T8IE bit prior entering STOP mode. After exiting STOP mode T8IF bit must be cleared before setting T8IE bit.

SECTION 11
DIGITAL TO ANALOG CONVERTER

This section describes Digital-to-Analog module used for DTMF generation.

11.1 DAC CONTROL AND DATA REGISTER

		BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
DACDR	R	DACEN	0	DA5	DA4	DA3	DA2	DA1	DA0
$000F	W								
reset:		0	0	0	0	0	0	0	0

Figure 11-1. DAC Control and Data Register

DACEN - DAC Channel Enable

Ths read/write bit enables/disables the DAC module for DTMF output.

1 = Enable DAC module and con gure PA1/DTMF as DTMF output pin.
0 = Disable DAC module and con gure PA1/DTMF as general purpose PA1 pin.

DA5-DA0

These bits determine the output voltage of the DAC channel. The output voltage value is determined by:

$$V_{OUT} = (V_{DD} \times DA[0:5]) \times 2^6$$

There are 64 evenly spaced voltage levels available between V_{DD} and V_{SS}. The lowest voltage is V_{SS} and the highest voltage is 63/64 V_{DD}.

11.2 DAC OPERATION DURING WAIT MODE

In WAIT mode, the DAC continues to output a xed voltage level which is set by the DA5-DA0 bits. The DAC should be disabled by clearing the DACEN bit if further power saving is required in WAIT mode.

11.3 DAC OPERATION DURING STOP MODE

In STOP mode, the DAC continues to output a xed voltage level which is set by the DA5-DA0 bits. The DAC should be disabled by clearing the DACEN bit if further power saving is required in STOP mode.

11.4 DAC CHARACTERISTICS

(V_{DD} = 4.0V ±10%, V_{SS} = 0 V_{dc}, T_A = T_L °C to T_H°C, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Resolution	—	6	6	Bits
Absolute Accuracy 4.0V 2.0V	 V_{out} V_{out}	0	±V_{DD}/64	V
DAC Output Resistance	R_{dac}	7600	15600	Ω

SECTION 12
INSTRUCTION SET

This section describes the addressing modes and instruction types.

12.1 ADDRESSING MODES

The CPU uses eight addressing modes for exibility in accessing data. The addressing modes de ne the manner in which the CPU nds the data required to execute an instruction. The eight addressing modes are the following:

- Inherent
- Immediate
- Direct
- Extended
- Indexed, No Offset
- Indexed, 8-Bit Offset
- Indexed, 16-Bit Offset
- Relative

12.1.1 Inherent

Inherent instructions are those that have no operand, such as return from interrupt (RTI) and stop (STOP). Some of the inherent instructions act on data in the CPU registers, such as set carry ag (SEC) and increment accumulator (INCA). Inherent instructions require no memory address and are one byte long.

12.1.2 Immediate

Immediate instructions are those that contain a value to be used in an operation with the value in the accumulator or index register. Immediate instructions require no memory address and are two bytes long. The opcode is the rst byte, and the immediate data value is the second byte.

12.1.3 Direct

Direct instructions can access any of the first 256 memory addresses with two bytes. The first byte is the opcode, and the second is the low byte of the operand address. In direct addressing, the CPU automatically uses $00 as the high byte of the operand address. BRSET and BRCLR are three-byte instructions that use direct addressing to access the operand and relative addressing to specify a branch destination.

12.1.4 Extended

Extended instructions use only three bytes to access any address in memory. The first byte is the opcode; the second and third bytes are the high and low bytes of the operand address.

When using the Freescale assembler, the programmer does not need to specify whether an instruction is direct or extended. The assembler automatically selects the shortest form of the instruction.

12.1.5 Indexed, No Offset

Indexed instructions with no offset are one-byte instructions that can access data with variable addresses within the first 256 memory locations. The index register contains the low byte of the conditional address of the operand. The CPU automatically uses $00 as the high byte, so these instructions can address locations $0000–$00FF.

Indexed, no offset instructions are often used to move a pointer through a table or to hold the address of a frequently used RAM or I/O location.

12.1.6 Indexed, 8-Bit Offset

Indexed, 8-bit offset instructions are two-byte instructions that can access data with variable addresses within the first 511 memory locations. The CPU adds the unsigned byte in the index register to the unsigned byte following the opcode. The sum is the conditional address of the operand. These instructions can access locations $0000–$01FE.

Indexed 8-bit offset instructions are useful for selecting the kth element in an n-element table. The table can begin anywhere within the first 256 memory locations and could extend as far as location 510 ($01FE). The k value is typically in the index register, and the address of the beginning of the table is in the byte following the opcode.

12.1.7 Indexed, 16-Bit Offset

Indexed, 16-bit offset instructions are three-byte instructions that can access data with variable addresses at any location in memory. The CPU adds the unsigned byte in the index register to the two unsigned bytes following the opcode. The sum is the conditional address of the operand. The rst byte after the opcode is the high byte of the 16-bit offset; the second byte is the low byte of the offset. These instructions can address any location in memory.

Indexed, 16-bit offset instructions are useful for selecting the kth element in an n-element table anywhere in memory.

As with direct and extended addressing, the Freescale assembler determines the shortest form of indexed addressing.

12.1.8 Relative

Relative addressing is only for branch instructions. If the branch condition is true, the CPU nds the conditional branch destination by adding the signed byte following the opcode to the contents of the program counter. If the branch condition is not true, the CPU goes to the next instruction. The offset is a signed, two's complement byte that gives a branching range of –128 to +127 bytes from the address of the next location after the branch instruction.

When using the Freescale assembler, the programmer does not need to calculate the offset, because the assembler determines the proper offset and veri es that it is within the span of the branch.

12.1.9 Instruction Types

The MCU instructions fall into the following v e categories:

- Register/Memory Instructions
- Read-Modify-Write Instructions
- Jump/Branch Instructions
- Bit Manipulation Instructions
- Control Instructions

12.1.10 Register/Memory Instructions

Most of these instructions use two operands. One operand is in either the accumulator or the index register. The CPU finds the other operand in memory. **Table 12-1** lists the register/memory instructions.

Table 12-1. Register/Memory Instructions

Instruction	Mnemonic
Add Memory Byte and Carry Bit to Accumulator	ADC
Add Memory Byte to Accumulator	ADD
AND Memory Byte with Accumulator	AND
Bit Test Accumulator	BIT
Compare Accumulator	CMP
Compare Index Register with Memory Byte	CPX
EXCLUSIVE OR Accumulator with Memory Byte	EOR
Load Accumulator with Memory Byte	LDA
Load Index Register with Memory Byte	LDX
Multiply	MUL
OR Accumulator with Memory Byte	ORA
Subtract Memory Byte and Carry Bit from Accumulator	SBC
Store Accumulator in Memory	STA
Store Index Register in Memory	STX
Subtract Memory Byte from Accumulator	SUB

12.1.11 Read-Modify-Write Instructions

These instructions read a memory location or a register, modify its contents, and write the modi ed v alue back to the memory location or to the register. The test for negative or zero instruction (TST) is an exception to the read-modify-write sequence because it does not write a replacement value. **Table 12-2** lists the read-modify-write instructions.

Table 12-2. Read-Modify-Write Instructions

Instruction	Mnemonic
Arithmetic Shift Left	ASL
Arithmetic Shift Right	ASR
Clear Bit in Memory	BCLR
Set Bit in Memory	BSET
Clear	CLR
Complement (One's Complement)	COM
Decrement	DEC
Increment	INC
Logical Shift Left	LSL
Logical Shift Right	LSR
Negate (Two's Complement)	NEG
Rotate Left through Carry Bit	ROL
Rotate Right through Carry Bit	ROR
Test for Negative or Zero	TST

12.1.12 Jump/Branch Instructions

Jump instructions allow the CPU to interrupt the normal sequence of the program counter. The unconditional jump instruction (JMP) and the jump to subroutine instruction (JSR) have no register operand. Branch instructions allow the CPU to interrupt the normal sequence of the program counter when a test condition is met. If the test condition is not met, the branch is not performed. All branch instructions use relative addressing.

Bit test and branch instructions cause a branch based on the state of any readable bit in the rst 256 memor y locations. These three-byte instructions use a combination of direct addressing and relative addressing. The direct address of the byte to be tested is in the byte following the opcode. The third byte is the signed offset byte. The CPU nds the conditional br anch destination by adding the

third byte to the program counter if the speci ed bit tests tr ue. The bit to be tested and its condition (set or clear) is part of the opcode. The span of branching is from –128 to +127 from the address of the next location after the branch instruction. The CPU also transfers the tested bit to the carry/borrow bit of the condition code register. **Table 12-3** lists the jump and branch instructions.

Table 12-3. Jump and Branch Instructions

Instruction	Mnemonic
Branch if Carry Bit Clear	BCC
Branch if Carry Bit Set	BCS
Branch if Equal	BEQ
Branch if Half-Carry Bit Clear	BHCC
Branch if Half-Carry Bit Set	BHCS
Branch if Higher	BHI
Branch if Higher or Same	BHS
Branch if $\overline{\text{IRQ}}$ Pin High	BIH
Branch if $\overline{\text{IRQ}}$ Pin Low	BIL
Branch if Lower	BLO
Branch if Lower or Same	BLS
Branch if Interrupt Mask Clear	BMC
Branch if Minus	BMI
Branch if Interrupt Mask Set	BMS
Branch if Not Equal	BNE
Branch if Plus	BPL
Branch Always	BRA
Branch if Bit Clear	BRCLR
Branch Never	BRN
Branch if Bit Set	BRSET
Branch to Subroutine	BSR
Unconditional Jump	JMP
Jump to Subroutine	JSR

12.1.13 Bit Manipulation Instructions

The CPU can set or clear any writable bit in the first 256 bytes of memory. Port registers, port data direction registers, timer registers, and on-chip RAM locations are in the first 256 bytes of memory. The CPU can also test and branch based on the state of any bit in any of the first 256 memory locations. Bit manipulation instructions use direct addressing. **Table 12-4** lists these instructions.

Table 12-4. Bit Manipulation Instructions

Instruction	Mnemonic
Clear Bit	BCLR
Branch if Bit Clear	BRCLR
Branch if Bit Set	BRSET
Set Bit	BSET

12.1.14 Control Instructions

These register reference instructions control CPU operation during program execution. Control instructions, listed in **Table 12-5**, use inherent addressing.

Table 12-5. Control Instructions

Instruction	Mnemonic
Clear Carry Bit	CLC
Clear Interrupt Mask	CLI
No Operation	NOP
Reset Stack Pointer	RSP
Return from Interrupt	RTI
Return from Subroutine	RTS
Set Carry Bit	SEC
Set Interrupt Mask	SEI
Stop Oscillator and Enable $\overline{\text{IRQ}}$ Pin	STOP
Software Interrupt	SWI
Transfer Accumulator to Index Register	TAX
Transfer Index Register to Accumulator	TXA
Stop CPU Clock and Enable Interrupts	WAIT

12.1.15 Instruction Set Summary

Table 12-6 is an alphabetical list of all M68HC05 instructions and shows the effect of each instruction on the condition code register.

Table 12-6. Instruction Set Summary

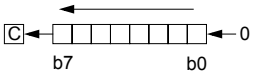
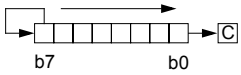
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ADC #*opr* ADC *opr* ADC *opr* ADC *opr*,X ADC *opr*,X ADC ,X	Add with Carry	A ← (A) + (M) + (C)	◊	—	◊	◊	◊	IMM DIR EXT IX2 IX1 IX	A9 B9 C9 D9 E9 F9	ii dd hh ll ee ff ff	2 3 4 5 4 3
ADD #*opr* ADD *opr* ADD *opr* ADD *opr*,X ADD *opr*,X ADD ,X	Add without Carry	A ← (A) + (M)	◊	—	◊	◊	◊	IMM DIR EXT IX2 IX1 IX	AB BB CB DB EB FB	ii dd hh ll ee ff ff	2 3 4 5 4 3
AND #*opr* AND *opr* AN*D opr* AND *opr*,X AND *opr*,X AND ,X	Logical AND	A ← (A) ∧ (M)	—	—	◊	◊	—	IMM DIR EXT IX2 IX1 IX	A4 B4 C4 D4 E4 F4	ii dd hh ll ee ff ff	2 3 4 5 4 3
ASL *opr* ASLA ASLX ASL *opr*,X ASL ,X	Arithmetic Shift Left (Same as LSL)		—	—	◊	◊	◊	DIR INH INH IX1 IX	38 48 58 68 78	dd ff 	5 3 3 6 5
ASR *opr* ASRA ASRX ASR *opr*,X ASR ,X	Arithmetic Shift Right		—	—	◊	◊	◊	DIR INH INH IX1 IX	37 47 57 67 77	dd ff 	5 3 3 6 5
BCC *rel*	Branch if Carry Bit Clear	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BCLR *n opr*	Clear Bit n	Mn ← 0	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	11 13 15 17 19 1B 1D 1F	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BCS *rel*	Branch if Carry Bit Set (Same as BLO)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BEQ *rel*	Branch if Equal	PC ← (PC) + 2 + *rel* ? Z = 1	—	—	—	—	—	REL	27	rr	3
BHCC *rel*	Branch if Half-Carry Bit Clear	PC ← (PC) + 2 + *rel* ? H = 0	—	—	—	—	—	REL	28	rr	3

Table 12-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BHCS *rel*	Branch if Half-Carry Bit Set	PC ← (PC) + 2 + *rel* ? H = 1	—	—	—	—	—	REL	29	rr	3
BHI *rel*	Branch if Higher	PC ← (PC) + 2 + *rel* ? C ∨ Z = 0	—	—	—	—	—	REL	22	rr	3
BHS *rel*	Branch if Higher or Same	PC ← (PC) + 2 + *rel* ? C = 0	—	—	—	—	—	REL	24	rr	3
BIH *rel*	Branch if $\overline{\text{IRQ}}$ Pin High	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 1	—	—	—	—	—	REL	2F	rr	3
BIL *rel*	Branch if $\overline{\text{IRQ}}$ Pin Low	PC ← (PC) + 2 + *rel* ? $\overline{\text{IRQ}}$ = 0	—	—	—	—	—	REL	2E	rr	3
BIT #*opr* BIT *opr* BIT *opr* BIT *opr*,X BIT *opr*,X BIT ,X	Bit Test Accumulator with Memory Byte	(A) ∧ (M)	—	—	◊	◊	—	IMM DIR EXT IX2 IX1 IX	A5 B5 C5 D5 E5 F5	ii dd hh ll ee ff ff p	2 3 4 5 4 3
BLO *rel*	Branch if Lower (Same as BCS)	PC ← (PC) + 2 + *rel* ? C = 1	—	—	—	—	—	REL	25	rr	3
BLS *rel*	Branch if Lower or Same	PC ← (PC) + 2 + *rel* ? C ∨ Z = 1	—	—	—	—	—	REL	23	rr	3
BMC *rel*	Branch if Interrupt Mask Clear	PC ← (PC) + 2 + *rel* ? I = 0	—	—	—	—	—	REL	2C	rr	3
BMI *rel*	Branch if Minus	PC ← (PC) + 2 + *rel* ? N = 1	—	—	—	—	—	REL	2B	rr	3
BMS *rel*	Branch if Interrupt Mask Set	PC ← (PC) + 2 + *rel* ? I = 1	—	—	—	—	—	REL	2D	rr	3
BNE *rel*	Branch if Not Equal	PC ← (PC) + 2 + *rel* ? Z = 0	—	—	—	—	—	REL	26	rr	3
BPL *rel*	Branch if Plus	PC ← (PC) + 2 + *rel* ? N = 0	—	—	—	—	—	REL	2A	rr	3
BRA *rel*	Branch Always	PC ← (PC) + 2 + *rel* ? 1 = 1	—	—	—	—	—	REL	20	rr	3
BRCLR *n opr rel*	Branch if bit n clear	PC ← (PC) + 2 + *rel* ? Mn = 0	—	—	—	—	◊	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRSET *n opr rel*	Branch if Bit n Set	PC ← (PC) + 2 + *rel* ? Mn = 1	—	—	—	—	◊	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN *rel*	Branch Never	PC ← (PC) + 2 + *rel* ? 1 = 0	—	—	—	—	—	REL	21	rr	3

Table 12-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
BSET *n opr*	Set Bit n	Mn ← 1	—	—	—	—	—	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	5 5 5 5 5 5 5 5
BSR *rel*	Branch to Subroutine	PC ← (PC) + 2; push (PCL) SP ← (SP) – 1; push (PCH) SP ← (SP) – 1 PC ← (PC) + *rel*	—	—	—	—	—	REL	AD	rr	6
CLC	Clear Carry Bit	C ← 0	—	—	—	—	0	INH	98		2
CLI	Clear Interrupt Mask	I ← 0	—	0	—	—	—	INH	9A		2
CLR *opr* CLRA CLRX CLR *opr*,X CLR ,X	Clear Byte	M ← $00 A ← $00 X ← $00 M ← $00 M ← $00	—	—	0	1	—	DIR INH INH IX1 IX	3F 4F 5F 6F 7F	dd ff 	5 3 3 6 5
CMP #*opr* CMP *opr* CMP *opr* CMP *opr*,X CMP *opr*,X CMP ,X	Compare Accumulator with Memory Byte	(A) – (M)	—	—	◊	◊	◊	IMM DIR EXT IX2 IX1 IX	A1 B1 C1 D1 E1 F1	ii dd hh ll ee ff ff 	2 3 4 5 4 3
COM *opr* COMA COMX COM *opr*,X COM ,X	Complement Byte (One's Complement)	M ← ($\overline{M}$) = $FF – (M) A ← ($\overline{A}$) = $FF – (M) X ← ($\overline{X}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M) M ← ($\overline{M}$) = $FF – (M)	—	—	◊	◊	1	DIR INH INH IX1 IX	33 43 53 63 73	dd ff 	5 3 3 6 5
CPX #*opr* CPX *opr* CPX *opr* CPX *opr*,X CPX *opr*,X CPX ,X	Compare Index Register with Memory Byte	(X) – (M)	—	—	◊	◊	1	IMM DIR EXT IX2 IX1 IX	A3 B3 C3 D3 E3 F3	ii dd hh ll ee ff ff 	2 3 4 5 4 3
DEC *opr* DECA DECX DEC *opr*,X DEC ,X	Decrement Byte	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1	—	—	◊	◊	—	DIR INH INH IX1 IX	3A 4A 5A 6A 7A	dd ff 	5 3 3 6 5
EOR #*opr* EOR *opr* EOR *opr* EOR *opr*,X EOR *opr*,X EOR ,X	EXCLUSIVE OR Accumulator with Memory Byte	A ← (A) ⊕ (M)	—	—	◊	◊	—	IMM DIR EXT IX2 IX1 IX	A8 B8 C8 D8 E8 F8	ii dd hh ll ee ff ff 	2 3 4 5 4 3

Table 12-6. Instruction Set Summary (Continued)

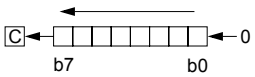
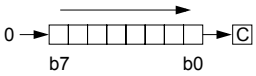
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
INC *opr* INCA INCX INC *opr*,X INC ,X	Increment Byte	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1	—	—	◊	◊	—	DIR INH INH IX1 IX	3C 4C 5C 6C 7C	dd ff 	5 3 3 6 5
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Unconditional Jump	PC ← Jump Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff 	2 3 4 3 2
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + n (n = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Conditional Address	—	—	—	—	—	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff 	5 6 7 6 5
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X	Load Accumulator with Memory Byte	A ← (M)	—	—	◊	◊	—	IMM DIR EXT IX2 IX1 IX	A6 B6 C6 D6 E6 F6	ii dd hh ll ee ff ff 	2 3 4 5 4 3
LDX #*opr* LDX *opr* LDX *opr* LDX *opr*,X LDX *opr*,X LDX ,X	Load Index Register with Memory Byte	X ← (M)	—	—	◊	◊	—	IMM DIR EXT IX2 IX1 IX	AE BE CE DE EE FE	ii dd hh ll ee ff ff 	2 3 4 5 4 3
LSL *opr* LSLA LSLX LSL *opr*,X LSL ,X	Logical Shift Left (Same as ASL)		—	—	◊	◊	◊	DIR INH INH IX1 IX	38 48 58 68 78	dd ff 	5 3 3 6 5
LSR *opr* LSRA LSRX LSR *opr*,X LSR ,X	Logical Shift Right		—	—	0	◊	◊	DIR INH INH IX1 IX	34 44 54 64 74	dd ff 	5 3 3 6 5
MUL	Unsigned Multiply	X : A ← (X) × (A)	0	—	—	—	0	INH	42		11
NEG *opr* NEGA NEGX NEG *opr*,X NEG ,X	Negate Byte (Two's Complement)	M ← –(M) = $00 – (M) A ← –(A) = $00 – (A) X ← –(X) = $00 – (X) M ← –(M) = $00 – (M) M ← –(M) = $00 – (M)	—	—	◊	◊	◊	DIR INH INH IX1 IX	30 40 50 60 70	ii ff 	5 3 3 6 5
NOP	No Operation		—	—	—	—	—	INH	9D		2

Table 12-6. Instruction Set Summary (Continued)

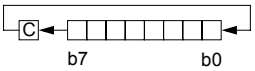
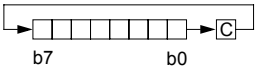
Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
ORA #*opr* ORA *opr* ORA *opr* ORA *opr*,X ORA *opr*,X ORA ,X	Logical OR Accumulator with Memory	A ← (A) ∨ (M)	—	—	◊	◊	—	IMM DIR EXT IX2 IX1 IX	AA BA CA DA EA FA	ii dd hh ll ee ff ff	2 3 4 5 4 3
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X	Rotate Byte Left through Carry Bit		—	—	◊	◊	◊	DIR INH INH IX1 IX	39 49 59 69 79	dd ff	5 3 3 6 5
ROR *opr* RORA RORX ROR *opr*,X ROR ,X	Rotate Byte Right through Carry Bit		—	—	◊	◊	◊	DIR INH INH IX1 IX	36 46 56 66 76	dd ff	5 3 3 6 5
RSP	Reset Stack Pointer	SP ← $00FF	—	—	—	—	—	INH	9C		2
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	◊	◊	◊	◊	◊	INH	80		6
RTS	Return from Subroutine	SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)						INH			
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X	Subtract Memory Byte and Carry Bit from Accumulator	A ← (A) – (M) – (C)	—	—	◊	◊	◊	IMM DIR EXT IX2 IX1 IX	A2 B2 C2 D2 E2 F2	ii dd hh ll ee ff ff	2 3 4 5 4 3
SEC	Set Carry Bit	C ← 1	—	—	—	—	1	INH	99		2
SEI	Set Interrupt Mask	I ← 1	—	1	—	—	—	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X	Store Accumulator in Memory	M ← (A)	—	—	◊	◊	—	DIR EXT IX2 IX1 IX	B7 C7 D7 E7 F7	dd hh ll ee ff ff	4 5 6 5 4
STOP	Stop Oscillator and Enable $\overline{\text{IRQ}}$ Pin		—	0	—	—	—	INH	8E		2
STX *opr* STX *opr* STX *opr*,X STX *opr*,X STX ,X	Store Index Register In Memory	M ← (X)	—	—	◊	◊	—	DIR EXT IX2 IX1 IX	BF CF DF EF FF	dd hh ll ee ff ff	4 5 6 5 4

Table 12-6. Instruction Set Summary (Continued)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			H	I	N	Z	C				
SUB #*opr* SUB *opr* SUB *opr* SUB *opr*,X SUB *opr*,X SUB ,X	Subtract Memory Byte from Accumulator	A ← (A) – (M)	—	—	◊	◊	◊	IMM DIR EXT IX2 IX1 IX	A0 B0 C0 D0 E0 F0	ii dd hh ll ee ff ff	2 3 4 5 4 3
SWI	Software Interrupt	PC ← (PC) + 1; Push (PCL) SP ← (SP) – 1; Push (PCH) SP ← (SP) – 1; Push (X) SP ← (SP) – 1; Push (A) SP ← (SP) – 1; Push (CCR) SP ← (SP) – 1; I ← 1 PCH ← Interrupt Vector High Byte PCL ← Interrupt Vector Low Byte	—	1	—	—	—	INH	83		10
TAX	Transfer Accumulator to Index Register	X ← (A)	—	—	—	—	—	INH	97		2
TST *opr* TSTA TSTX TST *opr*,X TST ,X	Test Memory Byte for Negative or Zero	(M) – $00	—	—	—	—	—	DIR INH INH IX1 IX	3D 4D 5D 6D 7D	dd ff	4 3 3 5 4
TXA	Transfer Index Register to Accumulator	A ← (X)	—	—	—	—	—	INH	9F		2
WAIT	Stop CPU Clock and Enable Interrupts		—	◊	—	—	—	INH	8F		2

A	Accumulator	*opr*	Operand (one or two bytes)
C	Carry/borrow flag	PC	Program counter
CCR	Condition code register	PCH	Program counter high byte
dd	Direct address of operand	PCL	Program counter low byte
dd rr	Direct address of operand and relative offset of branch instruction	REL	Relative addressing mode
DIR	Direct addressing mode	*rel*	Relative program counter offset byte
ee ff	High and low bytes of offset in indexed, 16-bit offset addressing	rr	Relative program counter offset byte
EXT	Extended addressing mode	SP	Stack pointer
ff	Offset byte in indexed, 8-bit offset addressing	X	Index register
H	Half-carry flag	Z	Zero flag
hh ll	High and low bytes of operand address in extended addressing	#	Immediate value
I	Interrupt mask	∧	Logical AND
ii	Immediate operand byte	∨	Logical OR
IMM	Immediate addressing mode	⊕	Logical EXCLUSIVE OR
INH	Inherent addressing mode	( )	Contents of
IX	Indexed, no offset addressing mode	–( )	Negation (two's complement)
IX1	Indexed, 8-bit offset addressing mode	←	Loaded with
IX2	Indexed, 16-bit offset addressing mode	?	If
M	Memory location	:	Concatenated with
N	Negative flag	↕	Set or cleared
n	Any bit	—	Not affected

Table 12-7. Opcode Map

	Bit Manipulation		Branch	Read-Modify-Write					Control		Register/Memory						
	DIR	DIR	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1	IX	
MSB / LSB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	MSB / LSB
0	BRSET0 5 3 DIR	BSET0 5 2 DIR	BRA 3 2 REL	NEG 5 2 DIR	NEGA 3 1 INH	NEGX 3 1 INH	NEG 6 2 IX1	NEG 5 1 IX	RTI 9 1 INH		SUB 2 2 IMM	SUB 3 2 DIR	SUB 4 3 EXT	SUB 5 3 IX2	SUB 4 2 IX1	SUB 3 1 IX	0
1	BRCLR0 5 3 DIR	BCLR0 5 2 DIR	BRN 3 2 REL						RTS 6 1 INH		CMP 2 2 IMM	CMP 3 2 DIR	CMP 4 3 EXT	CMP 5 3 IX2	CMP 4 2 IX1	CMP 3 1 IX	1
2	BRSET1 5 3 DIR	BSET1 5 2 DIR	BHI 3 2 REL		MUL 11 1 INH						SBC 2 2 IMM	SBC 3 2 DIR	SBC 4 3 EXT	SBC 5 3 IX2	SBC 4 2 IX1	SBC 3 1 IX	2
3	BRCLR1 5 3 DIR	BCLR1 5 2 DIR	BLS 3 2 REL	COM 5 2 DIR	COMA 3 1 INH	COMX 3 1 INH	COM 6 2 IX1	COM 5 1 IX	SWI 10 1 INH		CPX 2 2 IMM	CPX 3 2 DIR	CPX 4 3 EXT	CPX 5 3 IX2	CPX 4 2 IX1	CPX 3 1 IX	3
4	BRSET2 5 3 DIR	BSET2 5 2 DIR	BCC 3 2 REL	LSR 5 2 DIR	LSRA 3 1 INH	LSRX 3 1 INH	LSR 6 2 IX1	LSR 5 1 IX			AND 2 2 IMM	AND 3 2 DIR	AND 4 3 EXT	AND 5 3 IX2	AND 4 2 IX1	AND 3 1 IX	4
5	BRCLR2 5 3 DIR	BCLR2 5 2 DIR	BCS/BLO 3 2 REL								BIT 2 2 IMM	BIT 3 2 DIR	BIT 4 3 EXT	BIT 5 3 IX2	BIT 4 2 IX1	BIT 3 1 IX	5
6	BRSET3 5 3 DIR	BSET3 5 2 DIR	BNE 3 2 REL	ROR 5 2 DIR	RORA 3 1 INH	RORX 3 1 INH	ROR 6 2 IX1	ROR 5 1 IX			LDA 2 2 IMM	LDA 3 2 DIR	LDA 4 3 EXT	LDA 5 3 IX2	LDA 4 2 IX1	LDA 3 1 IX	6
7	BRCLR3 5 3 DIR	BCLR3 5 2 DIR	BEQ 3 2 REL	ASR 5 2 DIR	ASRA 3 1 INH	ASRX 3 1 INH	ASR 6 2 IX1	ASR 5 1 IX		TAX 2 1 INH		STA 4 2 DIR	STA 5 3 EXT	STA 6 3 IX2	STA 5 2 IX1	STA 4 1 IX	7
8	BRSET4 5 3 DIR	BSET4 5 2 DIR	BHCC 3 2 REL	ASL/LSL 5 2 DIR	ASLA/LSLA 3 1 INH	ASLX/LSLX 3 1 INH	ASL/LSL 6 2 IX1	ASL/LSL 5 1 IX		CLC 2 1 INH	EOR 2 2 IMM	EOR 3 2 DIR	EOR 4 3 EXT	EOR 5 3 IX2	EOR 4 2 IX1	EOR 3 1 IX	8
9	BRCLR4 5 3 DIR	BCLR4 5 2 DIR	BHCS 3 2 REL	ROL 5 2 DIR	ROLA 3 1 INH	ROLX 3 1 INH	ROL 6 2 IX1	ROL 5 1 IX		SEC 2 1 INH	ADC 2 2 IMM	ADC 3 2 DIR	ADC 4 3 EXT	ADC 5 3 IX2	ADC 4 2 IX1	ADC 3 1 IX	9
A	BRSET5 5 3 DIR	BSET5 5 2 DIR	BPL 3 2 REL	DEC 5 2 DIR	DECA 3 1 INH	DECX 3 1 INH	DEC 6 2 IX1	DEC 5 1 IX		CLI 2 1 INH	ORA 2 2 IMM	ORA 3 2 DIR	ORA 4 3 EXT	ORA 5 3 IX2	ORA 4 2 IX1	ORA 3 1 IX	A
B	BRCLR5 5 3 DIR	BCLR5 5 2 DIR	BMI 3 2 REL							SEI 2 1 INH	ADD 2 2 IMM	ADD 3 2 DIR	ADD 4 3 EXT	ADD 5 3 IX2	ADD 4 2 IX1	ADD 3 1 IX	B
C	BRSET6 5 3 DIR	BSET6 5 2 DIR	BMC 3 2 REL	INC 5 2 DIR	INCA 3 1 INH	INCX 3 1 INH	INC 6 2 IX1	INC 5 1 IX		RSP 2 1 INH		JMP 2 2 DIR	JMP 3 3 EXT	JMP 4 3 IX2	JMP 3 2 IX1	JMP 2 1 IX	C
D	BRCLR6 5 3 DIR	BCLR6 5 2 DIR	BMS 3 2 REL	TST 4 2 DIR	TSTA 3 1 INH	TSTX 3 1 INH	TST 5 2 IX1	TST 4 1 IX		NOP 2 1 INH	BSR 6 2 REL	JSR 5 2 DIR	JSR 6 3 EXT	JSR 7 3 IX2	JSR 6 2 IX1	JSR 5 1 IX	D
E	BRSET7 5 3 DIR	BSET7 5 2 DIR	BIL 3 2 REL						STOP 2 1 INH		LDX 2 2 IMM	LDX 3 2 DIR	LDX 4 3 EXT	LDX 5 3 IX2	LDX 4 2 IX1	LDX 3 1 IX	E
F	BRCLR7 5 3 DIR	BCLR7 5 2 DIR	BIH 3 2 REL	CLR 5 2 DIR	CLRA 3 1 INH	CLRX 3 1 INH	CLR 6 2 IX1	CLR 5 1 IX	WAIT 2 1 INH	TXA 2 1 INH		STX 4 2 DIR	STX 5 3 EXT	STX 6 3 IX2	STX 5 2 IX1	STX 4 1 IX	F

INH = Inherent
IMM = Immediate
DIR = Direct
EXT = Extended
REL = Relative
IX = Indexed, No Offset
IX1 = Indexed, 8-Bit Offset
IX2 = Indexed, 16-Bit Offset

LSB of Opcode in Hexadecimal

MSB / LSB	0
0	5 BRSET0 3 DIR

MSB of Opcode in Hexadecimal
Number of Cycles
Opcode Mnemonic
Number of Bytes/Addressing Mode

SECTION 13
ELECTRICAL SPECIFICATIONS

This section contains the electrical and timing specifications for the MC68HC05PL4.

13.1 MAXIMUM RATINGS

Maximum ratings are the extreme limits the device can be exposed to without causing permanent damage to the chip. The device is **not** intended to operate at these conditions. The MCU contains circuitry that protects the inputs against damage from high static voltages; however, do not apply voltages higher than those shown below. Keep V_{IN} and V_{OUT} within the range from V_{SS} to V_{DD}. Connect unused inputs to the appropriate logical voltage level, either V_{SS} or V_{DD}.

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	–0.3 to +7.0	V
Bootloader/Self-Check Mode ($\overline{IRQ}$ Pin Only)	V_{IN}	V_{SS}–0.3 to 17	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Junction Temperature	T_J	+150	°C
Storage Temperature Range	T_{STG}	–65 to +150	°C

13.2 OPERATING TEMPERATURE RANGE

Characteristic	Symbol	Value	Unit
Operating Temperature Range MC68HC05PL4	 T_A	T_L to T_H –40 to +80	 °C

13.3 THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance SOIC PDIP	 θ_{JA} θ_{JA}	 60 60	 °C/W °C/W

13.4 SUPPLY CURRENT CHARACTERISTICS

Characteristic	Symbol	Min	Typ	Max	Unit
V_{DD} = 4.4 to 3.6 V					
Internal RC (about 500kHz)					
Run	I_{DD}	—	394	—	µA
Wait	I_{DD}	—	36	—	µA
Stop	I_{DD}	—	5	—	µA
External Crystal/Ceramic Resonator @ 5.12MHz					
Run	I_{DD}	—	2.816	—	mA
Wait	I_{DD}	—	348	—	µA
Stop	I_{DD}	—	5	—	µA
V_{DD} = 2.5 to 2.0 V					
Internal RC (about 500kHz)					
Run	I_{DD}	—	128	—	µA
Wait	I_{DD}	—	16	—	µA
Stop	I_{DD}	—	3	—	µA
External Crystal/Ceramic Resonator @ 2MHz					
Run	I_{DD}	—	560	—	µA
Wait	I_{DD}	—	66	—	µA
Stop	I_{DD}	—	3	—	µA

NOTES:

1. V_{DD} as indicated, V_{SS} = 0 V, $T_L \leq T_A \leq T_H$, unless otherwise noted.
2. All values shown re ect a verage measurements.
3. Typical values at midpoint of voltage range, 25°C only.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source to OSC1 pin or internal oscillator, all inputs 0.2 VDC from either supply rail (V_{DD} or V_{SS}); no DC loads, less than 50 pF on all outputs, C_L = 20pF on OSC2.
5. Wait, Stop I_{DD}: All ports con gured as inputs , V_{IL} = 0.2 VDC, V_{IH} = V_{DD} – 0.2 VDC.
6. Stop I_{DD} measured with OSC1 = V_{DD}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

13.5 DC ELECTRICAL CHARACTERISTICS (4V)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage I_{load} = 10µA I_{load} = –10µA	 V_{OL} V_{OH}	 — V_{DD}–0.1	 — —	 0.1 —	 V V
Output High Voltage (I_{load} = –0.8 mA) PA0:6, PB0:7, PC0:7, PD0:3, $\overline{RESET}$	V_{OH}	V_{DD}–0.8	—	—	V
Output Low Voltage (I_{load} = 1.6 mA) PA0:6, PB0:7, PC0:7, PD0:3, $\overline{RESET}$ (I_{load} = 10 mA) LED/$\overline{IRQ}$/V_{PP}	V_{OL} V_{OL}	— —	0.15 0.20	0.4 0.4	V V
High Sink Current (V_{OL} = 0.4) Sink current per pin, PA5, PA6 Sink current total for PC4:7 pins	 I_{OL} I_{OL}	 — —	 9 9	 10 10	 mA mA
Input High Voltage PA0:6, PB0:7, PC0:7, PD0:3, $\overline{RESET}$, LED/$\overline{IRQ}$/V_{PP}	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA0:6, PB0:7, PC0:7,PD0:3, $\overline{RESET}$,LED/$\overline{IRQ}$/V_{PP}	V_{IL}	V_{SS}	—	0.3 x V_{DD}	V
Input Current (with pulldowns disabled) PA0:6, PB0:7, PC0:7, PD0:3, $\overline{RESET}$, LED/$\overline{IRQ}$/V_{PP}	I_{IN}	—	—	±1	µA
I/O Ports High-Z Leakage Current PA0:6, PB0:7, PC0:7, PD0:3	I_{OZ}	—	—	±10	µA
Input Pulldown Current (V_{DD} = 4.0V) PB0:7	I_{IL}	—	34	60	µA
Internal Pull-Up for PB0:7	R	—	110	—	kΩ

NOTES:

1. V_{DD} = 4.0V, V_{SS} = 0 V, $T_L \leq T_A \leq T_H$, unless otherwise noted.
2. All values shown re ect a verage measurements.
3. Typical values at midpoint of voltage range, 25°C only.

13.6 DC ELECTRICAL CHARACTERISTICS (2V)

Characteristic	Symbol	Min	Typ	Max	Unit
Output Voltage I_{load} = 10µA I_{load} = –10µA	 V_{OL} V_{OH}	 — V_{DD}–0.1	 — —	 0.1 —	 V V
Output High Voltage (I_{load} = –0.8 mA) PA0:6, PB0:7, PC0:7, PD0:3, $\overline{RESET}$	V_{OH}	V_{DD}–0.3	—	—	V
Output Low Voltage (I_{load} = 1.6 mA) PA0:6, PB0:7, PC0:7, PD0:3, $\overline{RESET}$ (I_{load} = 10 mA) LED/$\overline{IRQ}$/V_{PP}	 V_{OL} V_{OL}	 — —	 0.15 0.30	 — —	 V V
High Sink Current (V_{OL} = 0.4) Sink current per pin, PA5, PA6 Sink current total for PC4:7 pins	 I_{OL} I_{OL}	 — —	 3 3	 4 4	 mA mA
Input High Voltage PA0:6, PB0:7, PC0:7, PD0:3, $\overline{RESET}$, LED/$\overline{IRQ}$/V_{PP}	V_{IH}	0.7 x V_{DD}	—	V_{DD}	V
Input Low Voltage PA0:6, PB0:7, PC0:7,PD0:3, $\overline{RESET}$,LED/$\overline{IRQ}$/V_{PP}	V_{IL}	V_{SS}	—	0.2 x V_{DD}	V
Input Current (with pulldowns disabled) PA0:6, PB0:7, PC0:7, PD0:3, $\overline{RESET}$, LED/$\overline{IRQ}$/V_{PP}	I_{IN}	—	—	±1	µA
I/O Ports High-Z Leakage Current PA0:6, PB0:7, PC0:7, PD0:3	I_{OZ}	—	—	±10	µA
Input Pulldown Current (V_{DD} = 4.0V) PB0:7	I_{IL}	—	6	11	µA
Internal Pull-Up for PB0:7	R	—	330	—	kΩ

NOTES:

1. V_{DD} = 2.0V, V_{SS} = 0 V, $T_L \leq T_A \leq T_H$, unless otherwise noted.
2. All values shown re ect a verage measurements.
3. Typical values at midpoint of voltage range, 25°C only.

13.7 CONTROL TIMING (4V)

Characteristic	Symbol	Min	Max	Unit
Frequency of Oscillation (OSC) RC Oscillator Option Crystal Oscillator Option External Clock Source	 f_{OSC} f_{OSC} f_{OSC}	 200 0.1 DC	 500 5.12 5.12	 kHz MHz MHz
Internal Operating Frequency, Crystal or External Clock (f_{OSC}/2) RC Oscillator Option Crystal Oscillator Option External Clock Source	 f_{OP} f_{OP} f_{OP}	 100 0.05 DC	 250 2.56 2.56	 kHz MHz MHz
Cycle Time RC Oscillator Option External oscillator or clock source	 t_{CYC} t_{CYC}	 4 0.39	 — —	 µs µs
OSC1 Pulse Width (external clock input)	t_{OH},t_{OL}	195	—	ns
Timer Resolution Input Capture (TCAP) pulse width	 t_{RESL} t_{TH}, t_{TL}	 4 284	 — —	 t_{CYC} ns
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	284	—	ns
Interrupt Pulse Period	t_{ILIL}	see note 2	—	t_{CYC}

NOTES:

1. V_{DD} = 4.0V, V_{SS} = 0 V, $T_L \leq T_A \leq T_H$, unless otherwise noted.
2. The minimum period TILIL should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{CYC}.

13.8 CONTROL TIMING (2V)

Characteristic	Symbol	Min	Max	Unit
Frequency of Oscillation (OSC) RC Oscillator Option Crystal Oscillator Option External Clock Source	 f_{OSC} f_{OSC} f_{OSC}	 200 0.1 DC	 500 2 2	 kHz MHz MHz
Internal Operating Frequency, Crystal or External Clock (f_{OSC}/2) RC Oscillator Option Crystal Oscillator Option External Clock Source	 f_{OP} f_{OP} f_{OP}	 100 0.05 DC	 250 1 1	 kHz MHz MHz
Cycle Time RC Oscillator Option External oscillator or clock source	 t_{CYC} t_{CYC}	 4 1	 — —	 µs µs
OSC1 Pulse Width (external clock input)	t_{OH},t_{OL}	5	—	ns
Timer Resolution Input Capture (TCAP) pulse width	 t_{RESL} t_{TH}, t_{TL}	 4 284	 — —	 t_{CYC} ns
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	284	—	ns
Interrupt Pulse Period	t_{ILIL}	see note 2	—	t_{CYC}

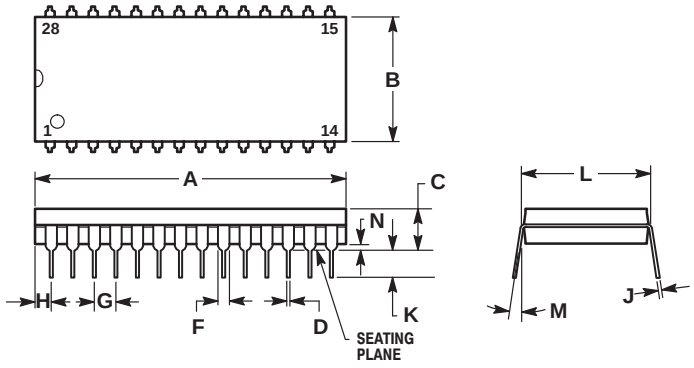
NOTES:

1. V_{DD} = 2.0V, V_{SS} = 0 V, $T_L \leq T_A \leq T_H$, unless otherwise noted.
2. The minimum period TILIL should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{CYC}.

SECTION 14
MECHANICAL SPECIFICATIONS

This section provides the mechanical dimensions for the 28-pin PDIP, 28-pin SOIC, and 28-pin SSOP packages.

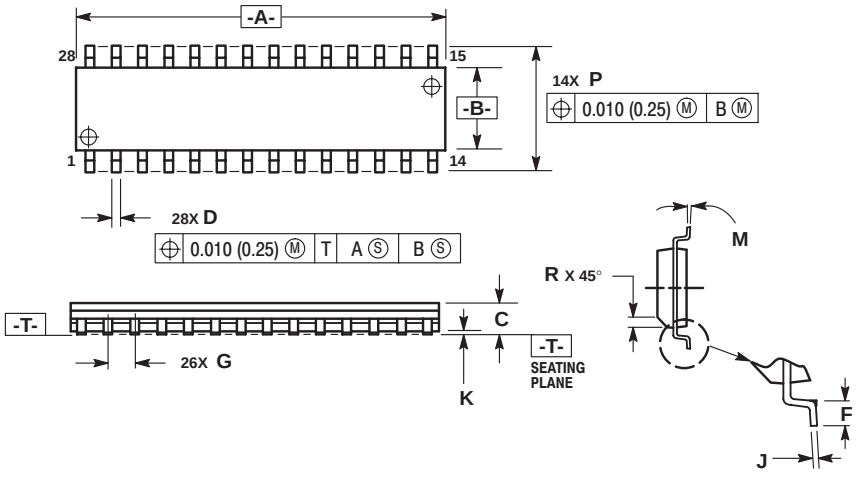
14.1 28-PIN PDIP (CASE 710)



NOTES:
1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25mm (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	36.45	37.21	1.435	1.465
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.36	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040

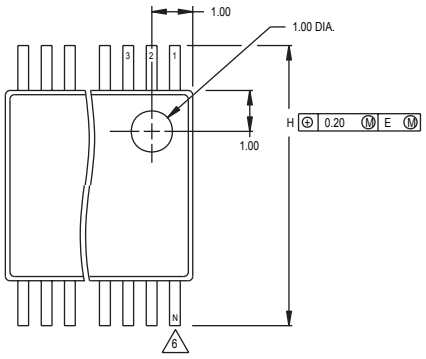
14.2 28-PIN SOIC (CASE 751F)



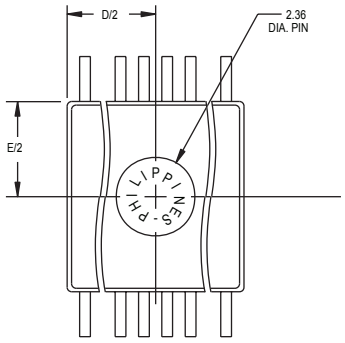
NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DIMENSION A AND B DO NOT INCLUDE MOLD PROTRUSION.
4. MAXIMUM MOLD PROTRUSION 0.15 (0.006) PER SIDE.
5. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.13 (0.005) TOTAL IN EXCESS OF D DIMENSION AT MAXIMUM MATERIAL CONDITION.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	17.80	18.05	0.701	0.711
B	7.40	7.60	0.292	0.299
C	2.35	2.65	0.093	0.104
D	0.35	0.49	0.014	0.019
F	0.41	0.90	0.016	0.035
G	1.27 BSC		0.050 BSC	
J	0.23	0.32	0.009	0.013
K	0.13	0.29	0.005	0.011
M	0°	8°	0°	8°
P	10.05	10.55	0.395	0.415
R	0.25	0.75	0.010	0.029

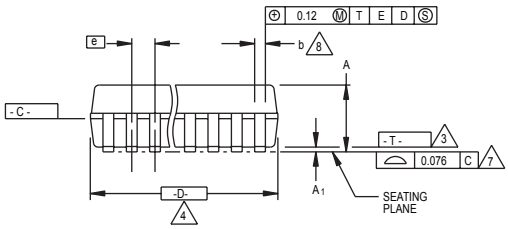
14.3 28-PIN SSOP



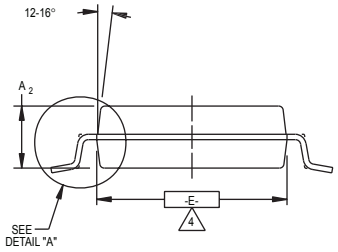
TOP VIEW



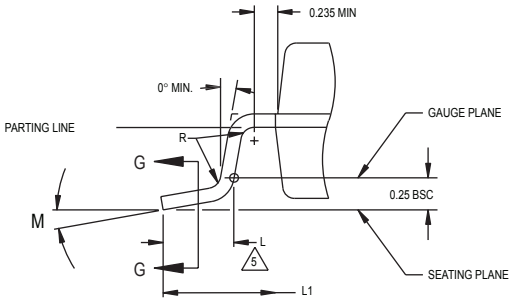
BOTTOM VIEW



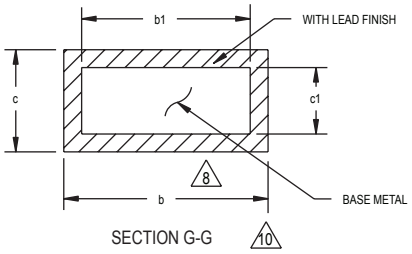
SIDE VIEW



END VIEW



DETAIL 'A'



SECTION G-G 10

NOTES:

1. MAXIMUM DIE THICKNESS ALLOWABLE IS 0.43mm (.017 INCHES).
2. DIMENSIONING & TOLERANCES PER ANSI.Y14.5M-1982.
3. "T" IS A REFERENCE DATUM.
4. "D" & "E" ARE REFERENCE DATUMS AND DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS, BUT DO INCLUDE MOLD MISMATCH AND ARE MEASURED AT THE PARTING LINE, MOLD FLASH OR PROTRUSIONS SHALL NOT EXCEED 0.15mm PER SIDE.
5. DIMENSION IS THE LENGTH OF TERMINAL FOR SOLDERING TO A SUBSTRATE.
6. TERMINAL POSITIONS ARE SHOWN FOR REFERENCE ONLY.
7. FORMED LEADS SHALL BE PLANAR WITH RESPECT TO ONE ANOTHER WITHIN 0.08mm AT SEATING PLANE.
8. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION/INTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.13mm TOTAL IN EXCESS OF b DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR INTRUSION SHALL NOT REDUCE DIMENSION b BY MORE THAN 0.07mm AT LEAST MATERIAL CONDITION.
9. CONTROLLING DIMENSION: MILLIMETERS.
10. THESE DIMENSIONS APPLY TO THE FLAT SECTION OF THE LEAD BETWEEN 0.10 AND 0.25mm FROM LEAD TIPS.
11. THIS PACKAGE OUTLINE DRAWING COMPLIES WITH JEDEC SPECIFICATION NO. MO-150 FOR THE LEAD COUNTS SHOWN

SYMBOL	DIMENSIONS IN MM			DIMENSIONS IN INCH			NOTE
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.	
A	1.73	1.86	1.99	.068	.073	.078	
A1	0.05	0.13	0.21	.002	.005	.008	
A2	1.68	1.73	1.78	.066	.068	.070	
b	0.25	—	0.38	.010	—	.015	8,10
b1	0.25	0.30	0.33	.010	.012	.013	10
c	0.09	—	0.20	.004	—	.008	10
c1	0.09	0.15	0.16	.004	.006	.006	10
D	10.07	10.20	10.33	.397	.402	.407	4
E	5.20	5.30	5.38	.205	.209	.212	4
e		0.65 BSC			.0256 BSC		
H	7.65	7.80	7.90	.301	.307	.311	
L	0.63	0.75	0.95	.025	.030	.037	5
L1		1.25 REF.			.049 REF.		
N		28			28		6
M	0	4	8	0	4	8	
R	0.09	0.15		.004	.006		

APPENDIX A
MC68HC705PL4

This appendix describes the MC68HC705PL4 and MC68HC705PL4B, the emulation parts for MC68HC05PL4 and MC68HC05PL4B respectively. The entire MC68HC05PL4 data sheet applies to the MC68HC705PL4 and MC68HC705PL4B, with exceptions outlined in this appendix.

References to MC68HC705PL4 in this appendix refers to both the MC68HC705PL4 and MC68HC705PL4B devices, unless otherwise stated.

A.1 INTRODUCTION

The MC68HC705PL4 is an EPROM version of the MC68HC705PL4, and the MC68HC705PL4B is an EPROM version of the MC68HC705PL4B. Both HC705 parts are used as the emulation part for their MC68HC05 counterparts. Both MC68HC705 parts are functionally identical to their MC68HC05 counterparts, with the exception of the 4k-bytes user ROM is replaced by 4k-bytes user EPROM.

Table A-1. MC68HC705PL4 and MC68HC705PL4B Differences

Device	Pin27
MC68HC705PL4	PA0
MC68HC705PL4B	OSC2

A.2 MEMORY

The MC68HC705PL4 memory map is shown on **Figure A-1**.

A.3 BOOTLOADER MODE

Bootloader mode is entered upon the rising edge of $\overline{\text{RESET}}$ if LED/$\overline{\text{IRQ}}$/V_{PP} pin is at V_{TST} and PB0/KBI0 at V_{DD}. The Bootloader program is masked in the ROM area from $1E00 to $1FEF. This program handles copying of user code from an external EPROM into the on-chip EPROM. The bootload function has to be done from an external EPROM. The bootloader performs one programming pass at 1ms per byte then does a verify pass.

A.4 EPROM PROGRAMMING

Programming the on-chip EPROM is achieved by using the Program Control Register located at address $001E.

Please contact Freescale for programming board availability.

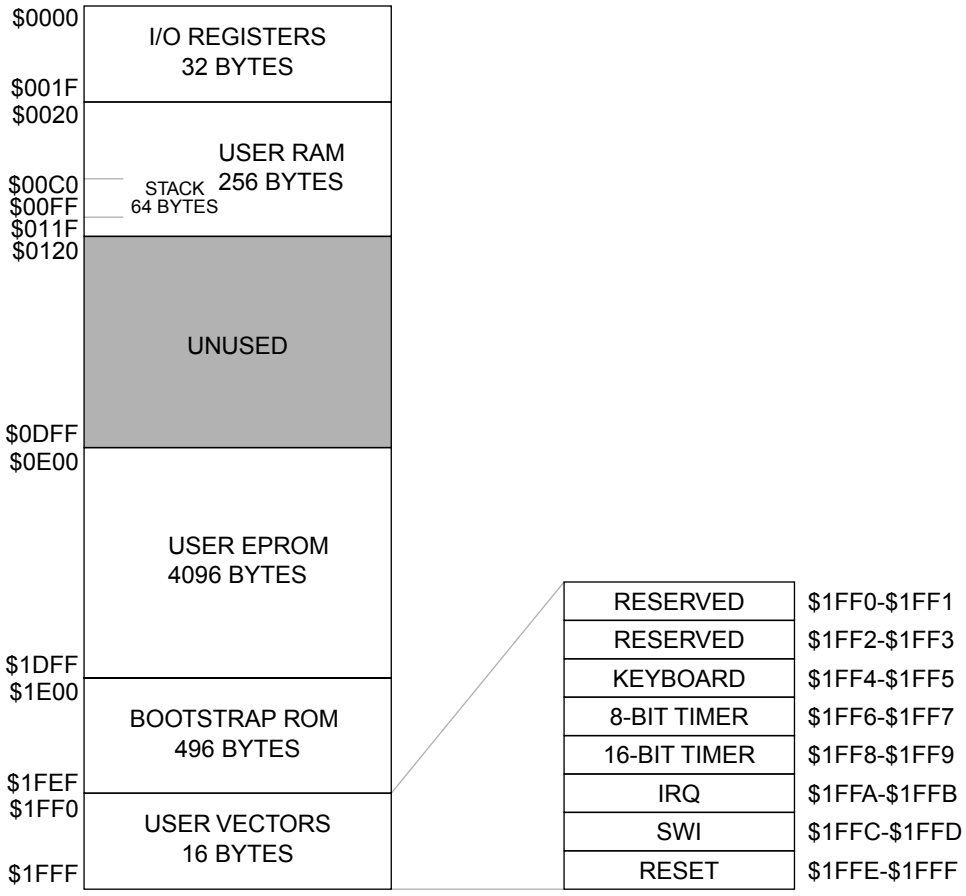


Figure A-1. MC68HC705PL4B Memory Map

A.4.1 EPROM Program Control Register (PCN)

This register is provided for programming the on-chip EPROM in the MC68HC705PL4.

		bit-7	bit-6	bit-5	bit4	bit-3	bit-2	bit1	bit-0
PCR $001E	**Read**	RESERVED						ELAT	PGM
	Write								
	Reset	0	0	0	0	0	0	0	0

ELAT – EPROM LATch control

0 = EPROM address and data bus con gured f or normal reads

1 = EPROM address and data bus con gured f or programming (writes to EPROM cause address and data to be latched). EPROM is in programming mode and cannot be read if ELAT is 1. This bit should not be set when no programming voltage is applied to the V_{PP} pin.

PGM – EPROM ProGraM command

0 = Programming power is switched OFF from EPROM array.
1 = Programming power is switched ON to EPROM array. If ELAT ≠ 1, then PGM = 0.

A.4.2 Programming Sequence

The EPROM programming sequence is:

1. Set the ELAT bit
2. Write the data to the address to be programmed
3. Set the PGM bit
4. Delay for a time t_{PGMR}
5. Clear the PGM bit
6. Clear the ELAT bit

The last two steps must be performed with separate CPU writes.

CAUTION

It is important to remember that an external programming voltage must be applied to the V_{PP} pin while programming, but it should be equal to V_{DD} during normal operations.

Figure A-2 shows the o w required to successfully program the EPROM.

A.5 EPROM PROGRAMMING SPECIFICATIONS

Table A-2. EPROM Programming Electrical Characteristics

(V_{DD} = 4V ±10%, V_{SS} = 0 Vdc, T_A = 0°C to +70°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Programming Voltage	V_{PP}	—	12.5	—	V
Programming Current	I_{PP}	—	5	10	mA
Programming Time per byte	t_{EPGM}	—	1	—	ms

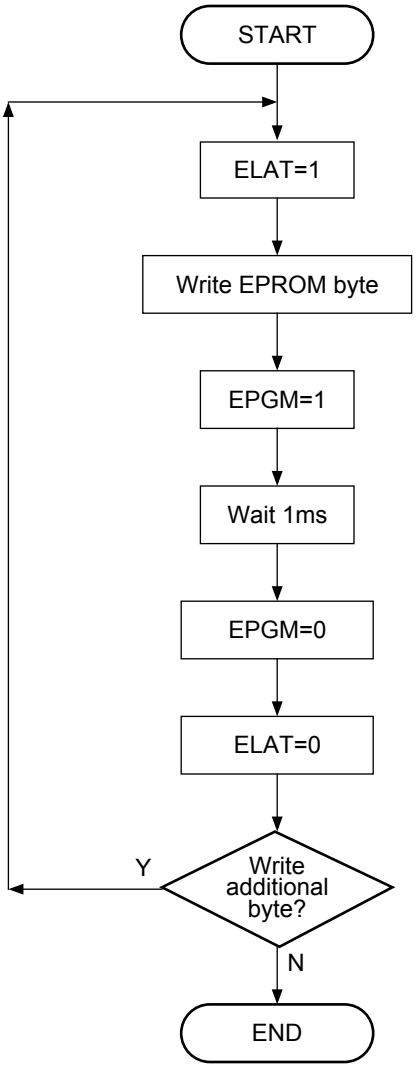


Figure A-2. EPROM Programming Sequence

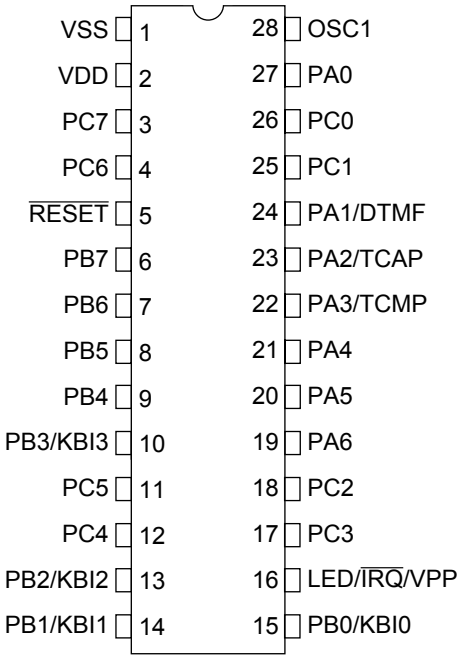


Figure A-3. MC68HC705PL4 Pin Assignment

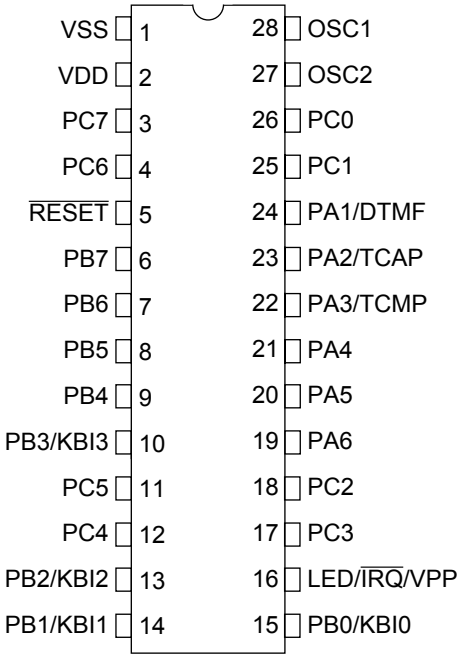


Figure A-4. MC68HC705PL4B Pin Assignment

A.6 SUPPLY CURRENT CHARACTERISTICS

Characteristic	Symbol	Min	Typ	Max	Unit
V_{DD} = 4.4 V to 3.6 V					
Internal RC (about 500kHz)					
Run	I_{DD}	—	966	—	µA
Wait	I_{DD}	—	486	—	µA
Stop	I_{DD}	—	4	—	µA
External Crystal/Ceramic Resonator @ 5.12MHz					
Run	I_{DD}	—	4.398	—	mA
Wait	I_{DD}	—	922	—	µA
Stop	I_{DD}	—	5	—	µA

NOTES:

1. V_{DD} as indicated, V_{SS} = 0 V, $T_L \leq T_A \leq T_H$, unless otherwise noted.
2. All values shown re ect a verage measurements.
3. Typical values at midpoint of voltage range, 25°C only.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source to OSC1 pin or internal oscillator, all inputs 0.2 VDC from either supply rail (V_{DD} or V_{SS}); no DC loads, less than 50 pF on all outputs, C_L = 20pF on OSC2.
5. Wait, Stop I_{DD}: All ports con gured as inputs , V_{IL} = 0.2 VDC, V_{IH} = V_{DD} – 0.2 VDC.
6. Stop I_{DD} measured with OSC1 = V_{DD}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.

Home Page:
www.freescale.com
email:
support@freescale.com
USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130
support@freescale.com
Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com
Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 2666 8080
support.japan@freescale.com
Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com
For Literature Requests Only:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com

RoHS-compliant and/or Pb- free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb- free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale.s Environmental Products program, go to http://www.freescale.com/epp.

Freescale Semiconductor, Inc.