HC05P15GRS/D

MC68HC05P15

SPECIFICATION

Rev. 0.0

(General Release)

Microcontroller Design Center
Munich, Germany

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1
INTRODUCTION

1.1 GENERAL

The MC68HC05P15 is a 28-pin device based on the MC68HC05P5 with an on-chip analog voltage comparator and a two channel PWM. The memory map includes 3072 bytes of user ROM, 64 bytes of user EPROM, and 128 bytes of RAM. The MCU has two I/O ports, A and C, consisting of 15 pins with mask programmable pull-up resistors. Port B consists of 3 pins, 2 input only and 1 output only, shared with an analog voltage comparator. PWM circuitry does not share the output pins with any other module. Port D has 1 input only pin. The MC68HC05P15 has a 16 bit timer, a mask programmable Computer Operating Properly (COP) Watchdog, a mask programmable STOP disable option, and Illegal Address Reset (ILADR) circuitry to insure proper functioning of any system.

1.2 FEATURES

- Low cost
- Low noise
- HC05 Core
- 28 pin package
- On-Chip Oscillator with RC or Crystal/Ceramic Resonator Mask Options
- 3072 Bytes of User ROM including 16 user vector locations
- 64 Bytes of User EPROM with EPROM Programming Register
- 128 Bytes of On-Chip RAM
- 16 Bit Timer
- 2 PWM channels with programmable polarity, period, counter width (8/6 bit)
- one analog voltage comparator with a digital output
- 15 Bidirectional I/O Lines, 3 input-only lines, 3 output-only lines
- User Mode and Bootloader Mode
- Power Saving STOP and WAIT Modes
- Edge-Sensitive or Edge- and Level-Sensitive Interrupt Trigger Mask Option
- Port Pull-up Mask Option for Ports A and C
- Mask option selectable Watchdog Timer (COP)
- Illegal Address Reset (ILADR)
- STOP disable Mask Option
- Internal Resistor and Capacitors for Oscillator Circuitry

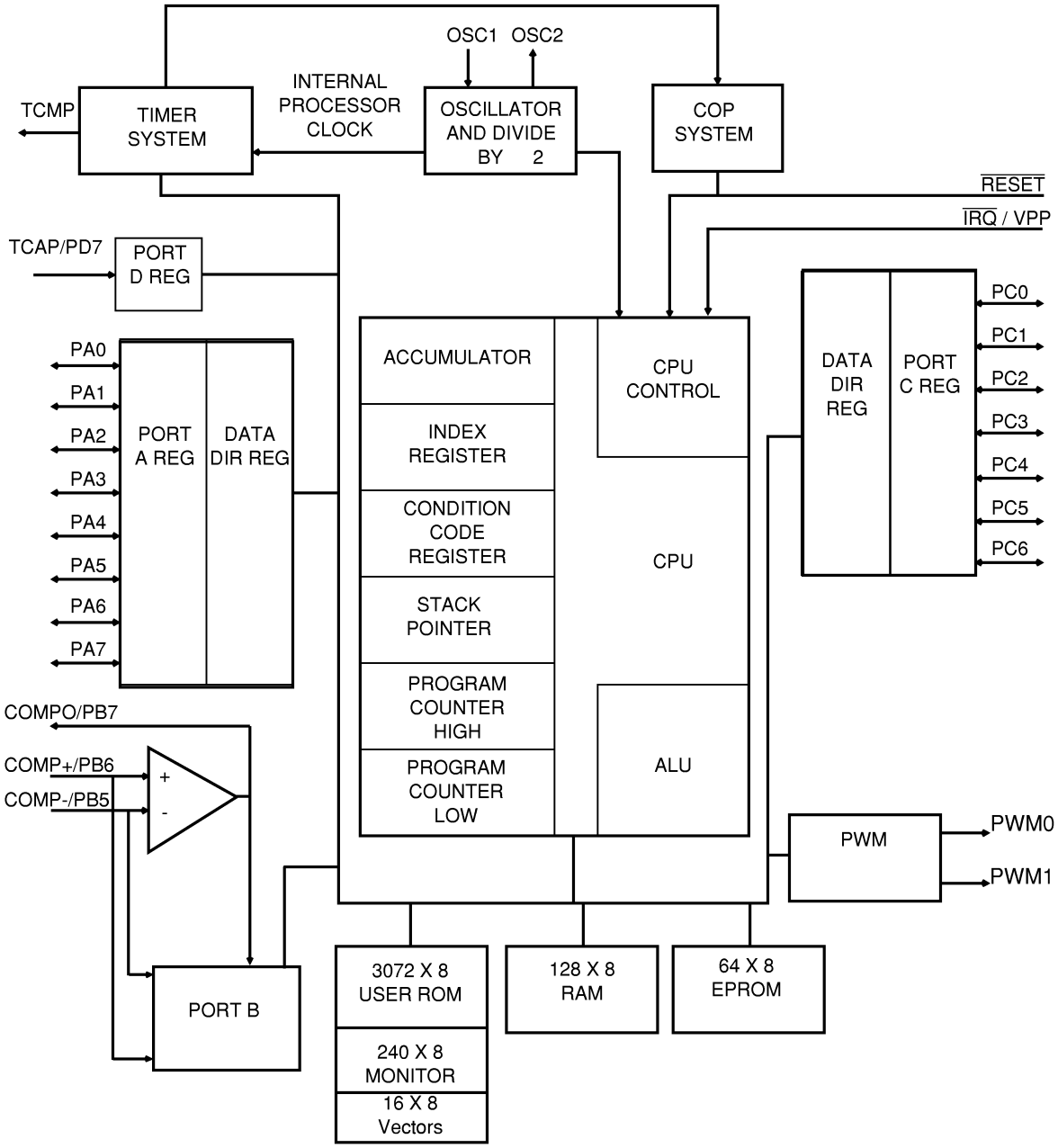


Figure 1-1 Block Diagram for the MC68HC05P15

1.3 MASK OPTIONS

Mask options on the MC68HC05P15 include: CLOCK (RC or Crystal), $\overline{IRQ}$ (edge-sensitive only or edge and level-sensitive), the COP Watchdog Timer (enable/disable), STOP instruction disable, and pull-up resistors (enable/disable) for ports A and C.

1.4 SIGNAL DESCRIPTION

1.4.1 V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is the positive supply, and V_{SS} is ground.

1.4.2 $\overline{IRQ}$/VPP

This pin has a mask option that provides two different choices of interrupt triggering sensitivity. The $\overline{IRQ}$ pin contains an internal Schmitt trigger as part of its input to improve noise immunity. Refer to **Section 4.5 INTERRUPTS** for more detail.

This pin is also used to provide programming voltage (VPP) to the EPROM.

NOTE: Voltage on this pin affects the mode of operation. See **CHAPTER 2 OPERATING MODES. While the EPROM Programming Voltage (VPP) is applied, under no circumstance must a logic zero be applied to the $\overline{RESET}$ Pin!**

1.4.3 OSC1, OSC2

These pins provide control input for an on-chip clock oscillator circuit. A crystal, a ceramic resonator, a resistor/capacitor combination, or an external signal connects to these pins providing a system clock. A mask option selects either a crystal/ceramic resonator or a resistor/capacitor as the frequency determining element. The oscillator frequency is two times the internal bus rate. A 1 MΩ resistor is included internally across OSC1 and OSC2. Two 22 pf capacitors are also included internally. One capacitor from OSC1 to ground the other from OSC2 to ground. These internal components are disconnected if the RC option is chosen over the crystal option.

1.4.4 $\overline{RESET}$

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

1.4.5 TCMP

This pin provides an output for the output compare feature of the on-chip timer system.

1.4.6 PA0-PA7

These eight I/O lines comprise port A. The state of any pin is software programmable and all port A lines are configured as input during power-on or reset. Port A has a Mask Option to enable/disable the pull-ups on the port. The pull-ups are disconnected when the pins are outputs. Refer to **Section 7.5 INPUT/OUTPUT PROGRAMMING** for a detailed description of I/O programming. For details on pull-up resistors see **CHAPTER 11 ELECTRICAL SPECIFICATIONS**.

1.4.7 PB7/CMPO, PB6/CMP+, PB5/CMP-

Port B is a 3-bit port which is shared with the comparator. PB7 is output only, PB5 and PB6 are input only. PB5, PB6, and PB7 are readable internally. The address of the port B data register is $0001. Refer to **Section 7.5 INPUT/OUTPUT PROGRAMMING** for a detailed description of I/O programming. When the comparator module is enabled PB7/CMPO is readable internally at $0001 bit 7. See **CHAPTER 10 COMPARATOR** for CMP+, CMP-, and CMPO for further information on the comparator.

1.4.8 PC0-PC6

Port C is a 7-bit bidirectional port. The address of the port C data register is $0002 and the data direction register is at address $0006. Reset does not affect the data registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode. Port C has a Mask Option to enable/disable the pull-ups on the port. The pull-ups are disconnected when the pins are outputs. Refer to **Section 7.5 INPUT/OUTPUT PROGRAMMING** for a detailed description of I/O programming. See **CHAPTER 11 ELECTRICAL SPECIFICATIONS** for details on pull-up resistors**.**

1.4.9 TCAP/PD7

Port D is a 1-bit port. TCAP/PD7 is input-only shared with the timer input capture. The address of the port D data register is $0003. Reset does not affect the data registers. The TCAP/PD7 pin controls the input capture feature for the on-chip programmable timer. This pin can be read at any time even if the TCAP function is enabled. For detailed information on the TCAP pin see **Section 5.4 INPUT CAPTURE REGISTER**.

CHAPTER 2
OPERATING MODES

The MCU has 2 modes of operation: User Mode and Bootloader Mode. Table 2-1 shows the conditions required to go into each mode, where $V_{TST} = 2xV_{DD}$.

Table 2-1 Operating Mode Conditions

RESET*	$\overline{IRQ}$	TCAP	MODE
↑	V_{SS} to V_{DD}	V_{SS} to V_{DD}	User Mode
	V_{TST}	V_{DD}	Bootloader

2.1 USER MODE

In single chip mode, the address and data buses are not available externally, but there are two I/O ports (A & C), one 3-bit port (B), and one 1-bit port (D). This mode allows the MCU to function as a self-contained microcontroller, with maximum use of the pins for on-chip peripheral functions. All address and data activity occurs within the MCU. User Mode is entered on the rising edge of $\overline{RESET}$ if the $\overline{IRQ}$ pin is within normal operating range.

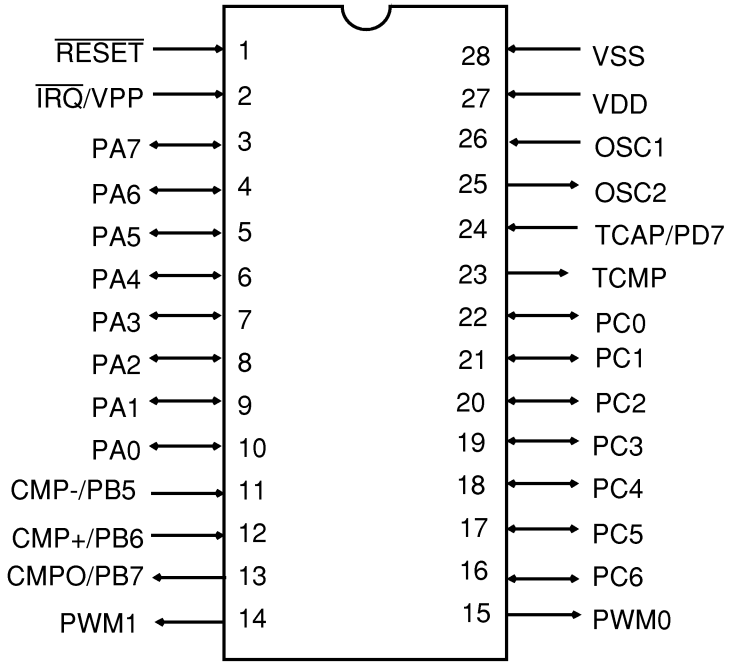


Figure 2-1 User Mode Pinout of the MC68HC05P15

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3
MEMORY

The MC68HC05P15 has a 8 K byte memory map, consisting of user ROM, user RAM, user EPROM, Bootloader ROM, and I/O. See Figure 3-1 and/or Figure 3-2.

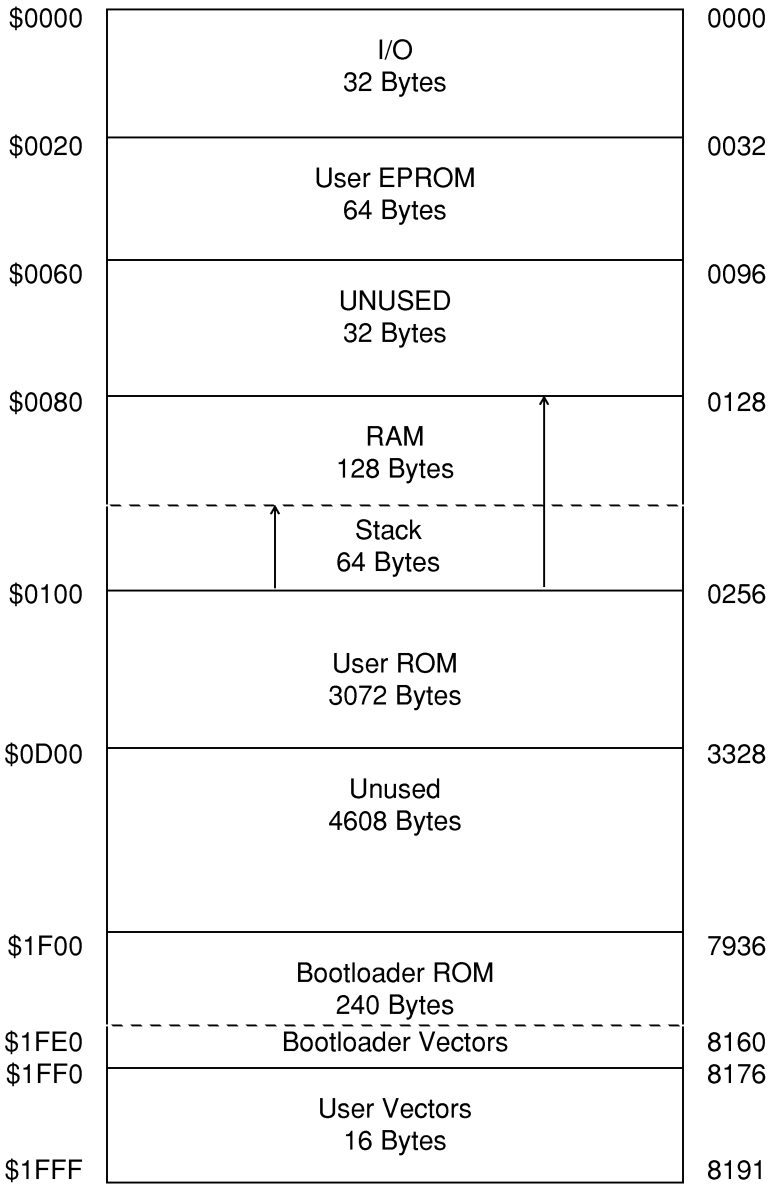


Figure 3-1 The 8K Memory Map of the MC68HC05P15

ADDRESS 0000 TO 001F	DATA							
	7	6	5	4	3	2	1	0
$00 PORT A DATA								
$01 PORT B DATA				0	0	0	0	0
$02 PORT C DATA	0							
$03 PORT D DATA		0	0	0	0	0	0	0
$04 PORT A DDR								
$05 UNUSED	--	--	--	--	--	--	--	--
$06 PORT C DDR	0							
$07 UNUSED	--	--	--	--	--	--	--	--
$08 UNUSED	--	--	--	--	--	--	--	--
$09 UNUSED	--	--	--	--	--	--	--	--
$0A UNUSED	--	--	--	--	--	--	--	--
$0B UNUSED	--	-	--	--	--	--	--	--
$0C COMP CONTROL	--	--	--	--	--	--	--	CEN0
$0D UNUSED	--	--	--	--	--	--	--	--
$0E PWM CONTROL	--	--	--	RAH	POL2	POL1	RA2	RA1
$0F PWM DATA 0								
$10 PWM DATA 1								
$11 UNUSED	--	--	--	--	--	--	--	--
$12 TIMER CONTROL	ICIE	OCIE	TOIE	--	--	--	IEDG	OLVL
$13 TIMER STATUS	ICF	OCF	TOF	--	--	--	--	--
$14 CAPTURE HIGH								
$15 CAPTURE LOW								
$16 COMPARE HIGH								
$17 COMPARE LOW								
$18 COUNTER HIGH								
$19 COUNTER LOW								
$1A DUAL TM HIGH								
$1B DUAL TM LOW								
$1C EPROM PROG	--	--	--	--	--	ELAT	--	EPGM
$1D UNUSED	--	--	--	--	--	--	--	--
$1E UNUSED	--	--	--	--	--	--	--	--
$1F TEST REGISTER	--	MSCAN	ROMON	IOOFF	ILADR	COPEN	TCNT	RAMON

Figure 3-2 I/O Registers

3.1 ROM

The user ROM consists of 48 bytes of page zero ROM from $0020 to $004F, 3072 bytes of ROM from $0100 to $0CFF and 16 bytes of user vectors from $1FF0 to $1FFF. The Bootloader ROM and vectors are located from $1F00 to $1FEF.

3.2 RAM

The user RAM consists of 128 bytes with a shared stack area. The stack begins at address $00FF. The stack pointer can access 64 bytes of RAM in the range $00FF to $00C0.

NOTE: Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

3.3 EPROM/OTPROM

The user EPROM consists of 64 bytes at addresses $0020 to $005F.

OTPROM (one-time programmable ROM) is the same as EPROM except it can be programmed only once and cannot be erased. OTPROM MCUs are shipped in an erased state and are packaged in an opaque plastic package; thus, erasing operations cannot be performed on OTPROM MCUs. See **CHAPTER 8 EPROM**.

THIS PAGE INTENTIONALLY LEFT BLANK

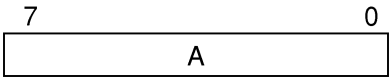
CHAPTER 4
CPU CORE

4.1 REGISTERS

The MCU contains the registers described in the following paragraphs.

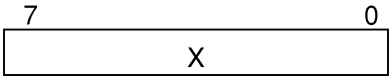
4.1.1 ACCUMULATOR (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.



4.1.2 INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing value to create an effective address. The index register may also be used as a temporary storage area.



4.1.3 CONDITION CODE REGISTER (CCR)

The CCR is a 5-bit register in which four bits are used to indicate the results of the instruction just executed, and the fifth bit indicates whether interrupts are masked. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.

4.1.3.1 H - HALF CARRY

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

4.1.3.2 I - INTERRUPT

When this bit is set, the timer and external interrupt is masked (disabled). If an interrupt occurs while this bit is set, the interrupt is latched and processed as soon as the interrupt bit is cleared.

4.1.3.3 N - NEGATIVE

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative.

4.1.3.4 Z - ZERO

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

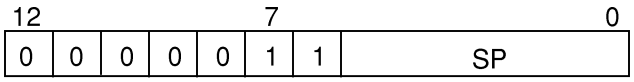
4.1.3.5 C - CARRY/BORROW

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions and during shifts and rotates.

4.1.4 STACK POINTER (SP)

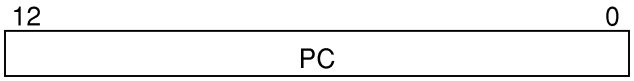
The stack pointer contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $00FF. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

When accessing memory, the seven most significant bits are permanently set to 0000011. These seven bits are appended to the six least significant register bits to produce and address within the range of $00FF to $00C0. Subroutines and interrupts may use up to 64 (decimal) locations. If 64 locations are exceeded, the stack pointer wraps around and loses the previously stored information. A subroutine call occupies two locations on the stack; an interrupt uses five locations.



4.1.5 PROGRAM COUNTER (PC)

The program counter is a 13-bit register that contains the address of the next byte to be fetched.



NOTE: The HC05 CPU core is capable of addressing a 64 K byte memory map. For this implementation, however, the addressing registers are limited to an 8 K byte memory map.

4.2 INSTRUCTION SET

The MCU has a set of 62 basic instructions. They can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. For more information on the instruction set, refer to the M6805 Family User's Manual (M6805UM/AD2) or the MC68HC05C4 Data Sheet (MC68HC05C4/D).

This MCU uses all the instructions available in the M146805 CMOS Family plus one more: the unsigned multiply (MUL) instruction. This instruction allows unsigned multiplication of the contents of the accumulator (A) and the index register (X). The high-order product is then stored in the index register, and the low-order product is stored in the accumulator. A detailed definition of the MUL instruction is shown below.

Operation	X:A X*A			
Description	Multiplies the eight bits in the index register by the eight bits in the accumulator to obtain a 16-bit unsigned number in the concatenated accumulator and index register.			
Condition Codes	H: Cleared I: Not affected N: Not affected Z: Not affected C: Cleared			
Source	MUL			
Form(s)	Addressing Mode Inherent	Cycles 11	Bytes 1	Opcode $42

4.2.1 REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following instruction list.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR
Multiply	MUL

4.2.2 READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (Twos Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST

4.2.3 BRANCH INSTRUCTIONS

This set of instruction branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two-byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
Branch if Higher or Same	BHS
Branch if Carry Set	BCS
Branch if Lower	BLO
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

4.2.4 BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any writable bit which resides in the first 256 bytes of the memory space where all port registers, port DDRs, timer, timer control, EEPROM, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition code register. These instructions are also read-modify-write instructions. Do not bit manipulate write-only locations. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n = 0. . .7)
Branch if bit n is Clear	BRCLR n (n = 0. . .7)
Set Bit n	BSET n (n = 0. . .7)
Clear Bit n	BCLR n (n = 0. . .7)

4.2.5 CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP
Stop	STOP
Wait	WAIT

4.3 ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions; the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. One- or two-byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term “effective address” (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

4.3.1 IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

4.3.2 DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

4.3.3 EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction. When using the Motorola assembler, the user need not specify whether an instruction uses direct or extended addressing. The assembler automatically selects the shortest form of the instruction.

4.3.4 RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. The span of relative addressing is from -126 to +129 from the opcode address. The programmer need not calculate the offset when using the Motorola assembler, since it calculates the proper offset and checks to see that it is within the span of the branch.

4.3.5 INDEXED, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to

move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

4.3.6 INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this two-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 ($1FE is the last location at which the instruction may begin).

4.3.7 INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This address mode can be used in a manner similar to indexed, 8-bit offset except that this three-byte instruction allows tables to be anywhere in memory. As with direct and extended addressing, the Motorola assembler determines the shortest form of indexed addressing.

4.3.8 BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode, and the byte following the opcode specifies the direct addressing of the byte in which the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can by selectively set or cleared with a single two-byte instruction.

4.3.9 BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear), is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single three-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from -125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

4.3.10 INHERENT

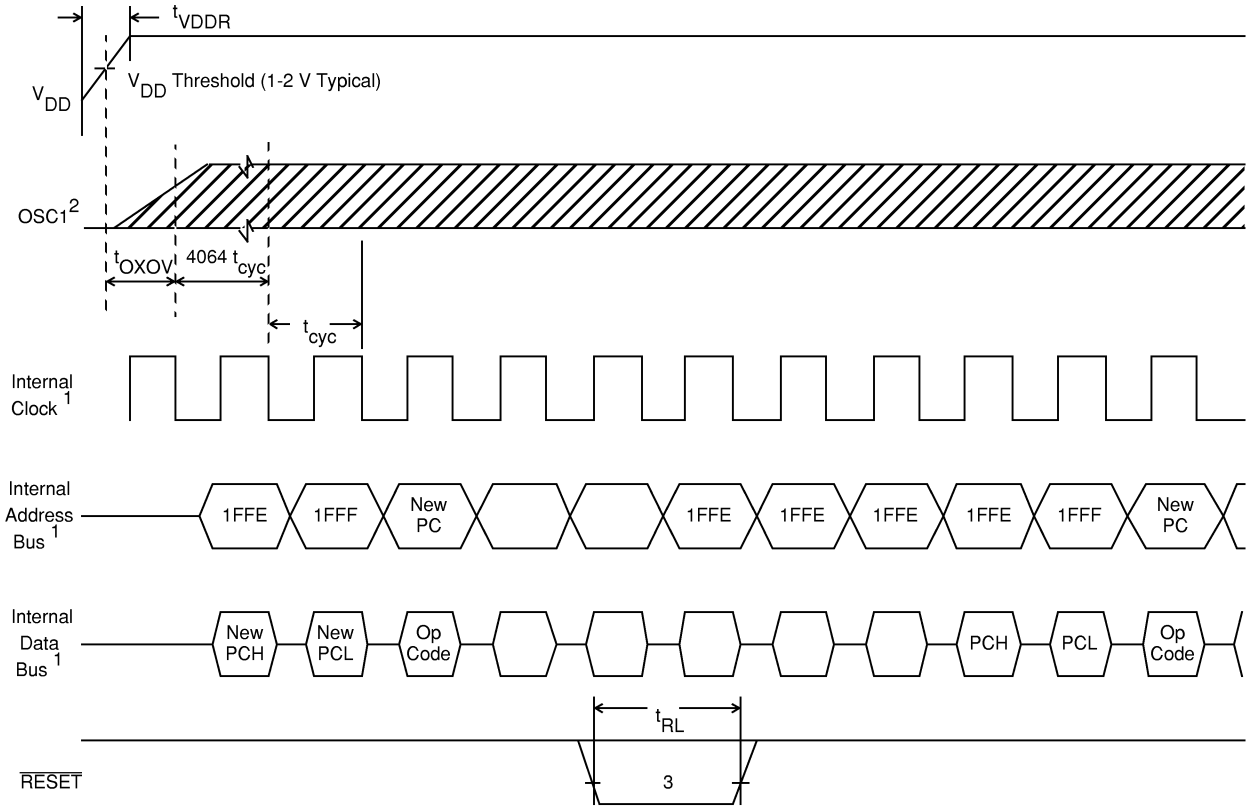
In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

4.4 RESETS

The MCU can be reset three ways: by the initial power-on reset function, by an active low input to the $\overline{\text{RESET}}$ pin, and by a computer operating properly (COP) watchdog-timer time-out.

4.4.1 POWER-ON RESET (POR)

An internal reset is generated on power-up to allow the internal clock generator to stabilize. The power-on reset is strictly for power turn-on conditions and should not be used to detect a drop in the power supply voltage. There is a 4064 internal processor clock cycle (t_{cyc}) oscillator stabilization delay after the oscillator becomes active. If the $\overline{\text{RESET}}$ pin is low at the end of this 4064 cycle delay, the MCU will remain in the $\overline{\text{reset}}$ condition until $\overline{\text{RESET}}$ goes high.



NOTES:
1. Internal timing signal and bus information not available externally.
2. OSC1 line is not meant to represent frequency. It is only used to represent time.
3. The next rising edge of the internal processor clock following the rising edge of $\overline{\text{RESET}}$ initiates the reset sequence.

Figure 4-1 Power-On Reset and $\overline{\text{RESET}}$

4.4.2 RESET PIN

The MCU is reset when a logic zero is applied to the $\overline{\text{RESET}}$ input for a period of one and one-half machine cycles (t_{cyc}).

4.4.3 COMPUTER OPERATING PROPERLY (COP) RESET

The MCU contains a watchdog timer that automatically times out if not reset (cleared) within a specific time by a program reset sequence. If the COP watchdog timer is allowed to time-out, an internal reset is generated to reset the MCU. Because the internal $\overline{\text{RESET}}$ signal is used, the MCU comes out of a COP reset in the same operating mode it was in when the COP time-out was generated.

The COP reset function is enabled or disabled by a mask option.

See **CHAPTER 6 COP - COMPUTER OPERATING PROPERLY**, for more information on the COP.

4.4.4 ILLEGAL ADDRESS RESET

When an opcode fetch occurs at an address which is not in the RAM or ROM the part automatically resets.

4.5 INTERRUPTS

The MCU can be interrupted three different ways: the two maskable hardware interrupts ($\overline{\text{IRQ}}$ and timer) and the nonmaskable software interrupt instruction (SWI).

Interrupts cause the processor to save register contents on the stack and to set the interrupt mask (I bit) to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and normal processing to resume.

Unlike $\overline{\text{RESET}}$, hardware interrupts do not cause the current instruction execution to be halted, but are considered pending until the current instruction is complete.

NOTE: The current instruction is the one already fetched and being operated on.

When the current instruction is complete, the processor checks all pending hardware interrupts. If interrupts are not masked (CCR I bit clear) and if the corresponding interrupt enable bit is set, the processor proceeds with interrupt processing; otherwise, the next instruction is fetched and executed.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction, regardless of the I-bit state.

Table 4-1 lists vector addresses for all interrupt including reset.

Table 4-1 Vector Address for Interrupts and Reset

Register	Flag Name	Interrupts	CPU Interrupt	Vector Address
N/A	N/A	$\overline{\text{Reset}}$	$\overline{\text{RESET}}$	$1FFE-$1FFF
N/A	N/A	Software	SWI	$1FFC-$1FFD
N/A	N/A	External Interrupt	$\overline{\text{IRQ}}$	$1FFA-$1FFB
TSR	ICF	Timer Input Capture	TIMER	$1FF8-$1FF9
TSR	OCF	Timer Output Compare	TIMER	$1FF8-$1FF9
TSR	TOF	Timer Overflow	TIMER	$1FF8-$1FF9

4.5.1 HARDWARE CONTROLLED INTERRUPT SEQUENCE

The following three functions ($\overline{\text{RESET}}$, STOP, and WAIT) are not in the strictest sense an interrupt; however, they are acted upon in a similar manner. Flowcharts for hardware interrupts are shown in Figure 4-2, and for STOP and WAIT in Figure 4-3. A discussion is provided below. When the STOP disable Mask Option is selected all STOP instructions become a NOP

1. $\overline{\text{RESET}}$ - A low input on the $\overline{\text{RESET}}$ input pin causes the program to vector to its starting address which is specified by the contents of memory locations $1FFE and $1FFF. The I bit in the condition code register is also set. Much of the MCU is configured to a known state during this type of reset as previously described in **Section 4.4 RESETS.**
2. STOP - The STOP instruction causes the oscillator to be turned off and the processor to “sleep” until and external interrupt ($\overline{\text{IRQ}}$) or reset occurs. When the STOP disable Mask Option is selected all STOP instructions become a NOP.
3. WAIT - The WAIT instruction causes all processor clocks to stop, but leaves the timer clock running. This “rest” state of the processor can be cleared by reset, an external interrupt ($\overline{\text{IRQ}}$), or Timer interrupt. There are no special wait vectors for these individual interrupts.

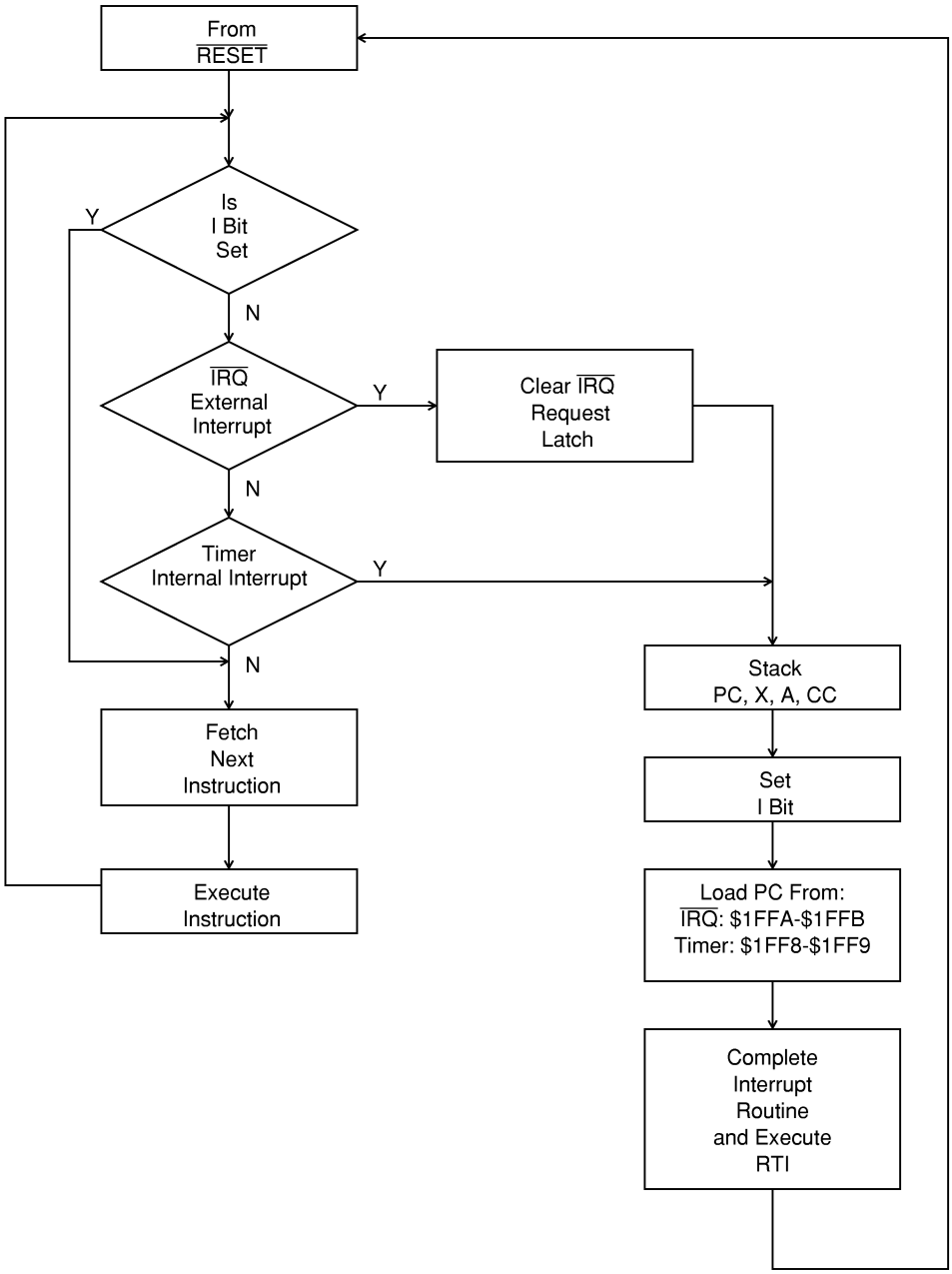


Figure 4-2 Hardware Interrupt Flowchart

4.5.2 TIMER INTERRUPT

There are three different timer interrupt flags that cause a timer interrupt whenever they are set and enabled. The interrupt flags are in the Timer Status Register (TSR), and the enable bits are in the Timer Control Register (TCR). Any of these interrupts will vector to

the same interrupt service routine, located at the address specified by the contents of memory location $1FF8 and $1FF9.

4.5.3 EXTERNAL INTERRUPT

If the interrupt mask bit (I bit) of the CCR is set, all maskable interrupts (internal and external) are disabled. Clearing the I bit enables interrupts. The interrupt request is latched immediately following the falling edge of $\overline{IRQ}$. It is then synchronized internally and serviced as specified by the contents of $1FFA and $1FFB.

Either a level-sensitive and edge-sensitive trigger, or an edge-sensitive-only trigger is available as a mask option.

NOTE: The internal interrupt latch is cleared in the first part of the interrupt service routine; therefore, another external interrupt pulse could be latched and serviced as soon as the I bit is cleared.

4.5.4 SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction and a nonmaskable interrupt: it is executed regardless of the state of the I bit in the CCR. If the I bit is zero (interrupts enabled), SWI executes after interrupts which were pending when the SWI was fetched, but before interrupts generated after the SWI was fetched. The interrupt service routine address is specified by the contents of memory locations $1FFC and $1FFD.

4.6 LOW-POWER MODES

4.6.1 STOP

The STOP instruction places the MCU in its lowest power consumption mode. In STOP mode, the internal oscillator is turned off, halting all internal processing, including timer (and COP Watchdog timer) operation.

During the STOP mode, the TCR bits are altered to remove any pending timer interrupt request and to disable any further timer interrupts. The timer prescaler is cleared. The I bit in the CCR is cleared to enable external interrupts. All other registers and memory remain unaltered. All input/output lines remain unchanged. The processor can be brought out of the STOP mode only by an external interrupt or reset.

When the STOP disable Mask Option is selected all STOP instructions become a NOP.

4.6.2 WAIT

The WAIT instruction places the MCU in a low power consumption mode, but the WAIT mode consumes more power than the STOP mode. All CPU action is suspended, but the timer and the oscillator remain active. Any interrupt or reset (including a COP reset) will cause the MCU to exit the WAIT mode.

During the WAIT mode, the I bit in the CCR is cleared to enable interrupts. All other registers, memory, and input/output lines remain in their previous state. The timer may be enabled to allow a periodic exit from the WAIT mode. WAIT mode must be exited and the COP must be reset to prevent a COP time-out.

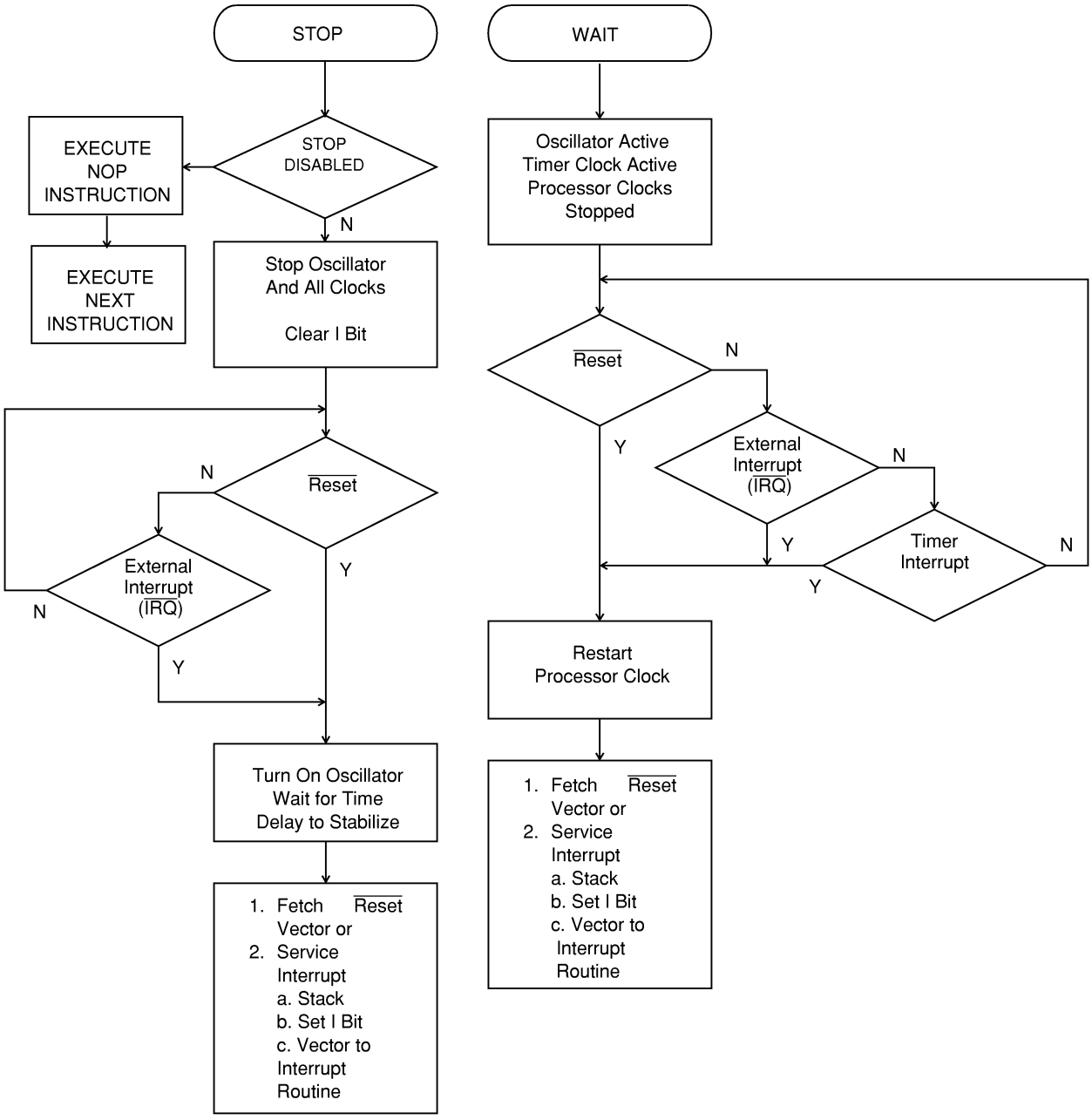


Figure 4-3 STOP/WAIT Flowcharts

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5
TIMER

5.1 INTRODUCTION

The timer consists of a 16-bit, software-programmable counter driven by a fixed divide-by-four prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from several microseconds to many seconds. Refer to Figure 5-1 for a timer block diagram.

Because the timer has a 16-bit architecture, each specific functional segment (capability) is represented by two registers. These registers contain the high and low byte of that functional segment. 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.

Note: The I bit in the CCR should be set while manipulating both the high and low byte register of a specific timer function to ensure that an interrupt does not occur.

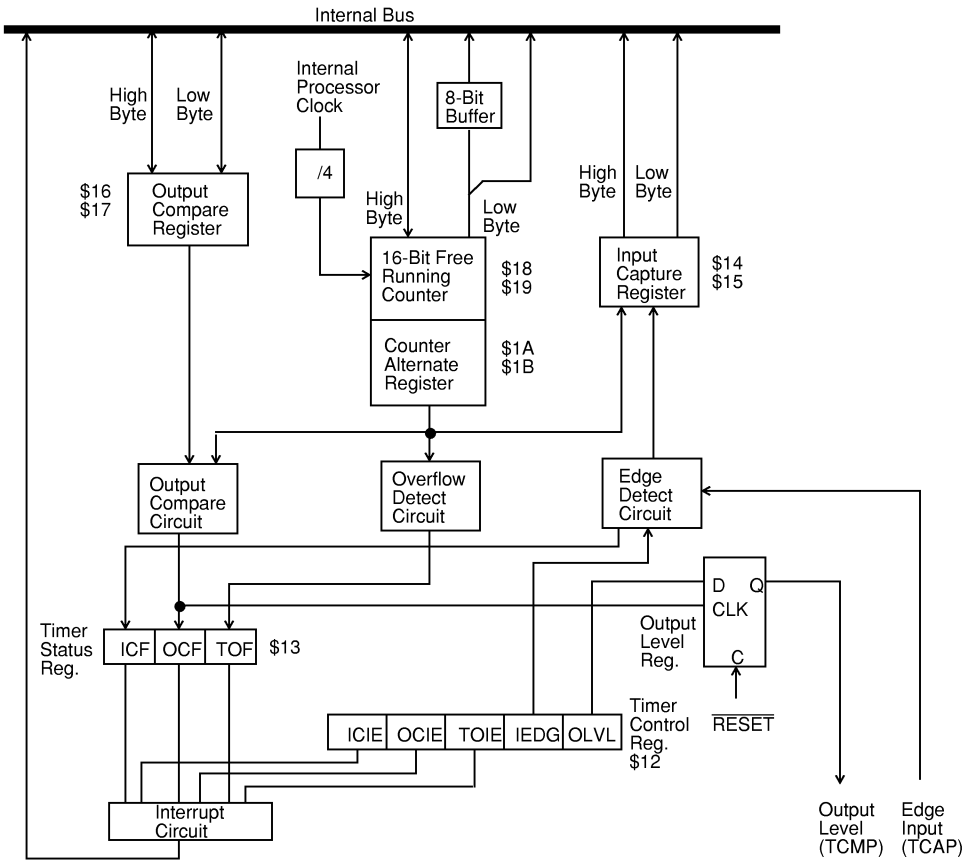


Figure 5-1 Timer Block Diagram

5.2 COUNTER

The key element in the programmable timer is a 16-bit, free-running counter or counter register, preceded by a prescaler that divides the internal processor clock by four. The prescaler gives the timer a resolution of 2.0 microseconds if the internal bus clock is 2.0 MHz. The counter is incremented during the low portion of the internal bus clock. Software can read the counter at any time without affecting its value.

The double-byte, free-running counter can be read from either of two locations, $18-$19 (counter register) or $1A-$1B (counter alternate register). A read from only the least significant byte (LSB) of the free-running counter ($19, $1B) receives the count value at the time of the read. If a read of the free-running counter or counter alternate register first addresses the most significant byte (MSB) ($18, $1A), the LSB ($19, $1B) is transferred to a buffer. This buffer value remains fixed after the first MSB read, even if the user reads the MSB several times. This buffer is accessed when reading the free-running counter or counter alternate register LSB ($19 or $1B) and, thus, completes a read sequence of the total counter value. In reading either the free-running counter or counter alternate register, if the MSB is read, the LSB must also be read to complete the sequence.

The counter alternate register differs from the counter register in one respect: a read of the counter register MSB can clear the timer overflow flag (TOF). Therefore, the counter alternate register can be read at any time without the possibility of missing timer overflow interrupts due to clearing of the TOF.

The free-running counter is configured to $FFFC during reset and is always a read-only register. During a power-on reset, the counter is also preset to $FFFC and begins running after the oscillator start-up delay. Because the free-running counter is 16 bits preceded by a fixed divided-by-four prescaler, the value in the free-running counter repeats every 262,144 internal bus clock cycles. When the counter rolls over from $FFFF to $0000, the TOF bit is set. An interrupt can also be enabled when counter rollover occurs by setting its interrupt enable bit (TOIE).

5.3 OUTPUT COMPARE REGISTER

The 16-bit output compare register is made up of two 8-bit registers at locations $16 (MSB) and $17 (LSB). The output compare register is used for several purposes, such as indicating when a period of time has elapsed. All bits are readable and writable and are not altered by the timer hardware or reset. If the compare function is not needed, the two bytes of the output compare register can be used as storage locations.

The output compare register contents are compared with the contents of the free-running counter continually, and if a match is found, the corresponding output compare flag (OCF) bit is set and the corresponding output level (OLVL) bit is clocked to an output level register. The output compare register values and the output level bit should be changed after each successful comparison to establish a new elapsed time-out. An interrupt can also accompany a successful output compare provided the corresponding interrupt enable bit (OCIE) is set.

After a processor write cycle to the output compare register containing the MSB ($16), the output compare function is inhibited until the LSB ($17) is also written. The user must write

both bytes (locations) if the MSB is written first. A write made only to the LSB ($17) will not inhibit the compare function. The free-running counter is updated every four internal bus clock cycles. The minimum time required to update the output compare register is a function of the program rather than the internal hardware.

The processor can write to either byte of the output compare register without affecting the other byte. The output level (OLVL) bit is clocked to the output level register regardless of whether the output compare flag (OCF) is set or clear.

5.4 INPUT CAPTURE REGISTER

Two 8-bit registers, which make up the 16-bit input capture register, are read-only and are used to latch the value of the free-running counter after the corresponding input capture edge detector senses a defined transition. The level transition which triggers the counter transfer is defined by the corresponding input edge bit (IEDG). $\overline{\text{RESET}}$ does not affect the contents of the input capture register.

The free-running counter contents are transferred to the input capture register on each proper signal transition regardless of whether the input capture flag (ICF) is set or clear. The input capture register always contains the free-running counter value that corresponds to the most recent input capture.

The result obtained by an input capture will be one more than the value of the free-running counter on the rising edge of the internal bus clock preceding the external transition. This delay is required for internal synchronization. Resolution is one count of the free-running counter, which is four internal bus clock cycles.

After a read of the input capture register ($14) MSB, the counter transfer is inhibited until the LSB ($15) is also read. This characteristic causes the time used in the input capture software routine and its interaction with the main program to determine the minimum pulse period.

A read of the input capture register LSB ($15) does not inhibit the free-running counter transfer since they occur on opposite edges of the internal bus clock.

5.5 TIMER CONTROL REGISTER (TCR) $12

The TCR is a read/write register containing five control bits. Three bits control interrupts associated with the timer status register flags ICF, OCF and TOF.

$12	ICIE	OCIE	TOIE	0	0	0	IEDG	OLVL
RESET:	0	0	0	0	0	0	U	0

Figure 5-2 Timer Control Register

ICIE - Input Capture Interrupt Enable

1 = Interrupt enabled

0 = Interrupt disabled

OCIE - Output Compare Interrupt Enable

1 = Interrupt enabled

0 = Interrupt disabled

TOIE - Timer Overflow Interrupt Enable

1 = Interrupt enabled

0 = Interrupt disabled

IEDG - Input Edge

Value of input edge determines which level transition on TCAP pin will trigger free-running counter transfer to the input capture register

1 = Positive edge

0 = Negative edge

Reset does not affect the IEDG bit (U=unaffected).

OLVL - Output Level

Value of output level is clocked into output level register by the next successful output compare and will appear on the TCMP pin

1 = High output

0 = Low output

Bits 2, 3 and 4 - Not used

Always read zero

5.6 TIMER STATUS REGISTER (TSR) $13

The TSR is a read-only register containing three status flag bits.

$13	ICF	OCF	TOF	0	0	0	0	0
RESET:	U	U	U	0	0	0	0	0

Figure 5-3 Timer Status Register

ICF - Input Capture Flag

1 = Flag set when selected polarity edge is sensed by input capture edge detector

0 = Flag cleared when TSR and input capture low register ($15) are accessed

OCF - Output Compare Flag

1 = Flag set when output compare register contents match the free-running counter contents

0 = Flag cleared when TSR and output compare low register ($17) are accessed

TOF - Timer Overflow Flag

1 = Flag set when free-running counter transition from $FFFF to $0000 occurs

0 = Flag cleared when TSR and counter low register ($19) are accessed

Bits 0-4 - Not used

Always read zero

Accessing the timer status register satisfies the first condition required to clear status bits. The remaining step is to access the register corresponding to the status bit.

A problem can occur when using the timer overflow function and reading the free-running counter at random times to measure an elapsed time. Without incorporating the proper precautions into software, the timer overflow flag could unintentionally be cleared if:

1) The timer status register is read or written when TOF is set, and

2) The LSB of the free-running counter is read but not for the purpose of servicing the flag.

The counter alternate register at address $1A and $1B contains the same value as the free-running counter (at address $18 and $19); therefore, this alternate register can be read at any time without affecting the timer overflow flag in the timer status register.

5.7 TIMER DURING WAIT MODE

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

5.8 TIMER DURING STOP MODE

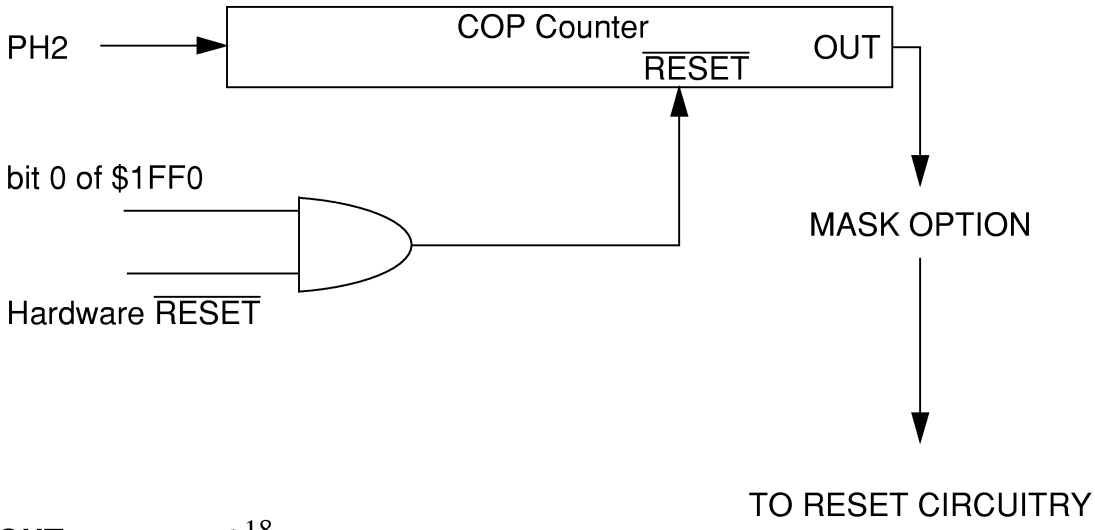
In the STOP mode, the timer stops counting and holds the last count value if STOP is exited by an interrupt. If $\overline{\text{RESET}}$ is used, the counter is forced to $FFFC. During STOP, if at least one valid input capture edge occurs at the TCAP pin, the input capture detect circuit is armed. This does not set any timer flags nor wake up the MCU, but when the MCU does wake up, there is an active input capture flag and data from the first valid edge that occurred during the STOP mode. If $\overline{\text{RESET}}$ is used to exit STOP mode, then no input capture flag or data remains, even if a valid input capture edge occurred.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 6
COP - COMPUTER OPERATING PROPERLY

6.1 INTRODUCTION

This device includes a “Watchdog” COP feature as a mask option. The COP is implemented with a 18-bit ripple counter. This provides a time-out period of 64 milliseconds at a bus rate of 2 MHz. If the COP should time-out, a system reset will occur and the device will be re-initialized in the same fashion as a POR or external $\overline{\text{RESET}}$.



OUT = PH2 ÷ 2^{18}
PH2 = Bus Frequency
MASK OPTION: Enabled or Disabled by Customer with each ROM Code

Figure 6-1 COP Block Diagram

6.2 RESETTING THE COP

Preventing a COP reset is done by writing a “0” to the COPF bit. This action will reset the counter and begin the time-out period again. The COPF bit is bit 0 of address $1FF0. A read of address $1FF0 will result in the user defined ROM data at that location.

6.3 COP FEATURES

For speeding up the COP test, a feature was added in the Bootloader Mode to split the 18-bit counter into 6-bit and 12-bit counters clocked in parallel where the output of the 6-bit counter drives the COP logic. Splitting the counter is accomplished by writing a “1” to

bit 7 of $1FF0. Writing a “0” to bit 7 of $1FF0 will reconnect the two halves of the COP counter.

6.4 COP DURING WAIT MODE

The COP will continue to operate normally during WAIT mode. The software should pull the device out of WAIT mode periodically and reset the COP by writing to the COPF bit to prevent a COP reset.

6.5 COP DURING STOP MODE

STOP mode disables the oscillator circuit and thereby turns the clock off for the entire device. The COP counter will be reset when STOP mode is entered. If a reset is used to exit STOP mode, the COP counter will be reset after the 4064 cycles of delay after STOP mode. If an IRQ is used to exit STOP mode, the COP counter will not be reset after the 4064 cycle delay and will have that many cycles already counted when control is returned to the program.

6.6 COP DURING BOOTLOADER MODE

The COP is disabled in Bootloader Mode.

CHAPTER 7
INPUT/OUTPUT PORTS

In User mode there will be 15 lines arranged as two I/O ports. These pins,PA0-7 and PC0-PC6, are programmable as either inputs or outputs under software control of the data direction registers.

To avoid a glitch on the output pins, write data to the I/O Port Data Register before writing a one to the corresponding Data Direction Register.

7.1 PORT A

Port A is an 8-bit bidirectional port which does not share any of its pins with other subsystems. The port A data register is at $0000 and the data direction register (DDR) is at $0004. Reset does not affect the data registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode.

7.2 PORT B

Port B is a 3-bit port which shares its pins with the on chip comparator. The port B data register is at $0001. Two input only (CMP+/PB6 and CMP-/PB5) pins and one output only pin (CMPO/PB7) comprise Port B. There is no DDR for Port B.

7.3 PORT C

Port C is a 7-bit bidirectional port which does not share any of its pins with other subsystems. The port C data register is at $0002 and the data direction register (DDR) is at $0006. Reset does not affect the data registers, but clears the data direction registers, thereby returning the ports to inputs. Writing a one to a DDR bit sets the corresponding port bit to output mode.

7.4 PORT D

Port D is a 1-bit port which is input only. The port D data register is at $0003. Port D is shared with the timer input capture circuitry. There is no DDR for port D. See **Section 5.4 INPUT CAPTURE REGISTER**.

7.5 INPUT/OUTPUT PROGRAMMING

Ports A and C may be programmed as input or output under software control. The direction of the pins is determined by the state of the corresponding bit in the port data direction register (DDR). Each port has an associated DDR. Any port A or port C pin is configured as an output if its corresponding DDR bit is set to a logic one. A pin is configured as an input if its corresponding DDR bit is cleared to a logic zero.

At power-on or reset, all DDRs are cleared, which configures port A and C pins as inputs. The data direction registers are capable of being written to or read by the processor.

During the programmed output state, a read of the data register actually reads the value of the output data latch and not the I/O pin. See Table 7-1 and Figure 7-1.

Table 7-1 I/O Pin Functions

R/W*	DDR	I/O Pin Function
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 of 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.

R/W is an internal signal.

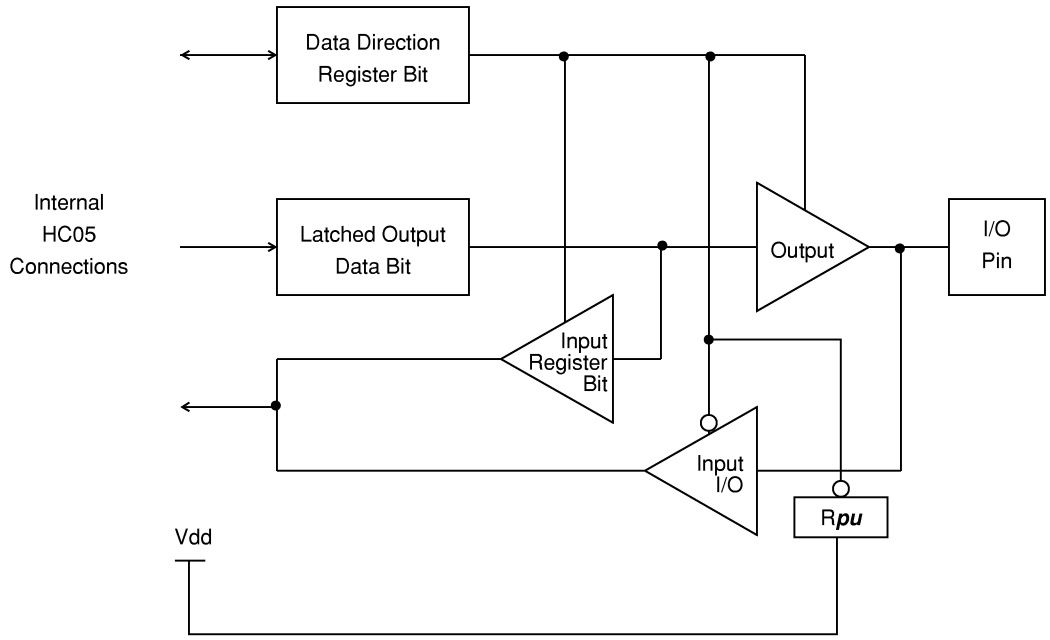


Figure 7-1 Port I/O Circuitry

CHAPTER 8
EPROM

8.1 EPROM INTRODUCTION

The user EPROM consists of 64 bytes situated between addresses $0020- $005F. The erased state of an EPROM bit is logical zero.

With this MCU variant, there is no bootload feature for the direct loading of the EPROM available. The user must supply his own Boot routine.

NOTE: **While the EPROM Programming Voltage (VPP) is applied, under no circumstance must a logic zero be applied to the $\overline{\text{RESET}}$ Pin!**

8.2 EPROM PROGRAMMING REGISTER $1C

$1C	--	--	--	--	--	ELAT	--	EPGM
RESET:	0	0	0	0	0	0	0	0

Figure 8-1 EPROM Programming Register

ELAT - EPROM Address and Data Bus Latch

READ : any time

WRITE : any time

0 = Address and data buses configured for normal operation

1 = Address and data buses configured for EPROM programming. Causes address and data bus to be latched when a write to EPROM is done. EPROM cannot be read if ELAT=1.

EPGM - EPROM Programming Power Control

READ : any time

WRITE : Set only if ELAT = 1, Cleared any time

0 = EPROM programming power is switched off

1 = EPROM programming power is switched on.

If ELAT = 0, then EPGM is automatically cleared. EPGM cannot be set if ELAT is not set. ELAT and EPGM can not both be set on the same write.

Bits 7-3 and 1 - not used; always read logical zero

EPROM Programming Sequence:

1. Apply VPP to the $\overline{\text{IRQ}}$/VPP pin
2. Set the ELAT bit
3. Write to any EPROM address
4. Set the EPGM bit for a time t_{EPGM} to apply the programming voltage
5. Clear the ELAT bit

CHAPTER 9
PWM

9.1 PWM INTRODUCTION

The pulse width modulator (PWM) system has two 8-bit channels (PWM0 and PWM1). PWM 1 is delayed by two (2) PWM clock cycles. The PWM has a programmable period of 256xT, where T can be E/2, E/4, and E/8 for an output frequency of 4 KHz, 2KHz, and 1 KHz respectively with E= 2MHz. E is the internal bus frequency fixed to half of the external oscillator frequency.

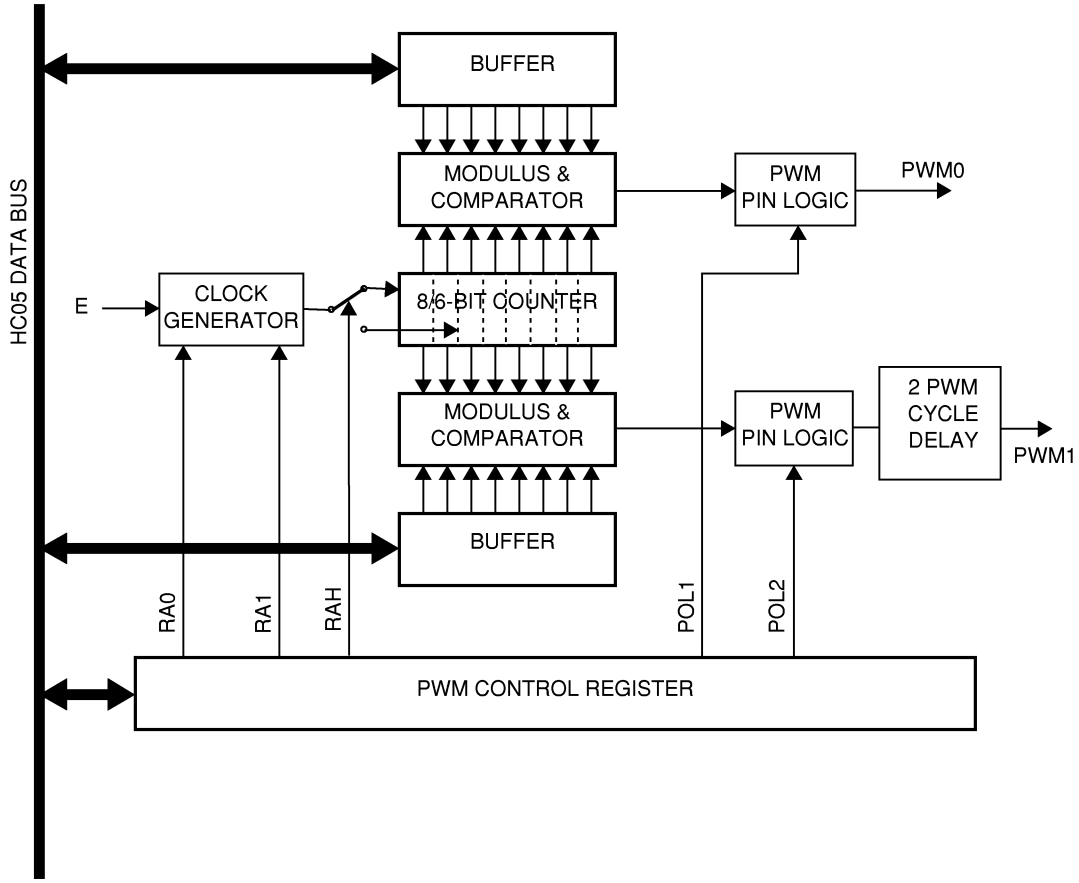


Figure 9-1 PWM Block Diagram

9.2 FUNCTIONAL DESCRIPTION

The PWM is capable of generating signals from 0% to 100% duty cycle. A $00 in the PWM Data Register yields an “off” output (0%), but an $FF yields a duty of 255/256. To achieve the 100% duty (“ON” output), the polarity control bit is set to one for that channel (i.e. PWM0 and PWM1) while the data register has $00 in it.

When not in use, the PWM system can be shut off to save power by clearing the clock rate select bits RA0 and RA1 in PWCR.

Writes to the PWM can be performed at any time without affecting the output signal until the two data latches and the control latch have been written to. Updates on the outputs, PWM0 and PWM1, occur at the end of the PWM period. At this time the new value is loaded into the PWM Data and Control registers. If a write to the registers is performed during the off state, the data gets transferred directly to the PWM registers. All Registers are updated after the PWM Data Register 1 is written to and the end of a PWM cycle occurs.

The PWM output can have an active high or an active low pulse under software control.

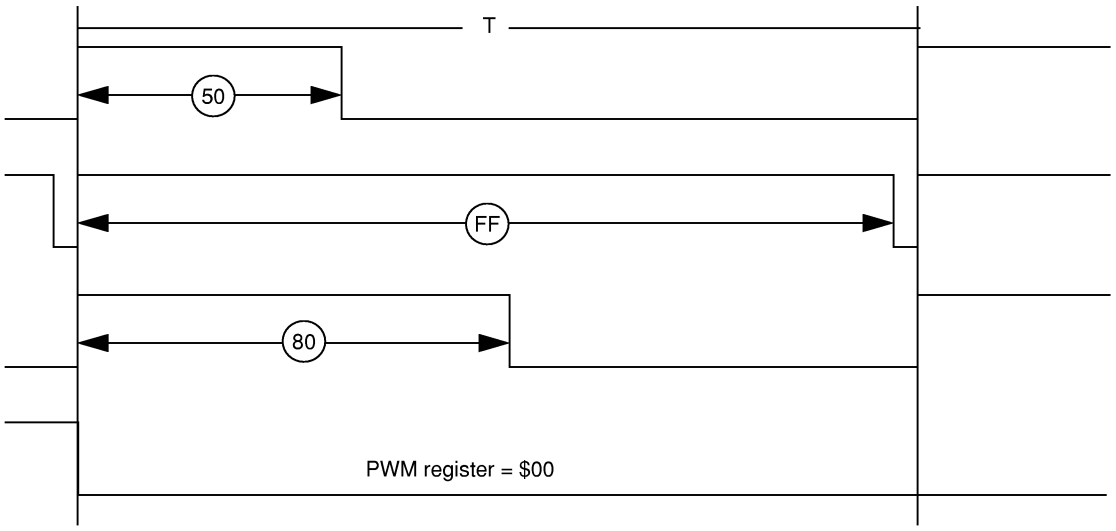


Figure 9-2 PWM Waveforms (POL = 1)

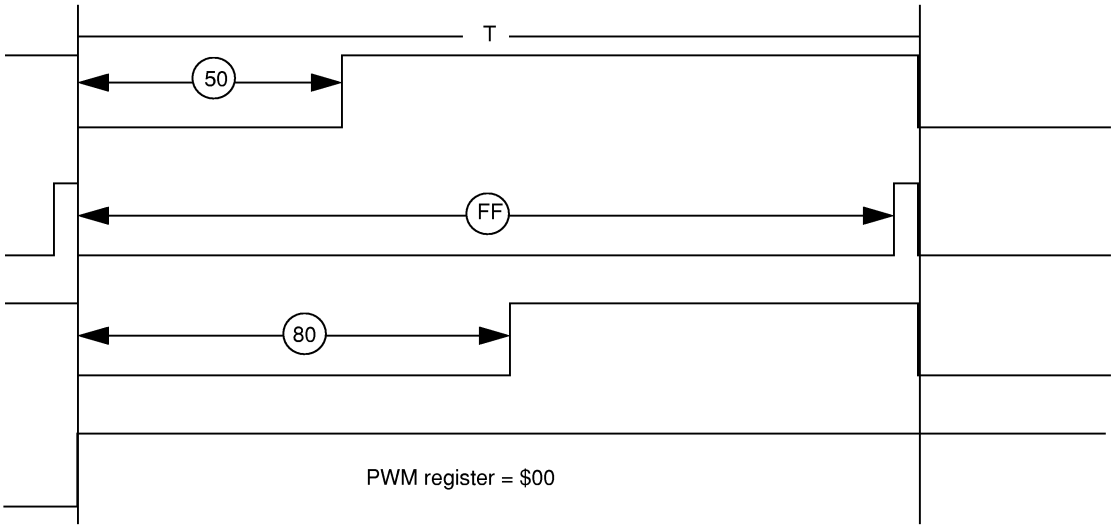


Figure 9-3 PWM Waveforms (POL = 0)

9.3 REGISTERS

Associated with the PWM system, there are two PWM data registers and a control register. The following registers can be written to and read at any time. The data is not latched until PWM Data Register 1 (address location = $10) has been written to and the end of a cycle has been reached. Upon RESET the user should write to the data registers prior to enabling the PWM system (i.e. prior to setting RA1 and or RA0 for PWM input clock rate). This will avoid an erroneous duty cycle from being driven. During regular user mode the user should write to PWM Data 1 data register after writing the PWM CONTROL register and the PWM Data 0 data register.

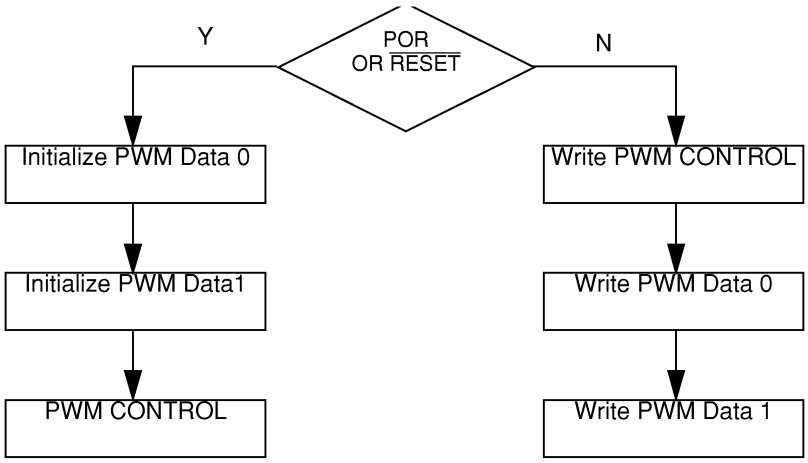


Figure 9-4 PWM Write Sequences

9.3.1 PWM CONTROL $0E.

$0E	--	--	--	RAH	POL1	POL0	RA1	RA0
RESET:	--	--	--	0	1	1	0	0

Figure 9-5 PWM Control Register

POL1 - PWM1 Polarity

These bits when set, makes the active PWM1 pulse high, when low, the pulse is active low. (e.g. $00 yields an all high signal.)

POL0 - PWM0 Polarity

These bits when set, makes the active PWM0 pulse high, when low, the pulse is active low. (e.g. $00 yields an all high signal.)

RAH - PWM0 Polarity

This bit selects the counter width (RAH=0 8-bit, RAH=1 6-bit) and determines the period. When the 6-bit counter is switched on, the two lsb-stages of the

comparator are disabled. That means, bit 0 and bit 1 are 'don't care'. The resolution is a lower one. For example, a $00, or $01, or $02, or $03 all yield the same duty of 0/256, a $FC, or $FD, or $FE, or $FF all yield the greatest duty of 255/256.

RA1, RA2 - PWM A Clock Rate bits

These bits select the input clock rate and determines the period.

Table 9-1 PWM Clock Rate and Output Frequency

RA1	RA0	RAH	PWM INPUT CLOCK	OUTPUT FREQUENCY
0	0	0	OFF	-
0	1	0	E/2	4 kHz
1	0	0	E/4	2 kHz
1	1	0	E/8	1 kHz
0	0	1	OFF	-
0	1	1	E/2	16 kHz
1	0	1	E/4	8 kHz
1	1	1	E/8	4 kHz

9.3.2 PWM DATA REGISTERS

The PWM system has two 8-bit data registers which hold the duty cycle for each PWM output. PWM Data 0 and PWM Data1 are the data registers located at $0F and $10 respectively. These registers are unaffected by RESET.

Figure 9-6 PWM Data Registers

A delay of 2 PWM clock counts exists between the rising edge of the PWM0 signal to the rising edge of the PWM1 signal. Refer to Table 9-1.

9.4 PWM DURING WAIT MODE

The PWM continues normal operation during WAIT mode. To decrease power consumption during WAIT, it is recommended that the rate select bits in both the PWM Control Registers be cleared if the PWM D/A converter is not being used.

9.5 PWM DURING STOP MODE

In STOP mode the oscillator is stopped causing the PWM to cease function. Any signal in process is aborted in whatever phase the signal happens to be in.

9.6 PWM DURING RESET

Upon RESET the RA0 and RA1 bits in PWM CONTROL are cleared. This in effect disables the PWM system and sets the outputs driving low. The user should write to the data registers prior to enabling the PWM system (i.e. prior to setting RA1 or RA0). This will avoid an erroneous duty cycle from being driven.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 10
COMPARATOR

Port B shares its pins with CMP+/PB6, CMP-/PB5 and CMPO/PB7 of the comparator. This circuitry is used for the comparison of analog signals, with a digital output. The comparator circuitry may be powered up by setting the CEN0 bit in the Comparator Control Register. The register is located at $000C and is cleared by reset. The state of the comparator output pin upon $\overline{\text{RESET}}$ is logic 0. For further electrical information see **CHAPTER 11 ELECTRICAL SPECIFICATIONS**.

10.1 COMPARATOR CONTROL REGISTER

$0C	--	--	--	--	--	--	--	CEN0
RESET:								0

Figure 10-1 Comparator Control Register

CEN0 - Comparator Enable

This bit powers up the voltage comparator on PB7-PB5 when set. The user must set this bit in order to allow any functionality of the comparator. After enabling the comparator, the user should delay for t_{CEN} before reading the state of the output. RESET clears this bit.

10.2 READING COMPARATOR OUTPUT

The state of the output pin, CMPO/PB7, is internally readable. From Figure 3-2 Port B Data is located at address location $01. A read of Port B bit 7 reads the data latch. When CEN0 is set this read will give the current state of the comparator output. Port B bit 5 and 6 are not internally readable. Writes to PB7/COMPO when the CEN0 bit is set have no effect to PB7. Writes to PB7/COMPO when the CEN0 bit is cleared writes to PB7 data latch.

10.3 COMPARATOR DURING STOP

STOP clears the CEN0 bit. This in effect disables the comparator. If the STOP disable Mask option is used all STOP instructions become a NOP and STOP mode will never occur. For further information on STOP mode see **Section 4.6 LOW-POWER MODES**.

10.4 COMPARATOR DURING WAIT

The comparator operates normally in WAIT. If the user wishes to save power during the WAIT mode, the CEN0 bit should be cleared before the WAIT instruction. For further information on WAIT mode see **Section 4.6 LOW-POWER MODES**.

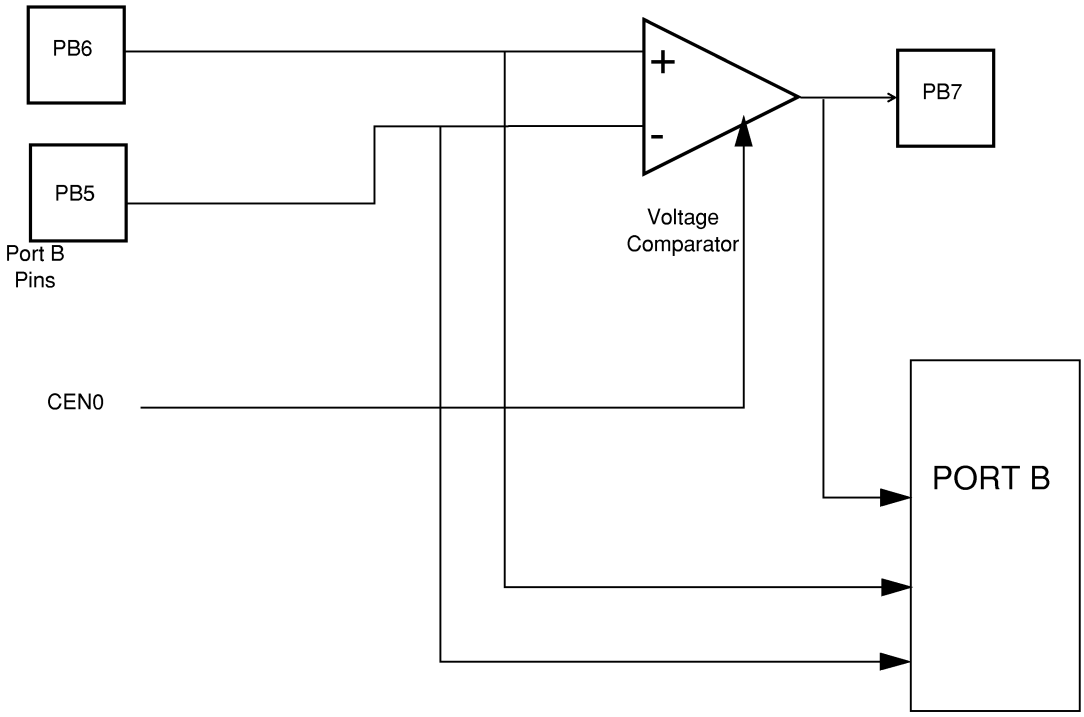


Figure 10-2 Port B Comparator Circuitry

CHAPTER 11
ELECTRICAL SPECIFICATIONS

11.1 MAXIMUM RATINGS

(Voltages referenced to VSS)

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	-0.5 to +7.0	V
Input Voltage	V_{in}	V_{SS} - 0.5 to V_{DD} + 0.5	V
Bootloader Mode ($\overline{IRQ}$ Pin Only)	V_{in}	V_{SS} - 0.5 to $2 \times V_{DD}$ + 0.5	V
EPROM Programming ($\overline{IRQ}$ Pin Only)	VPP	V_{SS} - 0.5 to 17 + 0.5	V
Current Drain Per Pin Excluding V_{DD} and V_{SS}	I	25	mA
Operating Temperature Range (Standard)	T_A	T_L to T_H -40 to +105	C
Storage Temperature Range	T_{stg}	-65 to +150	C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{DD}$. Reliability of operation is enhanced if unused inputs are connected to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

11.2 THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic	θ_{JA}	60	C/W
SOIC	θ_{JA}	TBD	C/W

11.3 DC ELECTRICAL CHARACTERISTICS (5.0 V_{dc})

(V_{DD} = 5.0 V_{dc} ±10%, V_{SS} = 0 V_{dc}, T_A = -40°C to +105°C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Output voltage I_{Load} = 10.0 A I_{Load} = -10.0 A	 V_{OL} V_{OH}	 - V_{DD}-0.1	 - -	 0.1 -	 V V
Output High Voltage (I_{Load} = -0.8 mA) PA0-7, PB5-7,PC0-6, PWM0, PWM1	V_{OH}	V_{DD}-0.8	-	-	V
Output Low Voltage (I_{Load} = 1.6 mA) PA0-7, PB5-7,PC0-6, PWM0, PWM1	V_{OL}	-	-	0.4	V
Input High Voltage PA0-7, PB0-7, PC0-6, PB6, PB5, TCAP/PB7, $\overline{IRQ}$, $\overline{RESET}$,OSC1	V_{IH}	0.7×V_{DD}	-	V_{DD}	V
Input Low Voltage PA0-7, PC0-6, PB6, PB5,OSC1	V_{IL}	V_{SS}	-	0.2×V_{DD}	V
Input Low Voltage TCAP/PD7	V_{IL}	-	-	0.2×V_{DD}	
Input Low Voltage $\overline{RESET}$	V_{IL}	-	-	0.2×V_{DD}	
EPROM Programming Voltage @33.0mA Ipp per byte	VPP	15.5	16.0	16.5	V
Supply Current (see Notes) Run (f_{osc}= 4MHz, f_{op}=2MHz) Wait (f_{osc}=4MHz, f_{op}=2MHz) Stop 25 C (Comparator Disabled) 0 C to +70(STANDARD)	 I_{DD} I_{DD} I_{DD} I_{DD}	 - - - -	 TBD TBD TBD TBD	 5 1.2 TBD TBD	 mA mA µA µA
I/O Ports pull-up Resistor PA0-7, PC0-6	R_{PU}	10	-	50	kΩ
Input Current with Port Pull-ups PA0-7, PC0-6	I_{in}	-	-		µA
Input Current $\overline{RESET}$, $\overline{IRQ}$, OSC1	I_{in}	-	-	1	µA
Input Current PB5, PB6, TCAP/PD7	I_{in}				µA
Capacitance Ports (as Input or Output) $\overline{RESET}$, $\overline{IRQ}$	 C_{ou} C_{in}	 - -	 - -	 12 8	 pF pF

NOTES:

1. All values shown reflect average measurements.
2. Typical values at midpoint of voltage range, 25°C only.
3. Wait I_{DD}: Only timer system active.
4. Run (Operating) I_{DD}, Wait I_{DD}: Measured using external square wave clock source (f_{osc} = 4 MHz), all inputs 0.2 V from rail; no dc loads, less than 50pF on all outputs, C_L = 20 pF on OSC2.
5. Wait, Stop I_{DD}: All ports configured as inputs, V_{IL} = 0.2 V, V_{IH} = V_{DD}-0.2 V.
6. Stop I_{DD} measured with OSC1 = V_{SS}.
7. Wait I_{DD} is affected linearly by the OSC2 capacitance.
8. Values for R_{PU} for Min, Typ, and Max taken at -40 C, 25 C, and 105 C respectively

11.4 CONTROL TIMING (5.0 V_{dc})

(V_{DD} = 5.0 V_{dc} ±10%, V_{SS} = 0 V_{dc}, T_A = 0 C to +70 C, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Frequency of Operation Crystal Option External Clock Option	 f_{osc} f_{osc}	 - dc	 4.2 4.2	 MHz MHz
Internal Operating Frequency Crystal External Clock	 f_{op} f_{op}	 - dc	 2.1 2.1	 MHz MHz
Cycle Time	t_{cyc}	480	-	ns
Crystal Oscillator Start-up Time	t_{OXOV}	-	100	ms
Stop Recovery Start-up Time	t_{ILCH}	-	100	ms
RESET* Pulse Width	t_{RL}	1.5	-	t_{cyc}
Interrupt Pulse Width Low (Edge-Triggered)	t_{ILIH}	125	-	ns
Interrupt Pulse Period	t_{ILIL}	*	-	t_{cyc}
OSC1 Pulse Width	t_{OH},t_{OL}	90	-	ns
EPROM Programming Time	t EPGM	4	-	ms

* The minimum period T_{ILIL} should not be less than the number of cycle times it takes to execute the interrupt service routine plus 21 t_{cyc}.

11.5 COMPARATOR

(V_{DD} = 5.0 V_{dc} ±10%, V_{SS} = 0 V_{dc}, T_A = -40 C to +105 C, unless otherwise noted)

Characteristic	Symbol	Min	Max	Unit
Input Voltage Range[1]	V_{INT}	Vss - 0.5	Vdd + 0.5	V
Voltage Comparator Propagation Time (100mV overdrive)[2]	t_{COMP}	—	10	s
Common Mode Range[3]	V_{COM}	Vss-0.5	Vss +2.5	V
Offset[4]	V_{OFF}	—	±15	mV
Comparator Enable Time[5]	t_{CEN}	--	100	ms
Input Current Vss ≤ Vin ≤ Vdd Input Current Vin ≤ Vss	IIn IIn	— —	1 TBD	A A
Gain Bandwidth Product		—	10	KHz
Slew Rate	SR	5	—	V/s
Saturation Voltage (IOUT = -5ma)	V_{SAT}	—	0.5	V

NOTE:

1. Comparator is guaranteed to function over the specified input voltage range with no erroneous outputs.
2. Signal propagation time through the comparator measured with 100mv of overdrive.
3. Comparator is guaranteed to meet specifications i.e. SR, t_{comp}, and $V_{off.}$
4. Input offset voltage is guaranteed over the temperature range.
5. Enable time is the time from enabling the comparator with the CEN0 bit until the comparator is fully functional.

11.6 PWM TIMING

Characteristic	Symbol	Min	Max	Unit
PWM Rise time	t_{PWMr}	15	35	ns
PWM Fall time	t_{PWMr}	15	35	ns

NOTE: See Figure 9-2 (POL = 1).

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.