PIC16CE62X

OTP 8-Bit CMOS MCU with EEPROM Data Memory

Devices included in this data sheet:

- PIC16CE623
- PIC16CE624
- PIC16CE625

High Performance RISC CPU:

- Only 35 instructions to learn
- All single-cycle instructions (200 ns), except for program branches which are two-cycle
- Operating speed:
  - DC - 20 MHz clock input
  - DC - 200 ns instruction cycle

Device	Program Memory	RAM Data Memory	EEPROM Data Memory
PIC16CE623	512x14	96x8	128x8
PIC16CE624	1Kx14	96x8	128x8
PIC16CE625	2Kx14	128x8	128x8

- Interrupt capability
- 16 special function hardware registers
- 8-level deep hardware stack
- Direct, Indirect and Relative addressing modes

Peripheral Features:

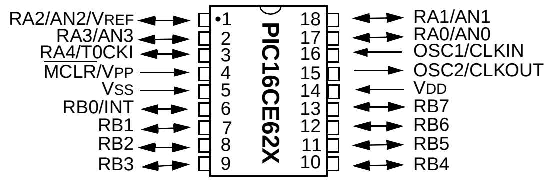
- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
- Analog comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (V_{REF}) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs can be output signals
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler

Special Microcontroller Features:

- In-Circuit Serial Programming (ICSP™) (via two pins)
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Reset
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation

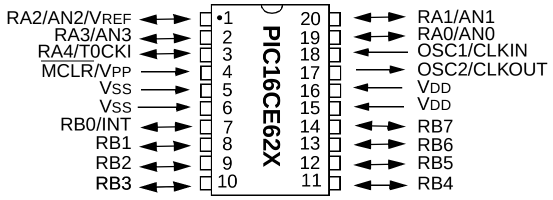
Pin Diagrams

PDIP, SOIC, Windowed CERDIP



RA2/AN2/VREF 1
RA3/AN3 2
RA4/T0CKI 3
MCLR/VPP 4
VSS 5
RB0/INT 6
RB1 7
RB2 8
RB3 9
PIC16CE62X
18 RA1/AN1
17 RA0/AN0
16 OSC1/CLKIN
15 OSC2/CLKOUT
14 VDD
13 RB7
12 RB6
11 RB5
10 RB4

SSOP



RA2/AN2/VREF 1
RA3/AN3 2
RA4/T0CKI 3
MCLR/VPP 4
VSS 5
VSS 6
RB0/INT 7
RB1 8
RB2 9
RB3 10
PIC16CE62X
20 RA1/AN1
19 RA0/AN0
18 OSC1/CLKIN
17 OSC2/CLKOUT
16 VDD
15 VDD
14 RB7
13 RB6
12 RB5
11 RB4

Special Microcontroller Features (cont'd)

- 1,000,000 erase/write cycle EEPROM data memory
- EEPROM data retention > 40 years
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Four user programmable ID locations

CMOS Technology:

- Low-power, high-speed CMOS EPROM/EEPROM technology
- Fully static design
- Wide operating voltage range
  - 2.5V to 5.5V
- Commercial, industrial and extended temperature range
- Low power consumption
  - < 2.0 mA @ 5.0V, 4.0 MHz
  - 15 µA typical @ 3.0V, 32 kHz
  - < 1.0 µA typical standby current @ 3.0V

Table of Contents

1.0 General Description 3
2.0 PIC16CE62X Device Varieties 5
3.0 Architectural Overview 7
4.0 Memory Organization 11
5.0 I/O Ports 23
6.0 EEPROM Peripheral Operation 29
7.0 Timer0 Module 35
8.0 Comparator Module 41
9.0 Voltage Reference Module 47
10.0 Special Features of the CPU 49
11.0 Instruction Set Summary 65
12.0 Development Support 77
13.0 Electrical Specifications 83
14.0 Packaging Information 97
Appendix A: Code for Accessing EEPROM Data Memory 103
Index 105
On Line Support 107
Reader Response 108
PIC16CE62X Product Identification System 109

To Our Valued Customers

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number. e.g., DS30000A is version A of document DS30000.

New Customer Notification System

Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

Errata

An errata sheet may exist for current devices, describing minor operational differences (from the data sheet) and recommended workarounds. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)
- The Microchip Corporate Literature Center; U.S. FAX: (480) 786-7277

When contacting a sales office or the literature center, please specify which device, revision of silicon and data sheet (include literature number) you are using.

Corrections to this Data Sheet

We constantly strive to improve the quality of all our products and documentation. We have spent a great deal of time to ensure that this document is correct. However, we realize that we may have missed a few things. If you find any information that is missing or appears in error, please:

- Fill out and mail in the reader response form in the back of this data sheet.
- E-mail us at webmaster@microchip.com.

We appreciate your assistance in making this a better document.

1.0 GENERAL DESCRIPTION

The PIC16CE62X are 18 and 20-Pin EPROM-based members of the versatile PICmicro® family of low-cost, high-performance, CMOS, fully-static, 8-bit microcontrollers with EEPROM data memory.

All PICmicro® microcontrollers employ an advanced RISC architecture. The PIC16CE62X family has enhanced core features, eight-level deep stack, and multiple internal and external interrupt sources. The separate instruction and data buses of the Harvard architecture allow a 14-bit wide instruction word with separate 8-bit wide data. The two-stage instruction pipeline allows all instructions to execute in a single-cycle, except for program branches (which require two cycles). A total of 35 instructions (reduced instruction set) are available. Additionally, a large register set gives some of the architectural innovations used to achieve a very high performance.

PIC16CE62X microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers in their class.

The PIC16CE623 and PIC16CE624 have 96 bytes of RAM. The PIC16CE625 has 128 bytes of RAM. Each microcontroller contains a 128x8 EEPROM memory array for storing non-volatile information, such as calibration data or security codes. This memory has an endurance of 1,000,000 erase/write cycles and a retention of 40 plus years.

Each device has 13 I/O pins and an 8-bit timer/counter with an 8-bit programmable prescaler. In addition, the PIC16CE62X adds two analog comparators with a programmable on-chip voltage reference module. The comparator module is ideally suited for applications requiring a low-cost analog interface (e.g., battery chargers, threshold detectors, white goods controllers, etc).

PIC16CE62X devices have special features to reduce external components, thus reducing system cost, enhancing system reliability and reducing power consumption. There are four oscillator options, of which the single pin RC oscillator provides a low-cost solution, the LP oscillator minimizes power consumption, XT is a standard crystal, and the HS is for High Speed crystals. The SLEEP (power-down) mode offers power savings. The user can wake-up the chip from SLEEP through several external and internal interrupts and reset.

A highly reliable Watchdog Timer with its own on-chip RC oscillator provides protection against software lock- up.

A UV-erasable CERDIP-packaged version is ideal for code development, while the cost-effective One-Time Programmable (OTP) version is suitable for production in any volume.

Table 1-1 shows the features of the PIC16CE62X mid-range microcontroller families.

A simplified block diagram of the PIC16CE62X is shown in Figure 3-1.

The PIC16CE62X series fits perfectly in applications ranging from multi-pocket battery chargers to low-power remote sensors. The EPROM technology makes customization of application programs (detection levels, pulse generation, timers, etc.) extremely fast and convenient. The small footprint packages make this microcontroller series perfect for all applications with space limitations. Low-cost, low-power, high-performance, ease of use and I/O flexibility make the PIC16CE62X very versatile.

1.1 Development Support

The PIC16CE62X family is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a low-cost development programmer and a full-featured programmer. A “C” compiler is also available.

TABLE 1-1: PIC16CE62X FAMILY OF DEVICES

		PIC16CE623	PIC16CE624	PIC16CE625
Clock	Maximum Frequency of Operation (MHz)	20	20	20
Memory	EPROM Program Memory (x14 words)	512	1K	2K
	Data Memory (bytes)	96	96	128
Peripherals	EEPROM Data Memory (bytes)	128	128	128
	Timer Module(s)	TMR0	TMR0	TMR0
	Comparators(s)	2	2	2
	Internal Reference Voltage	Yes	Yes	Yes
Features	Interrupt Sources	4	4	4
	I/O Pins	13	13	13
	Voltage Range (Volts)	2.5-5.5	2.5-5.5	2.5-5.5
	Brown-out Reset	Yes	Yes	Yes
	Packages	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP	18-pin DIP, SOIC; 20-pin SSOP

All PICmicro® Family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect and high I/O current capability. All PIC16CE62X Family devices use serial programming with clock pin RB6 and data pin RB7.

2.0 PIC16CE62X DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements the proper device option can be selected using the information in the PIC16CE62X Product Identification System section at the end of this data sheet. When placing orders, please use this page of the data sheet to specify the correct part number.

2.1 UV Erasable Devices

The UV erasable version, offered in the CERDIP package is optimal for prototype development and pilot programs. This version can be erased and reprogrammed to any of the oscillator modes.

Microchip's PICSTART® and PRO MATE® programmers both support programming of the PIC16CE62X.

2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers who need the flexibility for frequent code updates and small volume applications. In addition to the program memory, the configuration bits must also be programmed.

2.3 Quick-Turn-Programming (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who chose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices but with all EPROM locations and configuration options already programmed by the factory. Certain code and prototype verification procedures apply before production shipments are available. Please contact your Microchip Technology sales office for more details.

2.4 Serialized Quick-Turn-Programming (SQTP℠) Devices

Microchip offers a unique programming service where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number which can serve as an entry-code, password or ID number.

NOTES:

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16CE62X family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16CE62X uses a Harvard architecture in which program and data are accessed from separate memories using separate buses. This improves bandwidth over traditional von Neumann architecture where program and data are fetched from the same memory. Separating program and data memory further allows instructions to be sized differently than 8-bit wide data word. Instruction opcodes are 14-bits wide making it possible to have all single word instructions. A 14-bit wide program memory access bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions. Consequently, all instructions (35) execute in a single-cycle (200 ns @ 20 MHz) except for program branches.

The table below lists program memory (EPROM), data memory (RAM) and non-volatile memory (EEPROM) for each PIC16CE62X device.

Device	**Program Memory**	**RAM Data Memory**	**EEPROM Data Memory**
PIC16CE623	512x14	96x8	128x8
PIC16CE624	1Kx14	96x8	128x8
PIC16CE625	2Kx14	128x8	128x8

The PIC16CE62X can directly or indirectly address its register files or data memory. All special function registers including the program counter are mapped in the data memory. The PIC16CE62X family has an orthogonal (symmetrical) instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of ‘special optimal situations’ make programming with the PIC16CE62X simple yet efficient. In addition, the learning curve is reduced significantly.

The PIC16CE62X devices contain an 8-bit ALU and working register. The ALU is a general purpose arithmetic unit. It performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is 8 bits wide and capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. In two-operand instructions, typically one operand is the working register (W register). The other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

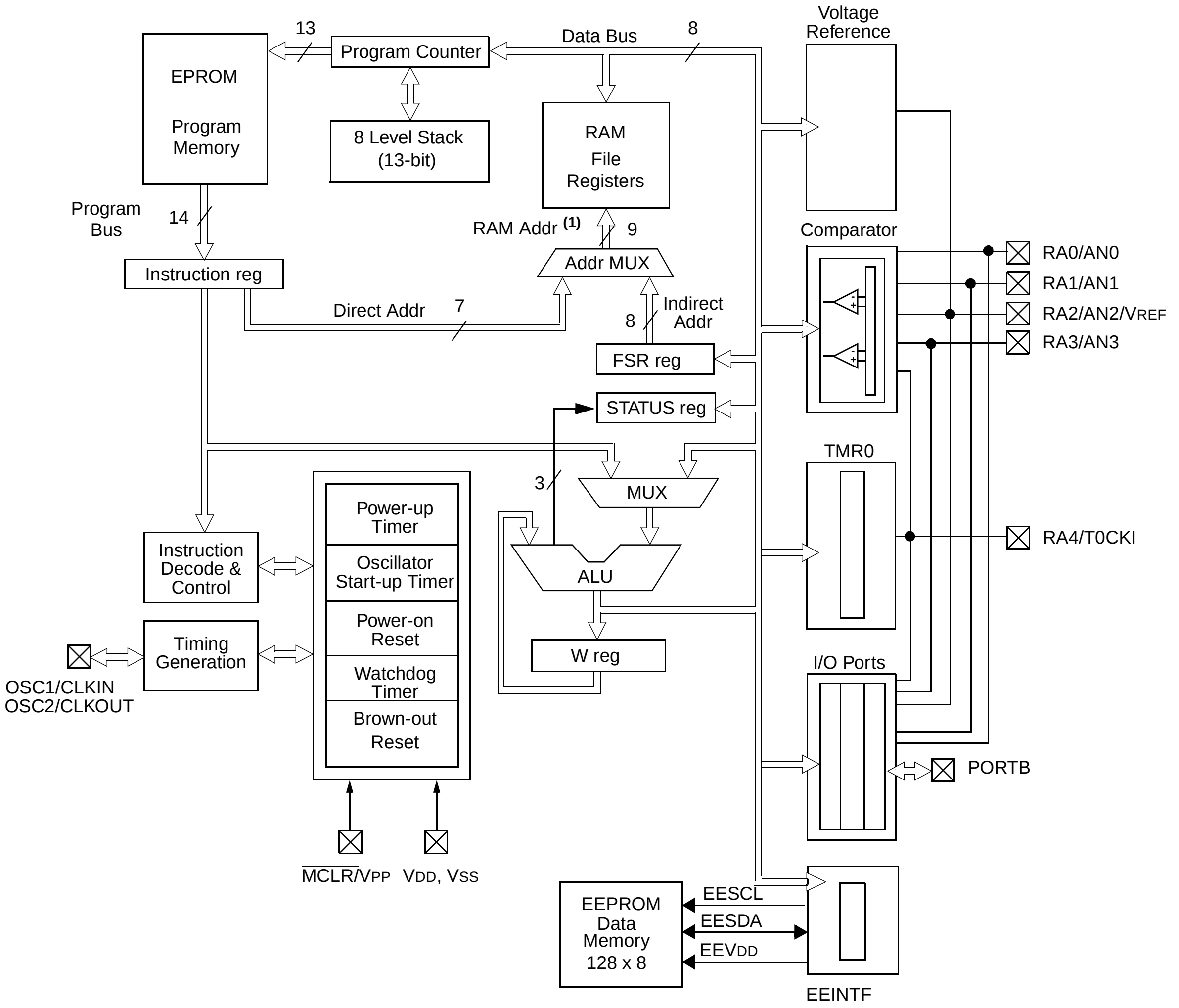
The W register is an 8-bit working register used for ALU operations. It is not an addressable register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{Borrow}}$ and $\overline{\text{Digit}}$ $\overline{\text{Borrow}}$ out bit respectively, bit in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

A simplified block diagram is shown in Figure 3-1, with a description of the device pins in Table 3-1.

FIGURE 3-1: BLOCK DIAGRAM

Device	Program Memory	Data Memory (RAM)	EEPROM DATA MEMORY
PIC16CE623	512 x 14	96 x 8	128 x 8
PIC16CE624	1K x 14	96 x 8	128 x 8
PIC16CE625	2K x 14	128 x 8	128 x 8



13
Program Counter
Data Bus
8
Voltage Reference
EPROM
Program Memory
8 Level Stack (13-bit)
RAM File Registers
Program Bus
14
RAM Addr (1)
9
Comparator
Instruction reg
Addr MUX
RA0/AN0
RA1/AN1
RA2/AN2/VREF
RA3/AN3
Direct Addr
7
Indirect Addr
8
FSR reg
STATUS reg
TMR0
3
MUX
RA4/T0CKI
Power-up Timer
Oscillator Start-up Timer
Power-on Reset
Watchdog Timer
Brown-out Reset
Instruction Decode & Control
Timing Generation
ALU
W reg
OSC1/CLKIN
OSC2/CLKOUT
I/O Ports
PORTB
MCLR/VPP
VDD, VSS
EEPROM Data Memory 128 x 8
EESCL
EESDA
EEVDD
EEINTF

Note 1: Higher order bits are from the STATUS register.

TABLE 3-1: PIC16CE62X PINOUT DESCRIPTION

Name	DIP/ SOIC Pin #	SSOP Pin #	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	18	I	ST/CMOS	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	17	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{\text{MCLR}}$/VPP	4	4	I/P	ST	Master clear (reset) input/programming voltage input. This pin is an active low reset to the device.
					PORTA is a bi-directional I/O port.
RA0/AN0	17	19	I/O	ST	Analog comparator input
RA1/AN1	18	20	I/O	ST	Analog comparator input
RA2/AN2/VREF	1	1	I/O	ST	Analog comparator input or VREF output
RA3/AN3	2	2	I/O	ST	Analog comparator input /output
RA4/T0CKI	3	3	I/O	ST	Can be selected to be the clock input to the Timer0 timer/counter or a comparator output. Output is open drain type.
					PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	6	7	I/O	TTL/ST(1)	RB0/INT can also be selected as an external interrupt pin.
RB1	7	8	I/O	TTL	
RB2	8	9	I/O	TTL	
RB3	9	10	I/O	TTL	
RB4	10	11	I/O	TTL	Interrupt on change pin.
RB5	11	12	I/O	TTL	Interrupt on change pin.
RB6	12	13	I/O	TTL/ST(2)	Interrupt on change pin. Serial programming clock.
RB7	13	14	I/O	TTL/ST(2)	Interrupt on change pin. Serial programming data.
VSS	5	5,6	P	—	Ground reference for logic and I/O pins.
VDD	14	15,16	P	—	Positive supply for logic and I/O pins.

Legend:
O = output
I/O = input/output
P = power
— = Not used
I = Input
ST = Schmitt Trigger input
TTL = TTL input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

Note 2: This buffer is a Schmitt Trigger input when used in serial programming mode.

3.1 Clocking Scheme/Instruction Cycle

The clock input (OSC1/CLKIN pin) is internally divided by four to generate four non-overlapping quadrature clocks namely Q1, Q2, Q3 and Q4. Internally, the program counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow is shown in Figure 3-2.

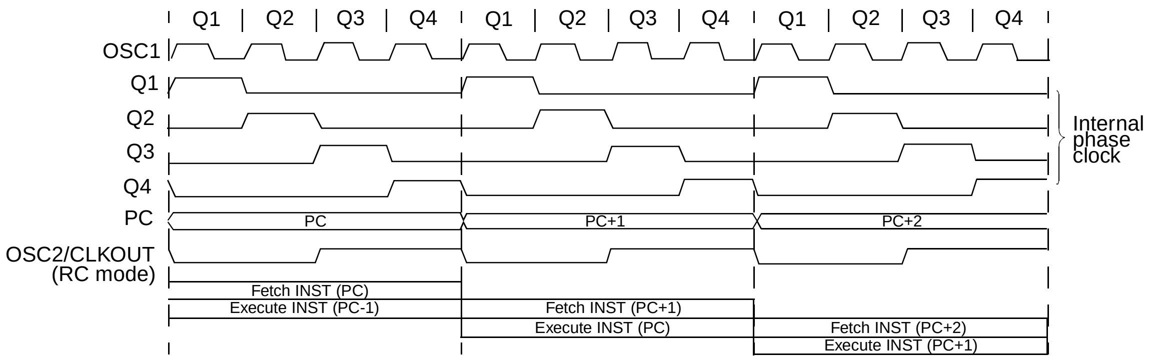
3.2 Instruction Flow/Pipelining

An “Instruction Cycle” consists of four Q cycles (Q1, Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle, while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (i.e., `GOTO`) then two cycles are required to complete the instruction (Example 3-1).

A fetch cycle begins with the program counter (PC) incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the “Instruction Register (IR)” in cycle Q1. This instruction is then decoded and executed during the Q2, Q3, and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

FIGURE 3-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW

`1. MOVLW 55h`	Fetch 1	Execute 1				
`2. MOVWF PORTB`		Fetch 2	Execute 2			
`3. CALL  SUB_1`			Fetch 3	Execute 3		
`4. BSF   PORTA, BIT3`				Fetch 4	Flush	
`5. Instruction @ address SUB_1`					Fetch SUB_1	Execute SUB_1

All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is “flushed” from the pipeline, while the new instruction is being fetched and then executed.

4.0 MEMORY ORGANIZATION

4.1 Program Memory Organization

The PIC16CE62X has a 13-bit program counter capable of addressing an 8K x 14 program memory space. Only the first 512 x 14 (0000h - 01FFh) for the PIC16CE623, 1K x 14 (0000h - 03FFh) for the PIC16CE624 and 2K x 14 (0000h - 07FFh) for the PIC16CE625 are physically implemented. Accessing a location above these boundaries will cause a wrap-around within the first 512 x 14 space (PIC16CE623) or 1K x 14 space (PIC16CE624) or 2K x 14 space (PIC16CE625). The reset vector is at 0000h and the interrupt vector is at 0004h (Figure 4-1, Figure 4-2, Figure 4-3).

FIGURE 4-1: PROGRAM MEMORY MAP AND STACK FOR THE PIC16CE623

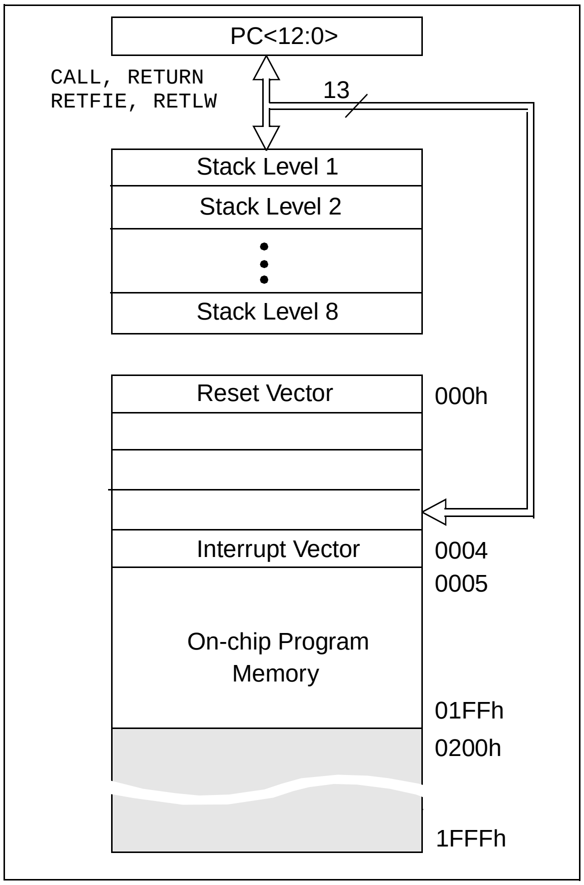

PC<12:0>
CALL, RETURN
RETFIE, RETLW
13
Stack Level 1
Stack Level 2
Stack Level 8
Reset Vector
000h
Interrupt Vector
0004
0005
On-chip Program Memory
01FFh
0200h
1FFFh

FIGURE 4-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC16CE624

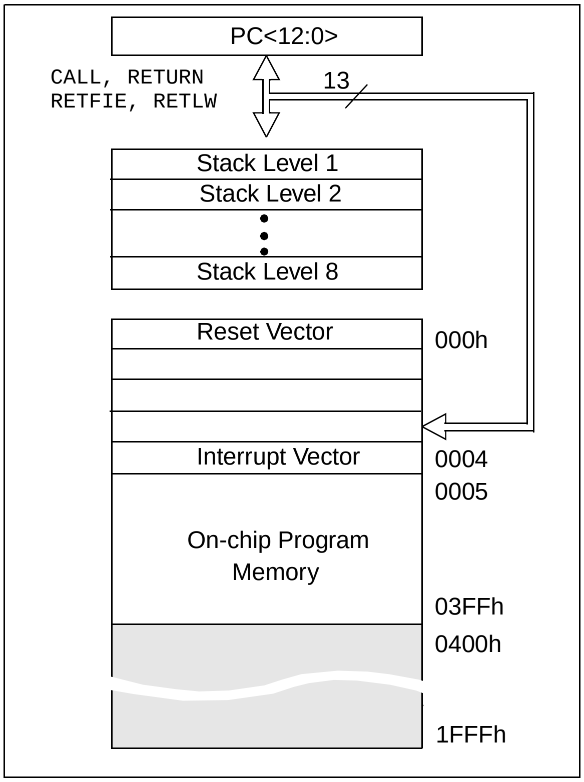
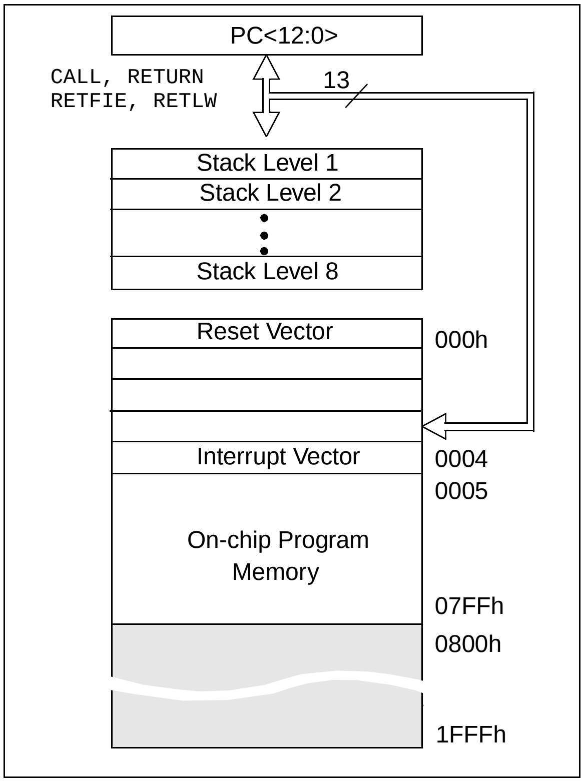

PC<12:0>
CALL, RETURN
RETFIE, RETLW
13
Stack Level 1
Stack Level 2
Stack Level 8
Reset Vector
000h
Interrupt Vector
0004
0005
On-chip Program Memory
03FFh
0400h
1FFFh

FIGURE 4-3: PROGRAM MEMORY MAP AND STACK FOR THE PIC16CE625


PC<12:0>
CALL, RETURN
RETFIE, RETLW
13
Stack Level 1
Stack Level 2
Stack Level 8
Reset Vector
000h
Interrupt Vector
0004
0005
On-chip Program Memory
07FFh
0800h
1FFFh

4.2 Data Memory Organization

The data memory (Figure 4-4 and Figure 4-5) is partitioned into two Banks which contain the General Purpose Registers and the Special Function Registers. Bank 0 is selected when the RP0 bit is cleared. Bank 1 is selected when the RP0 bit (STATUS <5>) is set. The Special Function Registers are located in the first 32 locations of each Bank. Register locations 20-7Fh (Bank0) on the PIC16CE623/624 and 20-7Fh (Bank0) and A0-BFh (Bank1) on the PIC16CE625 are General Purpose Registers implemented as static RAM. Some special purpose registers are mapped in Bank 1. In all three microcontrollers, address space F0h-FFh (Bank1) is mapped to 70-7Fh (Bank0) as common RAM.

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 96 x 8 in the PIC16CE623/624 and 128 x 8 in the PIC16CE625. Each is accessed either directly or indirectly through the File Select Register FSR (Section 4.4).

FIGURE 4-4: DATA MEMORY MAP FOR THE PIC16CE623/624

File Address	Bank 0	Bank 1	File Address
00h	INDF(1)	INDF(1)	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	▨	▨	87h
08h	▨	▨	88h
09h	▨	▨	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	PIR1	PIE1	8Ch
0Dh	▨	▨	8Dh
0Eh	▨	PCON	8Eh
0Fh	▨	▨	8Fh
10h	▨	EEINTF	90h
11h		▨	91h
12h	▨	▨	92h
13h	▨	▨	93h
14h	▨	▨	94h
15h	▨	▨	95h
16h	▨	▨	96h
17h	▨	▨	97h
18h	▨	▨	98h
19h	▨	▨	99h
1Ah	▨	▨	9Ah
1Bh	▨	▨	9Bh
1Ch	▨	▨	9Ch
1Dh	▨	▨	9Dh
1Eh	▨	▨	9Eh
1Fh	CMCON	VRCON	9Fh
20h	General Purpose Register	▨	A0h
			EFh
		Accesses 70h-7Fh	F0h
7Fh			FFh

▨ Unimplemented data memory locations, read as '0'.

Note 1: Not a physical register.

FIGURE 4-5: DATA MEMORY MAP FOR THE PIC16CE625

File Address	Bank 0	Bank 1	File Address
00h	INDF(1)	INDF(1)	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	▨	▨	87h
08h	▨	▨	88h
09h	▨	▨	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	PIR1	PIE1	8Ch
0Dh	▨	▨	8Dh
0Eh	▨	PCON	8Eh
0Fh	▨	▨	8Fh
10h	▨	EEINTF	90h
11h	▨	▨	91h
12h	▨	▨	92h
13h	▨	▨	93h
14h	▨	▨	94h
15h	▨	▨	95h
16h	▨	▨	96h
17h	▨	▨	97h
18h	▨	▨	98h
19h	▨	▨	99h
1Ah	▨	▨	9Ah
1Bh	▨	▨	9Bh
1Ch	▨	▨	9Ch
1Dh	▨	▨	9Dh
1Eh	▨	▨	9Eh
1Fh	CMCON	VRCON	9Fh
20h	General Purpose Register	General Purpose Register	A0h
			BFh
		▨	C0h
		Accesses 70h-7Fh	F0h
7Fh			FFh

▨ Unimplemented data memory locations, read as '0'.

Note 1: Not a physical register.

4.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (Table 4-1). These registers are static RAM.

The special registers can be classified into two sets (core and peripheral). The Special Function Registers associated with the "core" functions are described in this section. Those related to the operation of the peripheral features are described in the section of that peripheral feature.

TABLE 4-1: SPECIAL REGISTERS FOR THE PIC16CE62X

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR Reset	Value on all other resets[1]
Bank 0											
00h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								`xxxx xxxx`	`xxxx xxxx`
01h	TMR0	Timer0 Module's Register								`xxxx xxxx`	`uuuu uuuu`
02h	PCL	Program Counter's (PC) Least Significant Byte								`0000 0000`	`0000 0000`
03h	STATUS	IRP[2]	RP1[2]	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	`0001 1xxx`	`000q quuu`
04h	FSR	Indirect data memory address pointer								`xxxx xxxx`	`uuuu uuuu`
05h	PORTA	—	—	—	RA4	RA3	RA2	RA1	RA0	`---x 0000`	`---u 0000`
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	`xxxx xxxx`	`uuuu uuuu`
07h	Unimplemented									–	–
08h	Unimplemented									–	–
09h	Unimplemented									–	–
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					`---0 0000`	`---0 0000`
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	`0000 000x`	`0000 000u`
0Ch	PIR1	—	CMIF	—	—	—	—	—	—	`-0-- ----`	`-0-- ----`
0Dh-1Eh	Unimplemented									–	–
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	`00-- 0000`	`00-- 0000`
Bank 1											
80h	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								`xxxx xxxx`	`xxxx xxxx`
81h	OPTION	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	`1111 1111`	`1111 1111`
82h	PCL	Program Counter's (PC) Least Significant Byte								`0000 0000`	`0000 0000`
83h	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	`0001 1xxx`	`000q quuu`
84h	FSR	Indirect data memory address pointer								`xxxx xxxx`	`uuuu uuuu`
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	`---1 1111`	`---1 1111`
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	`1111 1111`	`1111 1111`
87h	Unimplemented									–	–
88h	Unimplemented									–	–
89h	Unimplemented									–	–
8Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of program counter					`---0 0000`	`---0 0000`
8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	`0000 000x`	`0000 000u`
8Ch	PIE1	—	CMIE	—	—	—	—	—	—	`-0-- ----`	`-0-- ----`
8Dh	Unimplemented									–	–
8Eh	PCON	—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOD}}$	`---- --0x`	`---- --uq`
8Fh-9Eh	Unimplemented									–	–
90h	EEINTF	—	—	—	—	—	EESCL	EESDA	EEV$_{DD}$	`---- -111`	`---- -111`
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	`000- 0000`	`000- 0000`

Legend: — = Unimplemented locations read as '0', u = unchanged, x = unknown, q = value depends on condition, shaded = unimplemented

Note 1: Other (non power-up) resets include $\overline{\text{MCLR}}$ reset, Brown-out Reset and Watchdog Timer Reset during normal operation.

Note 2: IRP & RPI bits are reserved; always maintain these bits clear.

4.2.2.1 STATUS REGISTER

The STATUS register, shown in Register 4-1, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

The STATUS register can be the destination for any instruction, like any other register. If the STATUS register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as destination may be different than intended.

For example, `CLRF STATUS` will clear the upper-three bits and set the Z bit. This leaves the status register as `000uu1uu` (where u = unchanged).

It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF` and `MOVWF` instructions are used to alter the STATUS register, because these instructions do not affect any status bit. For other instructions, not affecting any status bits, see the “Instruction Set Summary”.

Note 1: The IRP and RP1 bits (STATUS<7:6>) are not used by the PIC16CE62X and should be programmed as ’0'. Use of these bits as general purpose R/W bits is NOT recommended, since this may affect upward compatibility with future products.

Note 2: The C and DC bits operate as a $\overline{\text{Borrow}}$ and $\overline{\text{Digit Borrow}}$ out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

REGISTER 4-1: STATUS REGISTER (ADDRESS 03H OR 83H)

Reserved	Reserved	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C

bit7 bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
-n = Value at POR reset
-x = Unknown at POR reset

bit 7: **IRP**: The IRP bit is reserved on the PIC16CE62X, always maintain this bit clear.

bit 6:5 **RP<1:O>:** Register Bank Select bits (used for direct addressing)
`11` = Bank 3 (180h - 1FFh)
`10` = Bank 2 (100h - 17Fh)
`01` = Bank 1 (80h - FFh)
`00` = Bank 0 (00h - 7Fh)
Each bank is 128 bytes. The RP1 bit is reserved, always maintain this bit clear.

bit 4: **$\overline{\text{TO}}$**: Time-out bit
`1` = After power-up, `CLRWDT` instruction, or `SLEEP` instruction
`0` = A WDT time-out occurred

bit 3: **$\overline{\text{PD}}$**: Power-down bit
`1` = After power-up or by the `CLRWDT` instruction
`0` = By execution of the `SLEEP` instruction

bit 2: **Z**: Zero bit
`1` = The result of an arithmetic or logic operation is zero
`0` = The result of an arithmetic or logic operation is not zero

bit 1: **DC**: Digit carry/$\overline{\text{borrow}}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions) (for $\overline{\text{borrow}}$ the polarity is reversed)
`1` = A carry-out from the 4th low order bit of the result occurred
`0` = No carry-out from the 4th low order bit of the result

bit 0: **C**: Carry/$\overline{\text{borrow}}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions)
`1` = A carry-out from the most significant bit of the result occurred
`0` = No carry-out from the most significant bit of the result occurred
Note: For $\overline{\text{borrow}}$ the polarity is reversed. A subtraction is executed by adding the two’s complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low order bit of the source register.

4.2.2.2 OPTION REGISTER

The OPTION register is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the external RB0/INT interrupt, TMR0 and the weak pull-ups on PORTB.

Note: To achieve a 1:1 prescaler assignment for TMR0, assign the prescaler to the WDT (PSA = 1).

REGISTER 4-2: OPTION REGISTER (ADDRESS 81H)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
-n = Value at POR reset
-x = Unknown at POR reset

bit 7: **$\overline{\text{RBPU}}$**: PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6: **INTEDG**: Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin

bit 5: **T0CS**: TMR0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4: **T0SE**: TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/T0CKI pin

bit 3: **PSA**: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0: **PS<2:0>**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

4.2.2.3 INTCON REGISTER

The INTCON register is a readable and writable register which contains the various enable and flag bits for all interrupt sources except the comparator module. See Section 4.2.2.4 and Section 4.2.2.5 for a description of the comparator enable and flag bits.

Note: Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>).

REGISTER 4-3: INTCON REGISTER (ADDRESS 0BH OR 8BH)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
-n = Value at POR reset
-x = Unknown at POR reset

bit 7: **GIE:** Global Interrupt Enable bit
`1` = Enables all un-masked interrupts
`0` = Disables all interrupts

bit 6: **PEIE**: Peripheral Interrupt Enable bit
`1` = Enables all un-masked peripheral interrupts
`0` = Disables all peripheral interrupts

bit 5: **T0IE**: TMR0 Overflow Interrupt Enable bit
`1` = Enables the TMR0 interrupt
`0` = Disables the TMR0 interrupt

bit 4: **INTE**: RB0/INT External Interrupt Enable bit
`1` = Enables the RB0/INT external interrupt
`0` = Disables the RB0/INT external interrupt

bit 3: **RBIE**: RB Port Change Interrupt Enable bit
`1` = Enables the RB port change interrupt
`0` = Disables the RB port change interrupt

bit 2: **T0IF**: TMR0 Overflow Interrupt Flag bit
`1` = TMR0 register has overflowed (must be cleared in software)
`0` = TMR0 register did not overflow

bit 1: **INTF**: RB0/INT External Interrupt Flag bit
`1` = The RB0/INT external interrupt occurred (must be cleared in software)
`0` = The RB0/INT external interrupt did not occur

bit 0: **RBIF**: RB Port Change Interrupt Flag bit
`1` = When at least one of the RB<7:4> pins changed state (must be cleared in software)
`0` = None of the RB<7:4> pins have changed state

4.2.2.4 PIE1 REGISTER

This register contains the individual enable bit for the comparator interrupt.

REGISTER 4-4: PIE1 REGISTER (ADDRESS 8CH)

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	CMIE	—	—	—	—	—	—
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
-n = Value at POR reset
-x = Unknown at POR reset

bit 7: **Unimplemented:** Read as ’0’

bit 6: **CMIE**: Comparator Interrupt Enable bit
1 = Enables the Comparator interrupt
0 = Disables the Comparator interrupt

bit 5-0: **Unimplemented**: Read as ’0’

4.2.2.5 PIR1 REGISTER

This register contains the individual flag bit for the comparator interrupt.

Note: Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

REGISTER 4-5: PIR1 REGISTER (ADDRESS 0CH)

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	CMIF	—	—	—	—	—	—
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
-n = Value at POR reset
-x = Unknown at POR reset

bit 7: **Unimplemented:** Read as ’0’

bit 6: **CMIF**: Comparator Interrupt Flag bit
1 = Comparator input has changed
0 = Comparator input has not changed

bit 5-0: **Unimplemented**: Read as ’0’

4.2.2.6 PCON REGISTER

The PCON register contains flag bits to differentiate between a Power-on Reset, an external $\overline{\text{MCLR}}$ reset, WDT reset or a Brown-out Reset.

> **Note:** $\overline{\text{BOD}}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent resets to see if $\overline{\text{BOD}}$ is cleared, indicating a brown-out has occurred. The $\overline{\text{BOD}}$ status bit is a "don't care" and is not necessarily predictable if the brown-out circuit is disabled (by programming BODEN bit in the configuration word).

REGISTER 4-6: PCON REGISTER (ADDRESS 8Eh)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOD}}$
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as '0'
-n = Value at POR reset
-x = Unknown at POR reset

bit 7-2: **Unimplemented:** Read as '0'

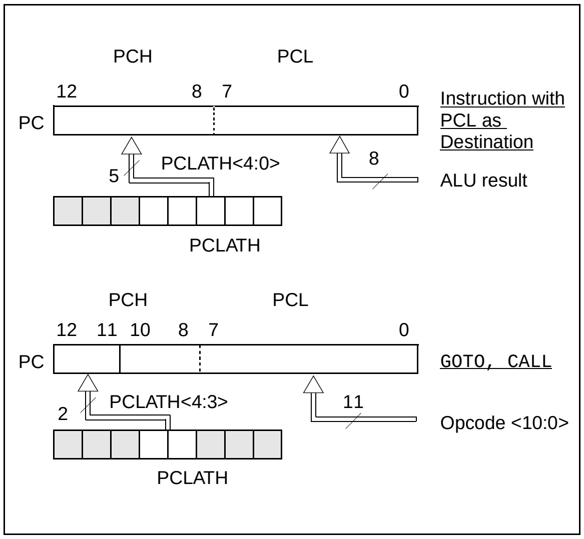
bit 1: **$\overline{\text{POR}}$**: Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0: **$\overline{\text{BOD}}$**: Brown-out Reset Status bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

4.3 PCL and PCLATH

The program counter (PC) is 13 bits wide. The low byte comes from the PCL register, which is a readable and writable register. The high byte (PC<12:8>) is not directly readable or writable and comes from PCLATH. On any reset, the PC is cleared. Figure 4-6 shows the two situations for the loading of the PC. The upper example in the figure shows how the PC is loaded on a write to PCL (PCLATH<4:0> → PCH). The lower example in the figure shows how the PC is loaded during a `CALL` or `GOTO` instruction (PCLATH<4:3> → PCH).

FIGURE 4-6: LOADING OF PC IN DIFFERENT SITUATIONS



4.3.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (`ADDWF PCL`). When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256 byte block). Refer to the application note, *“Implementing a Table Read"* (AN556).

4.3.2 STACK

The PIC16CE62X family has an 8 level deep x 13-bit wide hardware stack (Figure 4-2 and Figure 4-3). The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed onto the stack when a `CALL` instruction is executed or an interrupt causes a branch. The stack is POPed in the event of a `RETURN`, `RETLW` or a `RETFIE` instruction execution. PCLATH is not affected by a PUSH or POP operation.

The stack operates as a circular buffer. This means that after the stack has been PUSHed eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

Note 1: There are no STATUS bits to indicate stack overflow or stack underflow conditions.

Note 2: There are no instruction/mnemonics called `PUSH` or `POP`. These are actions that occur from the execution of the `CALL`, `RETURN`, `RETLW` and `RETFIE` instructions or the vectoring to an interrupt address.

4.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

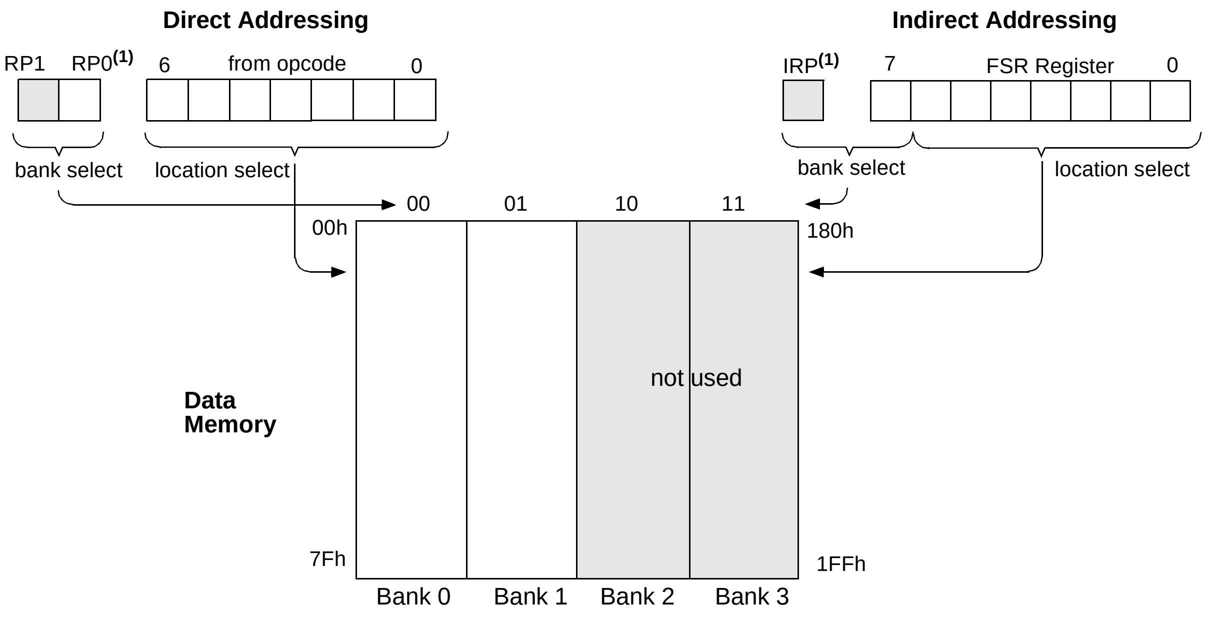
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the File Select Register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 4-7. However, IRP is not used in the PIC16CE62X.

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 4-1.

EXAMPLE 4-1: INDIRECT ADDRESSING

```
          movlw  0x20    ;initialize pointer
          movwf  FSR     ;to RAM
NEXT      clrf   INDF    ;clear INDF register
          incf   FSR     ;inc pointer
          btfss  FSR,4   ;all done?
          goto   NEXT    ;no clear next
                         ;yes continue
CONTINUE:
```

FIGURE 4-7: DIRECT/INDIRECT ADDRESSING PIC16CE62X


Direct Addressing
RP1
RP0(1)
6
from opcode
0
bank select
location select
Indirect Addressing
IRP(1)
7
FSR Register
0
bank select
location select
00
01
10
11
00h
180h
Data Memory
not used
7Fh
1FFh
Bank 0
Bank 1
Bank 2
Bank 3

For memory map detail see Figure 4-4 and Figure 4-5.

Note 1: The RP1 and IRP bits are reserved; always maintain these bits clear.

NOTES:

5.0 I/O PORTS

The PIC16CE62X parts have two ports, PORTA and PORTB. Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

5.1 PORTA and TRISA Registers

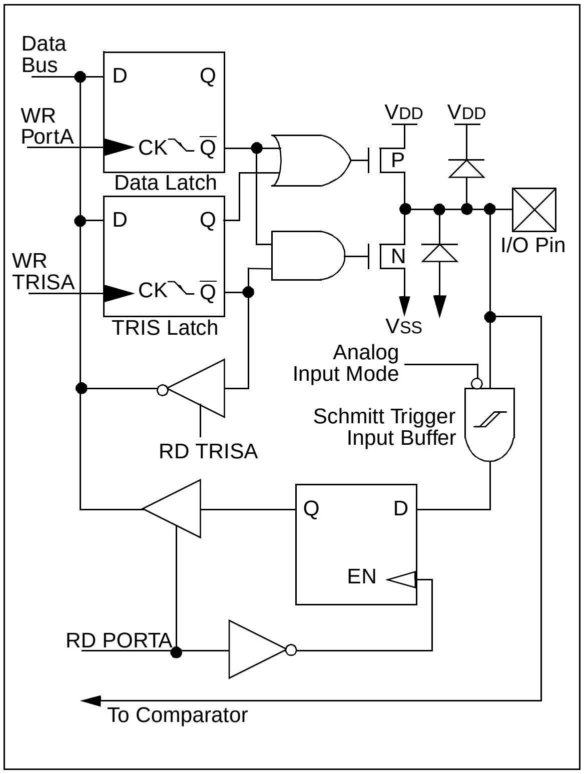
PORTA is a 5-bit wide latch. RA4 is a Schmitt Trigger input and an open drain output. Port RA4 is multiplexed with the T0CKI clock input. All other RA port pins have Schmitt Trigger input levels and full CMOS output drivers. All pins have data direction bits (TRIS registers), which can configure these pins as input or output.

A '1' in the TRISA register puts the corresponding output driver in a hi- impedance mode. A '0' in the TRISA register puts the contents of the output latch on the selected pin(s).

Reading the PORTA register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

The PORTA pins are multiplexed with comparator and voltage reference functions. The operation of these pins are selected by control bits in the CMCON (Comparator Control Register) register and the VRCON (Voltage Reference Control Register) register. When selected as a comparator input, these pins will read as '0's.

FIGURE 5-1: BLOCK DIAGRAM OF RA<1:0> PINS


Data Bus
D
Q
WR PortA
CK
Q̄
Data Latch
D
Q
WR TRISA
CK
Q̄
TRIS Latch
VDD
VDD
P
N
I/O Pin
VSS
Analog Input Mode
Schmitt Trigger Input Buffer
RD TRISA
Q
D
EN
RD PORTA
To Comparator

Note:	On reset, the TRISA register is set to all inputs. The digital inputs are disabled and the comparator inputs are forced to ground to reduce excess current consumption.

TRISA controls the direction of the RA pins, even when they are being used as comparator inputs. The user must make sure to keep the pins configured as inputs when using them as comparator inputs.

The RA2 pin will also function as the output for the voltage reference. When in this mode, the VREF pin is a very high impedance output. The user must configure TRISA<2> bit as an input and use high impedance loads.

In one of the comparator modes defined by the CMCON register, pins RA3 and RA4 become outputs of the comparators. The TRISA<4:3> bits must be cleared to enable outputs to use this function.

EXAMPLE 5-1: INITIALIZING PORTA

```
CLRF   PORTA        ;Initialize PORTA by setting
                    ;output data latches
MOVLW  0X07         ;Turn comparators off and
MOVWF  CMCON        ;enable pins for I/O
                    ;functions
BSF    STATUS, RP0 ;Select Bank1
MOVLW  0x1F         ;Value used to initialize
                    ;data direction
MOVWF  TRISA        ;Set RA<4:0> as inputs
                    ;TRISA<7:5> are always
                    ;read as ’0’.
```

FIGURE 5-2: BLOCK DIAGRAM OF RA2 PIN

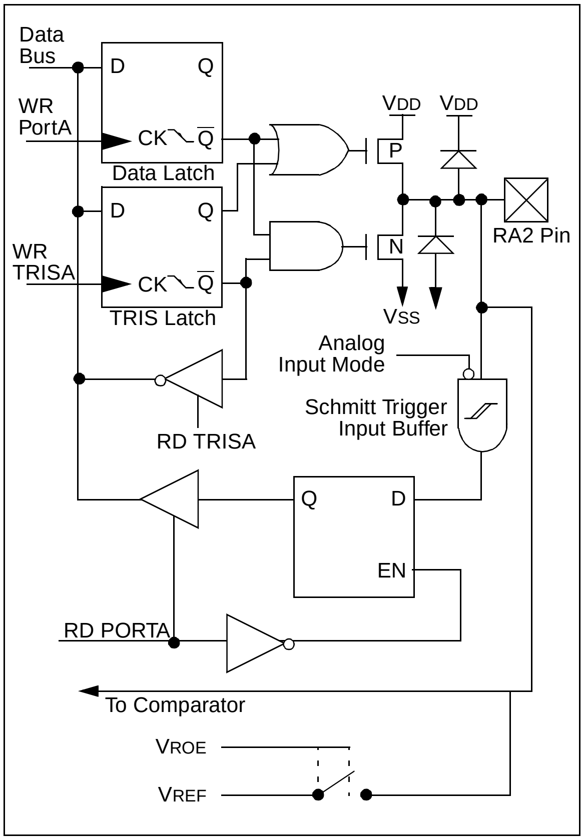

Data Bus
D
Q
WR PortA
CK
Q̄
Data Latch
D
Q
WR TRISA
CK
Q̄
TRIS Latch
VDD
VDD
P
N
RA2 Pin
VSS
Analog Input Mode
Schmitt Trigger Input Buffer
RD TRISA
Q
D
EN
RD PORTA
To Comparator
VROE
VREF

FIGURE 5-3: BLOCK DIAGRAM OF RA3 PIN

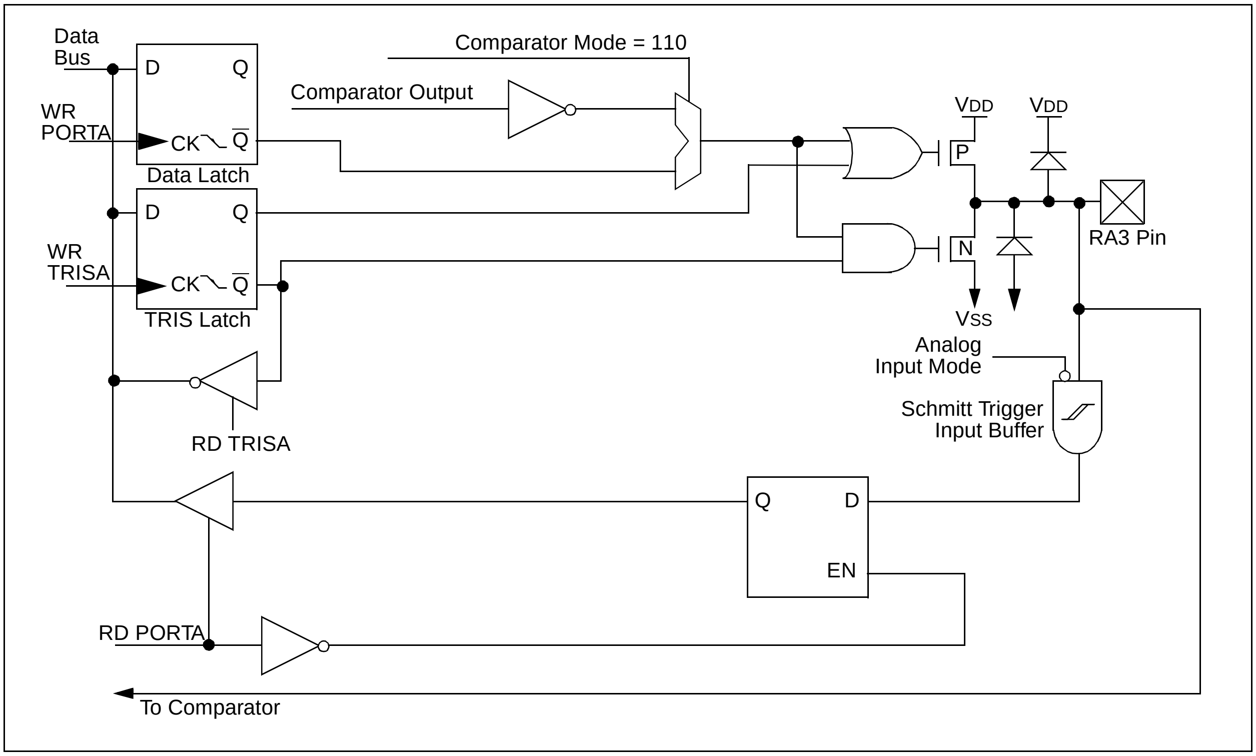

Data Bus
D
Q
WR PORTA
CK
Q̅
Data Latch
D
Q
WR TRISA
CK
Q̅
TRIS Latch
RD TRISA
Comparator Mode = 110
Comparator Output
VDD
VDD
P
N
VSS
RA3 Pin
Analog Input Mode
Schmitt Trigger Input Buffer
Q
D
EN
RD PORTA
To Comparator

FIGURE 5-4: BLOCK DIAGRAM OF RA4 PIN

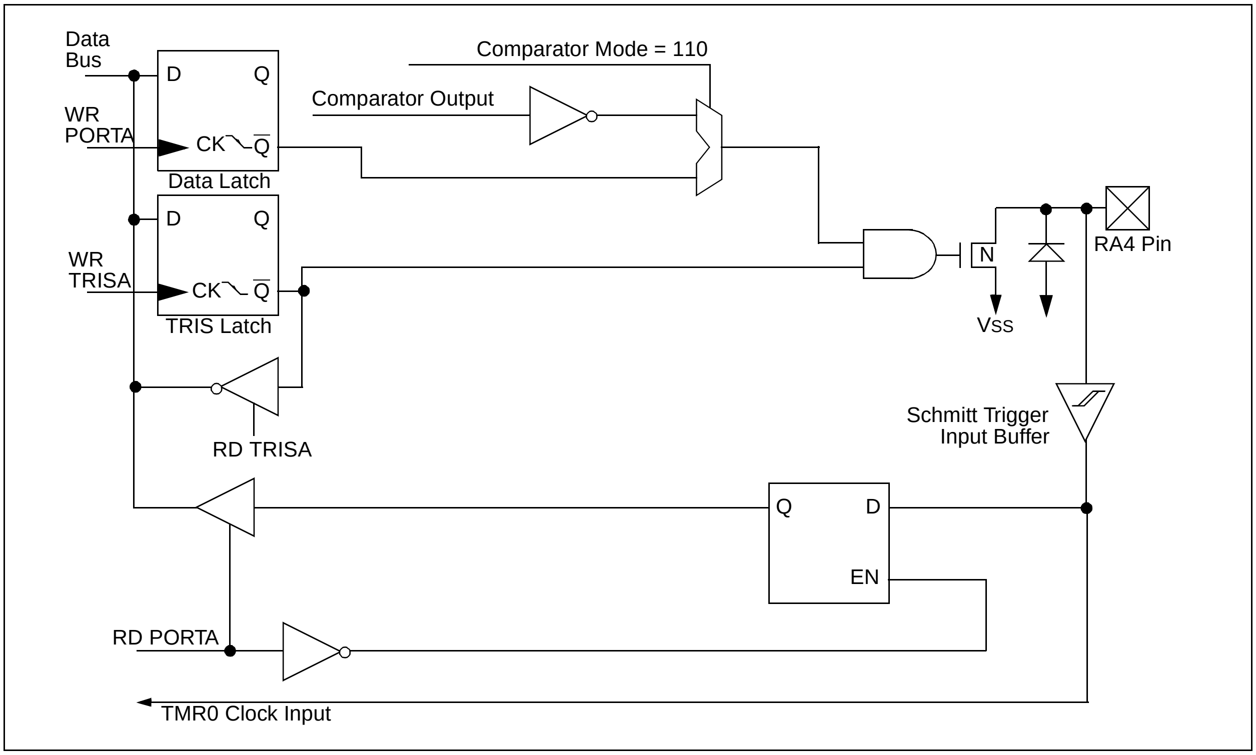

Data Bus
D
Q
WR PORTA
CK
Q̅
Data Latch
D
Q
WR TRISA
CK
Q̅
TRIS Latch
RD TRISA
Comparator Mode = 110
Comparator Output
N
VSS
RA4 Pin
Schmitt Trigger Input Buffer
Q
D
EN
RD PORTA
TMR0 Clock Input

TABLE 5-1: PORTA FUNCTIONS

Name	Bit #	Buffer Type	Function
RA0/AN0	bit0	ST	Input/output or comparator input
RA1/AN1	bit1	ST	Input/output or comparator input
RA2/AN2/VREF	bit2	ST	Input/output or comparator input or VREF output
RA3/AN3	bit3	ST	Input/output or comparator input/output
RA4/T0CKI	bit4	ST	Input/output or external clock input for TMR0 or comparator output. Output is open drain type.

Legend: ST = Schmitt Trigger input

TABLE 5-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR	Value on All Other Resets
05h	PORTA	—	—	—	RA4	RA3	RA2	RA1	RA0	`---x 0000`	`---u 0000`
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	`---1 1111`	`---1 1111`
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	`00-- 0000`	`00-- 0000`
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	`000- 0000`	`000- 0000`

Legend: — = Unimplemented locations, read as ‘0’, x = unknown, u = unchanged

Note: Shaded bits are not used by PORTA.

5.2 PORTB and TRISB Registers

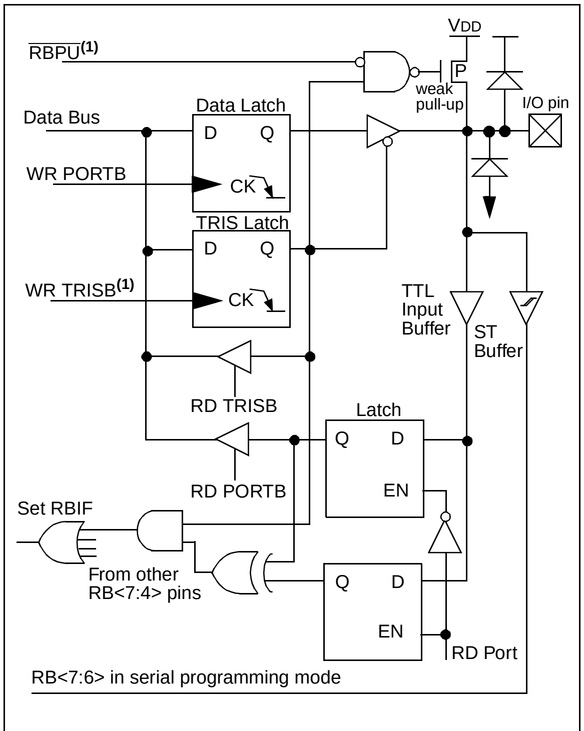
PORTB is an 8-bit wide, bi-directional port. The corresponding data direction register is TRISB. A '1' in the TRISB register puts the corresponding output driver in a high impedance mode. A '0' in the TRISB register puts the contents of the output latch on the selected pin(s).

Reading PORTB register reads the status of the pins, whereas writing to it will write to the port latch. All write operations are read-modify-write operations. So a write to a port implies that the port pins are first read, then this value is modified and written to the port data latch.

Each of the PORTB pins has a weak internal pull-up (≈200 µA typical). A single control bit can turn on all the pull-ups. This is done by clearing the $\overline{\text{RBPU}}$ (OPTION<7>) bit. The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on Power-on Reset.

Four of PORTB's pins, RB<7:4>, have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB<7:4> pin configured as an output is excluded from the interrupt on change comparison). The input pins of RB<7:4> are compared with the old value latched on the last read of PORTB. The "mismatch" outputs of RB<7:4> are OR'ed together to generate the RBIF interrupt (flag latched in INTCON<0>).

FIGURE 5-5: BLOCK DIAGRAM OF RB<7:4> PINS



$\overline{\text{RBPU}}^{(1)}$
VDD
P
weak pull-up
I/O pin
Data Bus
Data Latch
D
Q
WR PORTB
CK
TRIS Latch
D
Q
WR TRISB(1)
CK
TTL Input Buffer
ST Buffer
RD TRISB
Latch
Q
D
RD PORTB
EN
Set RBIF
From other RB<7:4> pins
Q
D
EN
RD Port
RB<7:6> in serial programming mode

Note 1: TRISB = 1 enables weak pull-up if $\overline{\text{RBPU}}$ = '0' (OPTION<7>).

This interrupt can wake the device from SLEEP. The user, in the interrupt service routine, can clear the interrupt in the following manner:

a) Any read or write of PORTB. This will end the mismatch condition.

b) Clear flag bit RBIF.

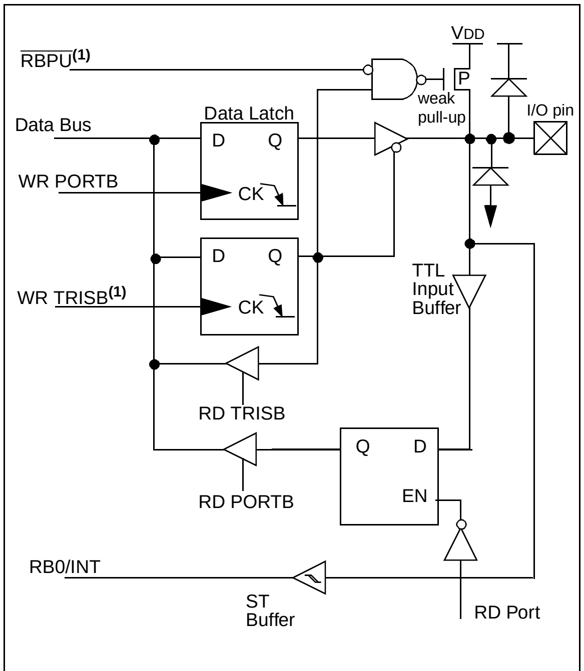
A mismatch condition will continue to set flag bit RBIF. Reading PORTB will end the mismatch condition and allow flag bit RBIF to be cleared.

This interrupt on mismatch feature, together with software configurable pull-ups on these four pins allow easy interface to a key pad and make it possible for wake-up on key-depression. (See AN552, "Implementing Wake-Up on Key Strokes".)

Note:	If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not get set.

The interrupt on change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt on change feature. Polling of PORTB is not recommended while using the interrupt on change feature.

FIGURE 5-6: BLOCK DIAGRAM OF RB<3:0> PINS



VDD
$\overline{\text{RBPU}}^{(1)}$
P
weak pull-up
I/O pin
Data Bus
Data Latch
D
Q
WR PORTB
CK
D
Q
WR TRISB(1)
CK
TTL Input Buffer
RD TRISB
Q
D
RD PORTB
EN
RB0/INT
ST Buffer
RD Port

Note 1: TRISB = 1 enables weak pull-up if $\overline{\text{RBPU}}$ = '0' (OPTION<7>).

TABLE 5-3: PORTB FUNCTIONS

Name	Bit #	Buffer Type	Function
RB0/INT	bit0	TTL/ST(1)	Input/output or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3	bit3	TTL	Input/output pin. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt on change). Internal software programmable weak pull-up.
RB6	bit6	TTL/ST(2)	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming clock pin.
RB7	bit7	TTL/ST(2)	Input/output pin (with interrupt on change). Internal software programmable weak pull-up. Serial programming data pin.

Legend: ST = Schmitt Trigger, TTL = TTL input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

Note 2: This buffer is a Schmitt Trigger input when used in serial programming mode.

TABLE 5-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR	Value on All Other Resets
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	`xxxx xxxx`	`uuuu uuuu`
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	`1111 1111`	`1111 1111`
81h	OPTION	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	`1111 1111`	`1111 1111`

Legend: u = unchanged, x = unknown

Note: Shaded bits are not used by PORTB.

5.3 I/O Programming Considerations

5.3.1 BI-DIRECTIONAL I/O PORTS

Any instruction which writes, operates internally as a read followed by a write operation. The `BCF` and `BSF` instructions, for example, read the register into the CPU, execute the bit operation and write the result back to the register. Caution must be used when these instructions are applied to a port with both inputs and outputs defined. For example, a `BSF` operation on bit5 of PORTB will cause all eight bits of PORTB to be read into the CPU. Then the `BSF` operation takes place on bit5 and PORTB is written to the output latches. If another bit of PORTB is used as a bidirectional I/O pin (i.e., bit0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and re-written to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit0 is switched into output mode later on, the content of the data latch may now be unknown.

Reading the port register, reads the values of the port pins. Writing to the port register writes the value to the port latch. When using read modify write instructions (i.e., `BCF, BSF`, etc.) on a port, the value of the port pins is read, the desired operation is done to this value, and this value is then written to the port latch.

Example 5-2 shows the effect of two sequential read-modify-write instructions (i.e., `BCF, BSF`, etc.) on an I/O port.

A pin actively outputting a Low or High should not be driven from external devices at the same time in order to change the level on this pin (“wired-or”, “wired-and”). The resulting high output currents may damage the chip.

EXAMPLE 5-2: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

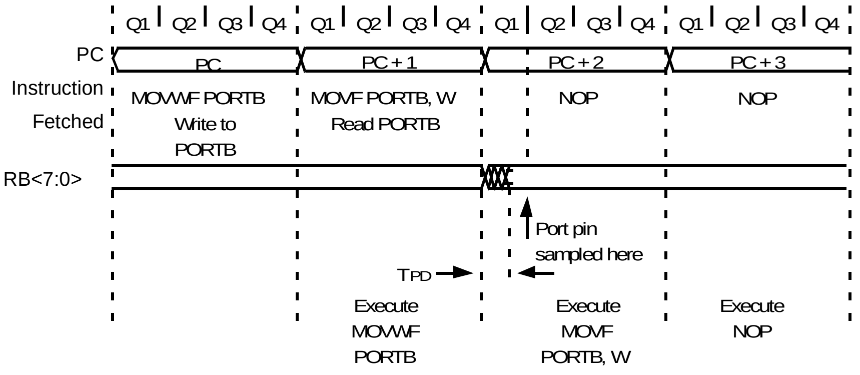
```
; Initial PORT settings:  PORTB<7:4> Inputs
;
;                         PORTB<3:0> Outputs
; PORTB<7:6> have external pull-up and are not
; connected to other circuitry
;
;                         PORT latch  PORT pins
;                         ----------  ----------

     BCF PORTB, 7         ; 01pp pppp  11pp pppp
     BCF PORTB, 6         ; 10pp pppp  11pp pppp
     BSF STATUS,RP0       ;
     BCF TRISB, 7         ; 10pp pppp  11pp pppp
     BCF TRISB, 6         ; 10pp pppp  10pp pppp
;
; Note that the user may have expected the pin
; values to be 00pp pppp. The 2nd BCF caused
; RB7 to be latched as the pin value (High).
```

5.3.2 SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 5-7). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should allow the pin voltage to stabilize (load dependent) before the next instruction causes that file to be read into the CPU. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with an NOP or another instruction not accessing this I/O port.

FIGURE 5-7: SUCCESSIVE I/O OPERATION



Note:

This example shows write to PORTB followed by a read from PORTB.

Note that:

data setup time = (0.25 T_{CY} - T_{PD}) where T_{CY} = instruction cycle and T_{PD} = propagation delay of Q1 cycle to output valid.

Therefore, at higher clock frequencies, a write followed by a read may be problematic.

6.0 EEPROM PERIPHERAL OPERATION

The PIC16CE623/624/625 each have 128 bytes of EEPROM data memory. The EEPROM data memory supports a bi-directional, 2-wire bus and data transmission protocol. These two-wires are serial data (SDA) and serial clock (SCL), and are mapped to bit1 and bit2, respectively, of the EEINTF register (SFR 90h). In addition, the power to the EEPROM can be controlled using bit0 (EEVDD) of the EEINTF register. For most applications, all that is required is calls to the following functions:

```
; Byte_Write: Byte write routine
;       Inputs: EEPROM Address     EEADDR
;               EEPROM Data        EEDATA
;       Outputs:   Return 01 in W if OK, else
;                  return 00 in W
;
; Read_Current: Read EEPROM at address
currently held by EE device.
;       Inputs: NONE
;       Outputs:   EEPROM Data    EEDATA
;                  Return 01 in W if OK, else
;                  return 00 in W
;
; Read_Random: Read EEPROM byte at supplied
; address
;       Inputs: EEPROM Address     EEADDR
;       Outputs:   EEPROM Data    EEDATA
;                  Return 01 in W if OK,
;                  else return 00 in W
```

The code for these functions is available on our web site (www.microchip.com). The code will be accessed by either including the source code FL62XINC.ASM or by linking FLASH62X.ASM. FLASH62.IMC provides external definition to the calling program.

6.0.1 SERIAL DATA

SDA is a bi-directional pin used to transfer addresses and data into and data out of the memory.

For normal data transfer, SDA is allowed to change only during SCL low. Changes during SCL high are reserved for indicating the START and STOP conditions.

6.0.2 SERIAL CLOCK

This SCL input is used to synchronize the data transfer to and from the memory.

6.0.3 EEINTF REGISTER

The EEINTF register (SFR 90h) controls the access to the EEPROM. Register 6-1 details the function of each bit. User code must generate the clock and data signals.

REGISTER 6-1: EEINTF REGISTER (ADDRESS 90h)

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1
—	—	—	—	—	EESCL	EESDA	EEVDD
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
- n = Value at POR reset

bit 7-3: **Unimplemented:** Read as ’0’

bit 2: **EESCL**: Clock line to the EEPROM
1 = Clock high
0 = Clock low

bit 1: **EESDA**: Data line to EEPROM
1 = Data line is high (pin is tri-stated, line is pulled high by a pull-up resistor)
0 = Data line is low

bit 0: **EEVDD**: VDD control bit for EEPROM
1 = VDD is turned on to EEPROM
0 = VDD is turned off to EEPROM (all pins are tri-stated and the EEPROM is powered down)

Note: EESDA, EESCL and EEVDD will read ‘0’ if EEVDD is turned off.

6.1 Bus Characteristics

In this section, the term “processor” refers to the portion of the PIC16CE62X that interfaces to the EEPROM through software manipulating the EEINTF register. The following **bus protocol** is to be used with the EEPROM data memory.

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is HIGH will be interpreted by the EEPROM as a START or STOP condition.

Accordingly, the following bus conditions have been defined (Figure 6-1).

6.1.1 BUS NOT BUSY (A)

Both data and clock lines remain HIGH.

6.1.2 START DATA TRANSFER (B)

A HIGH to LOW transition of the SDA line while the clock (SCL) is HIGH determines a START condition. All commands must be preceded by a START condition.

6.1.3 STOP DATA TRANSFER (C)

A LOW to HIGH transition of the SDA line while the clock (SCL) is HIGH determines a STOP condition. All operations must be ended with a STOP condition.

6.1.4 DATA VALID (D)

The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal.

The data on the line must be changed during the LOW period of the clock signal. There is one bit of data per clock pulse.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of the data bytes transferred between the START and STOP conditions is determined by the processor and is theoretically unlimited, although only the last sixteen will be stored when doing a write operation. When an overwrite does occur, it will replace data in a first-in, first-out fashion.

6.1.5 ACKNOWLEDGE

The EEPROM will generate an acknowledge after the reception of each byte. The processor must generate an extra clock pulse which is associated with this acknowledge bit.

Note:	Acknowledge bits are not generated if an internal programming cycle is in progress.

When the EEPROM acknowledges, it pulls down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. The processor must signal an end of data to the EEPROM by not generating an acknowledge bit on the last byte that has been clocked out of the EEPROM. In this case, the EEPROM must leave the data line HIGH to enable the processor to generate the STOP condition (Figure 6-2).

FIGURE 6-1: DATA TRANSFER SEQUENCE ON THE SERIAL BUS

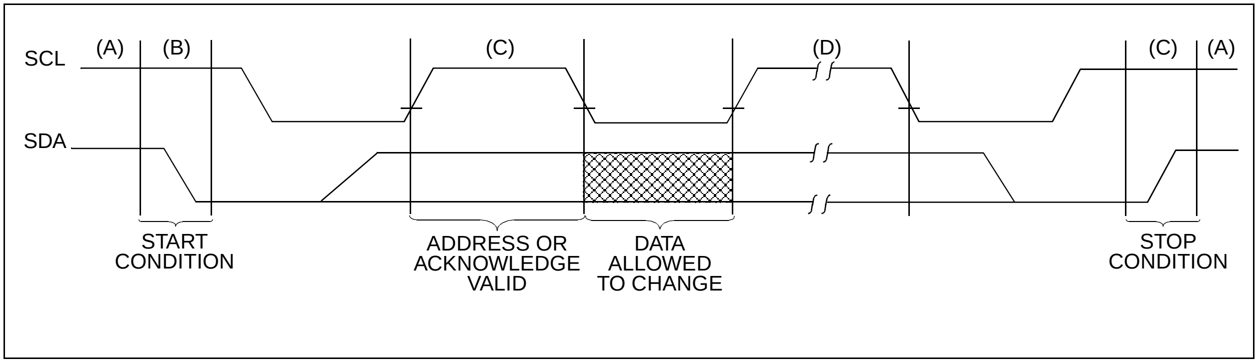
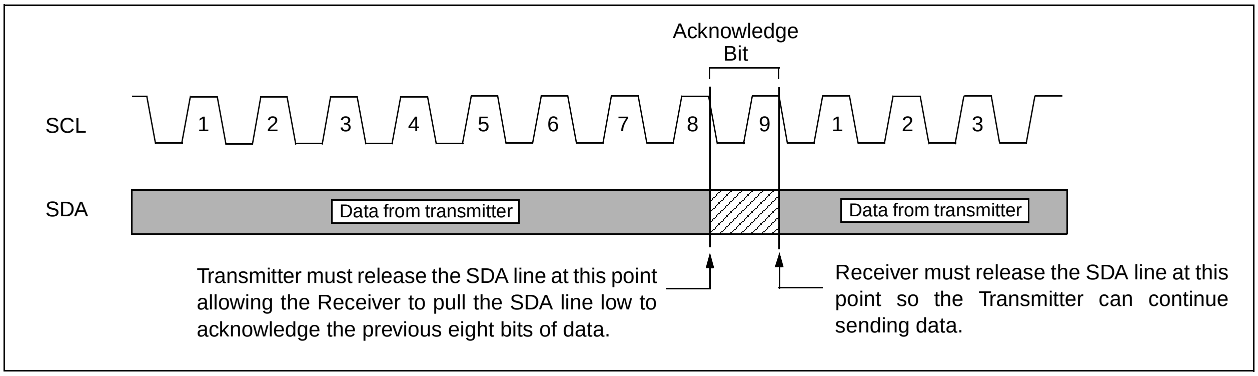

SCL
SDA
(A)
(B)
(C)
(D)
(C)
(A)
START CONDITION
ADDRESS OR ACKNOWLEDGE VALID
DATA ALLOWED TO CHANGE
STOP CONDITION

FIGURE 6-2: ACKNOWLEDGE TIMING

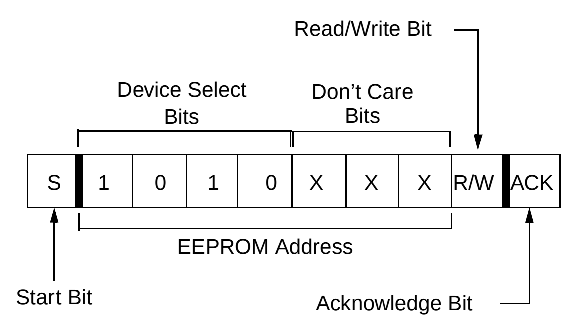

Acknowledge Bit
SCL
1
2
3
4
5
6
7
8
9
1
2
3
SDA
Data from transmitter
Data from transmitter
Transmitter must release the SDA line at this point allowing the Receiver to pull the SDA line low to acknowledge the previous eight bits of data.
Receiver must release the SDA line at this point so the Transmitter can continue sending data.

6.2 Device Addressing

After generating a START condition, the processor transmits a control byte consisting of a EEPROM address and a Read/$\overline{\text{Write}}$ bit that indicates what type of operation is to be performed. The EEPROM address consists of a 4-bit device code (1010) followed by three don't care bits.

The last bit of the control byte determines the operation to be performed. When set to a one, a read operation is selected, and when set to a zero, a write operation is selected. (Figure 6-3). The bus is monitored for its corresponding EEPROM address all the time. It generates an acknowledge bit if the EEPROM address was true and it is not in a programming mode.

FIGURE 6-3: CONTROL BYTE FORMAT


Read/Write Bit
Device Select Bits
Don't Care Bits
S
1
0
1
0
X
X
X
R/W
ACK
EEPROM Address
Start Bit
Acknowledge Bit

6.3 Write Operations

6.3.1 BYTE WRITE

Following the start signal from the processor, the device code (4 bits), the don't care bits (3 bits), and the R/$\overline{W}$ bit, which is a logic low, is placed onto the bus by the processor. This indicates to the EEPROM that a byte with a word address will follow after it has generated an acknowledge bit during the ninth clock cycle. Therefore, the next byte transmitted by the processor is the word address and will be written into the address pointer of the EEPROM. After receiving another acknowledge signal from the EEPROM, the processor will transmit the data word to be written into the addressed memory location. The EEPROM acknowledges again and the processor generates a stop condition. This initiates the internal write cycle, and during this time, the EEPROM will not generate acknowledge signals (Figure 6-5).

6.3.2 PAGE WRITE

The write control byte, word address and the first data byte are transmitted to the EEPROM in the same way as in a byte write. But instead of generating a stop condition, the processor transmits up to eight data bytes to the EEPROM, which are temporarily stored in the on-chip page buffer and will be written into the memory after the processor has transmitted a stop condition. After the receipt of each word, the three lower order address pointer bits are internally incremented by one. The higher order five bits of the word address remains constant. If the processor should transmit more than eight words prior to generating the stop condition, the address counter will roll over and the previously received data will be overwritten. As with the byte write operation, once the stop condition is received, an internal write cycle will begin (Figure 6-6).

6.4 Acknowledge Polling

Since the EEPROM will not acknowledge during a write cycle, this can be used to determine when the cycle is complete (this feature can be used to maximize bus throughput). Once the stop condition for a write command has been issued from the processor, the EEPROM initiates the internally timed write cycle. ACK polling can be initiated immediately. This involves the processor sending a start condition followed by the control byte for a write command (R/$\overline{W}$ = 0). If the device is still busy with the write cycle, then no ACK will be returned. If no ACK is returned, then the start bit and control byte must be re-sent. If the cycle is complete, then the device will return the ACK and the processor can then proceed with the next read or write command. See Figure 6-4 for flow diagram.

FIGURE 6-4: ACKNOWLEDGE POLLING FLOW

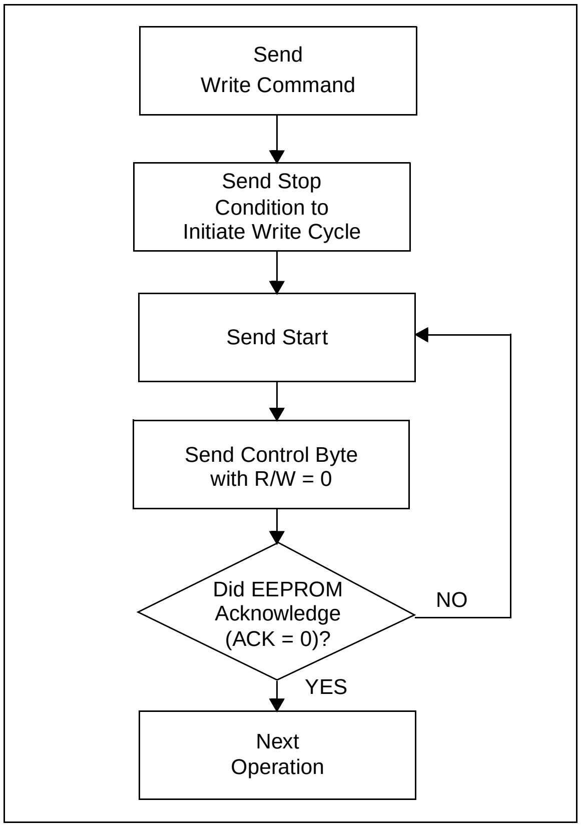


FIGURE 6-5: BYTE WRITE

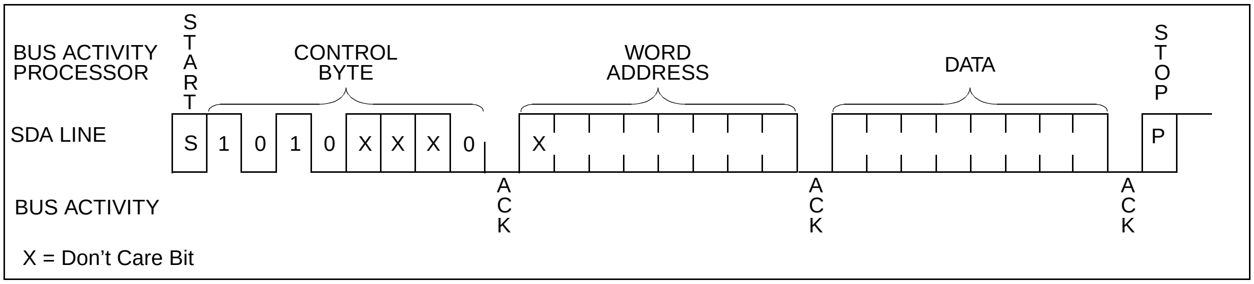
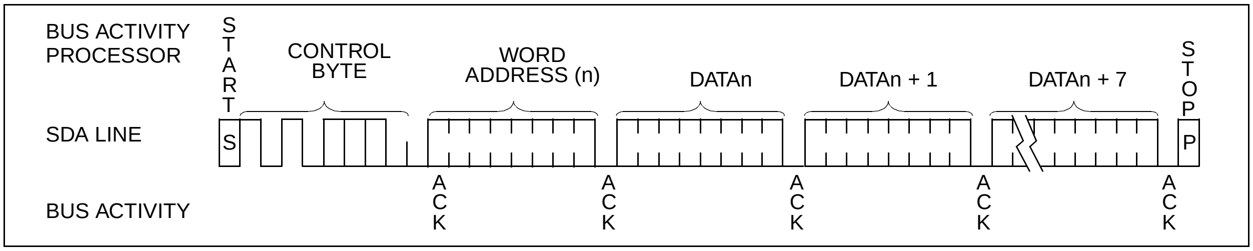


FIGURE 6-6: PAGE WRITE



6.5 Read Operation

Read operations are initiated in the same way as write operations with the exception that the R/$\overline{W}$ bit of the EEPROM address is set to one. There are three basic types of read operations: current address read, random read, and sequential read.

6.6 Current Address Read

The EEPROM contains an address counter that maintains the address of the last word accessed, internally incremented by one. Therefore, if the previous access (either a read or write operation) was to address n, the next current address read operation would access data from address n + 1. Upon receipt of the EEPROM address with R/$\overline{W}$ bit set to one, the EEPROM issues an acknowledge and transmits the eight bit data word. The processor will not acknowledge the transfer, but does generate a stop condition and the EEPROM discontinues transmission (Figure 6-7).

6.7 Random Read

Random read operations allow the processor to access any memory location in a random manner. To perform this type of read operation, first the word address must be set. This is done by sending the word address to the EEPROM as part of a write operation. After the word address is sent, the processor generates a start condition following the acknowledge. This terminates the write operation, but not before the internal address pointer is set. Then the processor issues the control byte again, but with the R/$\overline{W}$ bit set to a one. The EEPROM will then issue an acknowledge and transmits the eight bit data word. The processor will not acknowledge the transfer, but does generate a stop condition and the EEPROM discontinues transmission (Figure 6-8).

6.8 Sequential Read

Sequential reads are initiated in the same way as a random read except that after the EEPROM transmits the first data byte, the processor issues an acknowledge as opposed to a stop condition in a random read. This directs the EEPROM to transmit the next sequentially addressed 8-bit word (Figure 6-9).

To provide sequential reads, the EEPROM contains an internal address pointer which is incremented by one at the completion of each operation. This address pointer allows the entire memory contents to be serially read during one operation.

6.9 Noise Protection

The EEPROM employs a VCC threshold detector circuit, which disables the internal erase/write logic if the VCC is below 1.5 volts at nominal conditions.

The SCL and SDA inputs have Schmitt trigger and filter circuits, which suppress noise spikes to assure proper device operation even on a noisy bus.

FIGURE 6-7: CURRENT ADDRESS READ

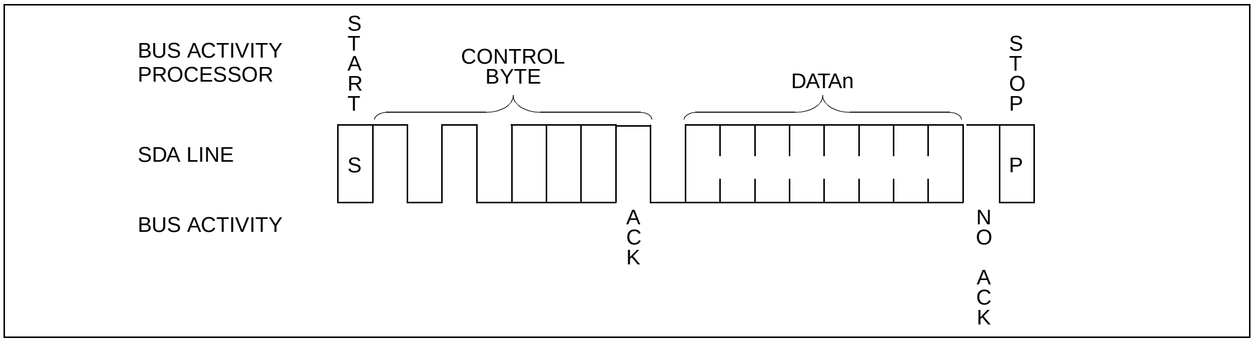

BUS ACTIVITY PROCESSOR
START
CONTROL BYTE
DATAn
STOP
SDA LINE
S
P
BUS ACTIVITY
ACK
NO ACK

FIGURE 6-8: RANDOM READ

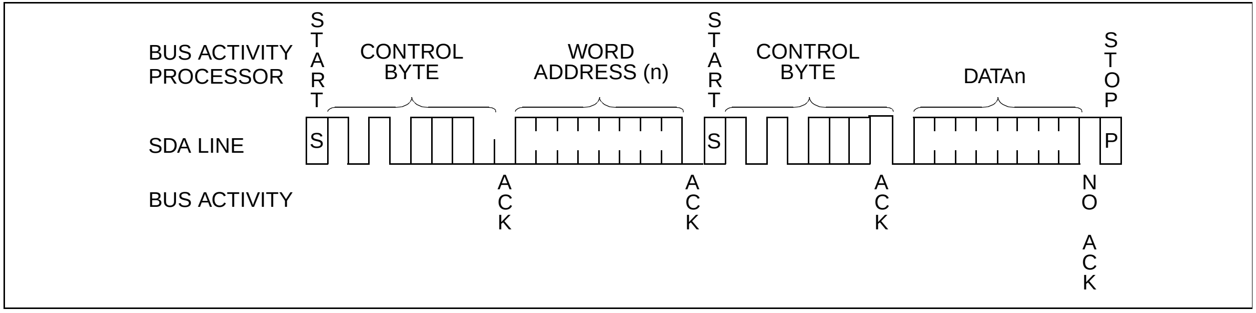
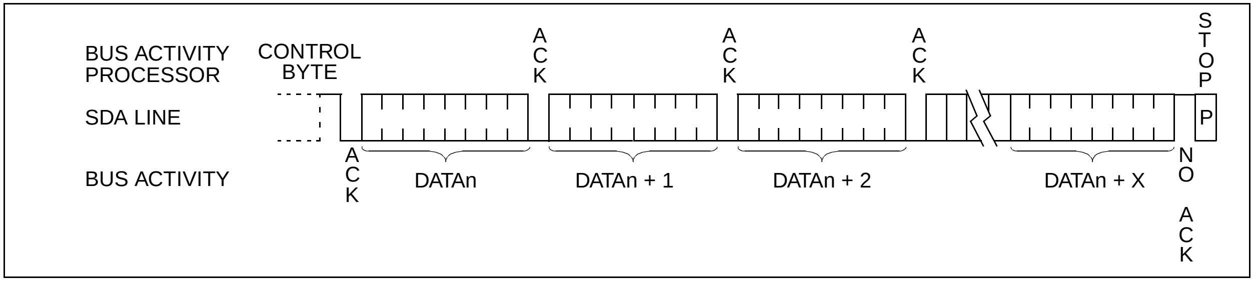

BUS ACTIVITY PROCESSOR
START
CONTROL BYTE
WORD ADDRESS (n)
START
CONTROL BYTE
DATAn
STOP
SDA LINE
S
S
P
BUS ACTIVITY
ACK
ACK
ACK
NO ACK

FIGURE 6-9: SEQUENTIAL READ


BUS ACTIVITY PROCESSOR
CONTROL BYTE
ACK
ACK
ACK
STOP
SDA LINE
P
BUS ACTIVITY
ACK
DATAn
DATAn + 1
DATAn + 2
DATAn + X
NO ACK

7.0 TIMER0 MODULE

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 7-1 is a simplified block diagram of the Timer0 module.

Timer mode is selected by clearing the T0CS bit (OPTION<5>). In timer mode, the TMR0 will increment every instruction cycle (without prescaler). If Timer0 is written, the increment is inhibited for the following two cycles (Figure 7-2 and Figure 7-3). The user can work around this by writing an adjusted value to TMR0.

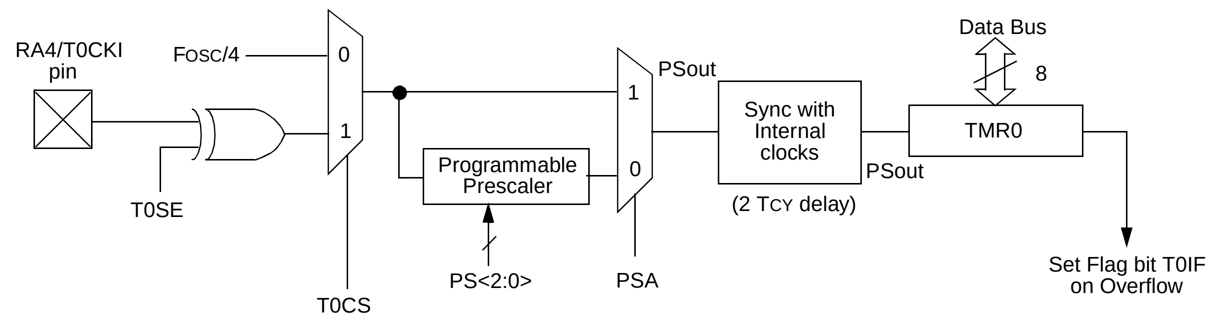
Counter mode is selected by setting the T0CS bit. In this mode Timer0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the source edge (T0SE) control bit (OPTION<4>). Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 7.2.

The prescaler is shared between the Timer0 module and the Watchdog Timer. The prescaler assignment is controlled in software by the control bit PSA (OPTION<3>). Clearing the PSA bit will assign the prescaler to Timer0. The prescaler is not readable or writable. When the prescaler is assigned to the Timer0 module, prescale value of 1:2, 1:4, ..., 1:256 are selectable. Section 7.3 details the operation of the prescaler.

7.1 Timer0 Interrupt

Timer0 interrupt is generated when the TMR0 register timer/counter overflows from FFh to 00h. This overflow sets the T0IF bit. The interrupt can be masked by clearing the T0IE bit (INTCON<5>). The T0IF bit (INTCON<2>) must be cleared in software by the Timer0 module interrupt service routine before re-enabling this interrupt. The Timer0 interrupt cannot wake the processor from SLEEP since the timer is shut off during SLEEP. See Figure 7-4 for Timer0 interrupt timing.

FIGURE 7-1: TIMER0 BLOCK DIAGRAM


RA4/T0CKI
pin
FOSC/4
0
1
T0SE
T0CS
Programmable
Prescaler
PS<2:0>
1
0
PSA
PSout
Sync with
Internal
clocks
(2 TCY delay)
PSout
Data Bus
8
TMR0
Set Flag bit T0IF
on Overflow

Note 1: Bits T0SE, T0CS, PS2, PS1, PS0 and PSA are located in the OPTION register.

2: The prescaler is shared with Watchdog Timer (Figure 7-6)

FIGURE 7-2: TIMER0 (TMR0) TIMING: INTERNAL CLOCK/NO PRESCALER

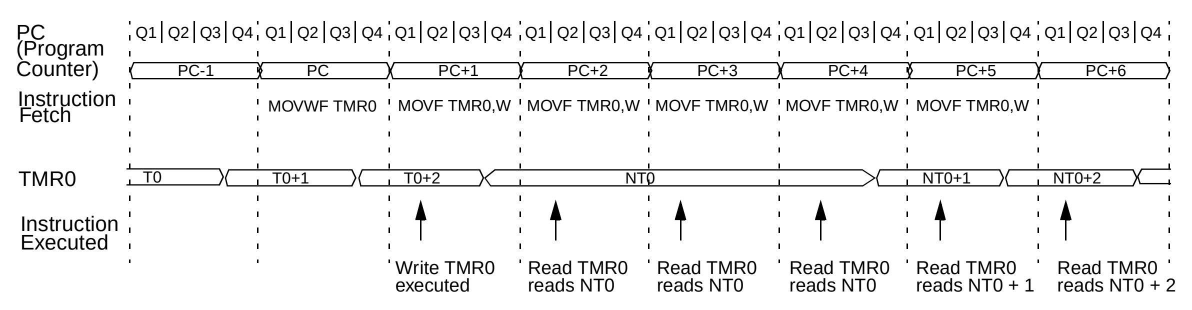

PC
(Program
Counter)
Q1 Q2 Q3 Q4
PC-1
PC
PC+1
PC+2
PC+3
PC+4
PC+5
PC+6
Instruction
Fetch
MOVWF TMR0
MOVF TMR0,W
MOVF TMR0,W
MOVF TMR0,W
MOVF TMR0,W
MOVF TMR0,W
TMR0
T0
T0+1
T0+2
NT0
NT0+1
NT0+2
Instruction
Executed
Write TMR0
executed
Read TMR0
reads NT0
Read TMR0
reads NT0
Read TMR0
reads NT0
Read TMR0
reads NT0 + 1
Read TMR0
reads NT0 + 2

FIGURE 7-3: TIMER0 TIMING: INTERNAL CLOCK/PRESCALE 1:2

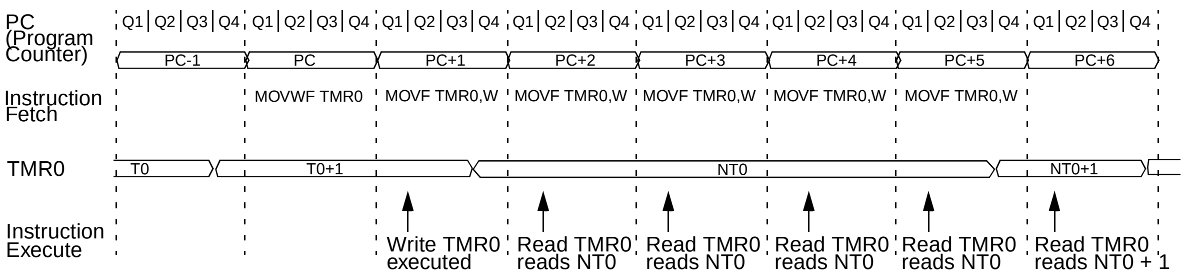
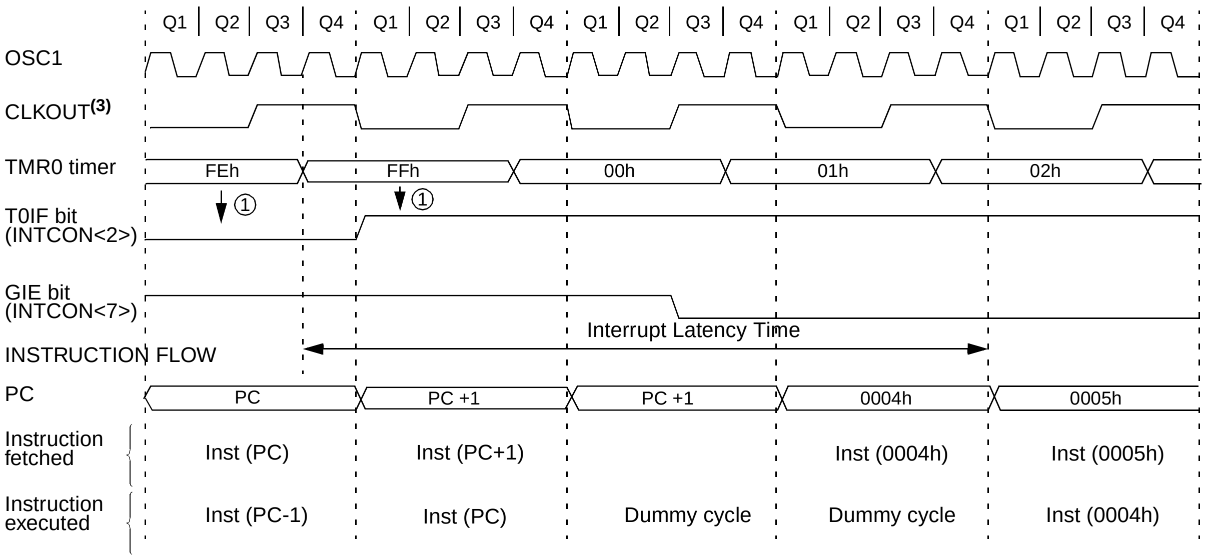

PC (Program Counter)
Q1 Q2 Q3 Q4
PC-1
PC
PC+1
PC+2
PC+3
PC+4
PC+5
PC+6
Instruction Fetch
MOVWF TMR0
MOVF TMR0,W
MOVF TMR0,W
MOVF TMR0,W
MOVF TMR0,W
MOVF TMR0,W
TMR0
T0
T0+1
NT0
NT0+1
Instruction Execute
Write TMR0 executed
Read TMR0 reads NT0
Read TMR0 reads NT0
Read TMR0 reads NT0
Read TMR0 reads NT0
Read TMR0 reads NT0 + 1

FIGURE 7-4: TIMER0 INTERRUPT TIMING


Q1 Q2 Q3 Q4
OSC1
CLKOUT(3)
TMR0 timer
FEh
FFh
00h
01h
02h
T0IF bit (INTCON<2>)
①
①
GIE bit (INTCON<7>)
Interrupt Latency Time
INSTRUCTION FLOW
PC
PC
PC +1
PC +1
0004h
0005h
Instruction fetched
Inst (PC)
Inst (PC+1)
Inst (0004h)
Inst (0005h)
Instruction executed
Inst (PC-1)
Inst (PC)
Dummy cycle
Dummy cycle
Inst (0004h)

Note 1: T0IF interrupt flag is sampled here (every Q1).
2: Interrupt latency = 3TCY, where TCY = instruction cycle time.
3: CLKOUT is available only in RC oscillator mode.

7.2 Using Timer0 with External Clock

When an external clock input is used for Timer0, it must meet certain requirements. The external clock requirement is due to internal phase clock (TOSC) synchronization. Also, there is a delay in the actual incrementing of Timer0 after synchronization.

7.2.1 EXTERNAL CLOCK SYNCHRONIZATION

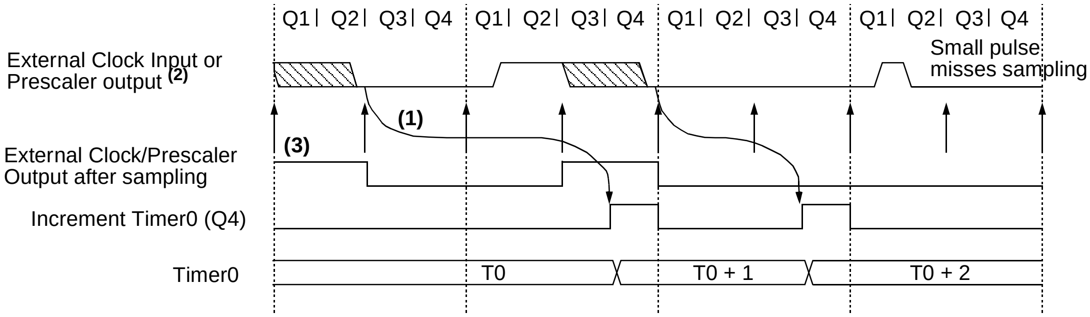
When no prescaler is used, the external clock input is the same as the prescaler output. The synchronization of T0CKI with the internal phase clocks is accomplished by sampling the prescaler output on the Q2 and Q4 cycles of the internal phase clocks (Figure 7-5). Therefore, it is necessary for T0CKI to be high for at least 2TOSC (and a small RC delay of 20 ns) and low for at least 2TOSC (and a small RC delay of 20 ns). Refer to the electrical specification of the desired device.

When a prescaler is used, the external clock input is divided by the asynchronous ripple-counter type prescaler so that the prescaler output is symmetrical. For the external clock to meet the sampling requirement, the ripple-counter must be taken into account. Therefore, it is necessary for T0CKI to have a period of at least 4TOSC (and a small RC delay of 40 ns) divided by the prescaler value. The only requirement on T0CKI high and low time is that they do not violate the minimum pulse width requirement of 10 ns. Refer to parameters 40, 41 and 42 in the electrical specification of the desired device.

7.2.2 TIMER0 INCREMENT DELAY

Since the prescaler output is synchronized with the internal clocks, there is a small delay from the time the external clock edge occurs to the time the TMR0 is actually incremented. Figure 7-5 shows the delay from the external clock edge to the timer incrementing.

FIGURE 7-5: TIMER0 TIMING WITH EXTERNAL CLOCK


Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4
External Clock Input or Prescaler output (2)
Small pulse misses sampling
(1)
(3)
External Clock/Prescaler Output after sampling
Increment Timer0 (Q4)
Timer0
T0
T0 + 1
T0 + 2

Note 1: Delay from clock input change to Timer0 increment is 3TOSC to 7TOSC. (Duration of Q = TOSC). Therefore, the error in measuring the interval between two edges on Timer0 input = ±4TOSC max.

2: External clock if no prescaler selected; prescaler output otherwise.

3: The arrows indicate the points in time where sampling occurs.

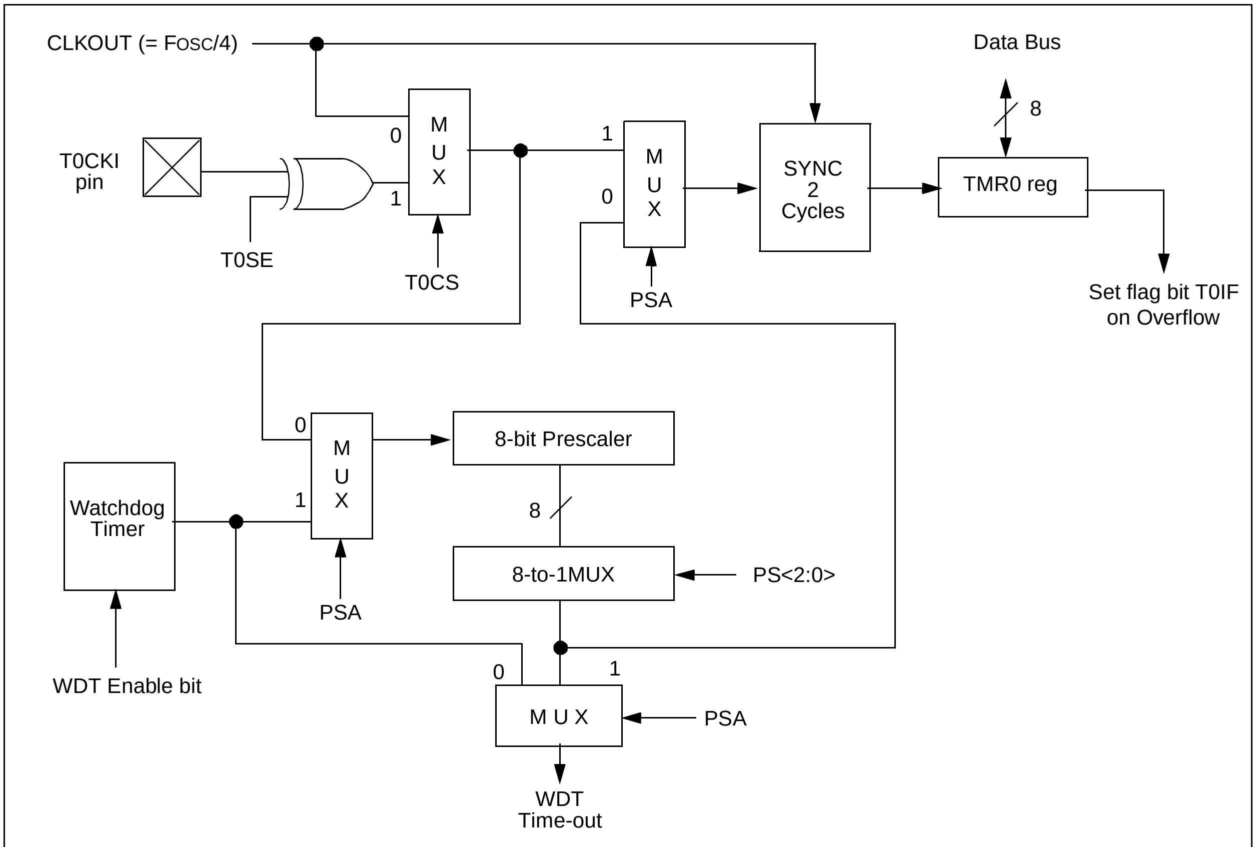
7.3 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module, or as a postscaler for the Watchdog Timer, respectively (Figure 7-6). For simplicity, this counter is being referred to as “prescaler” throughout this data sheet. Note that there is only one prescaler available which is mutually exclusive between the Timer0 module and the Watchdog Timer. Thus, a prescaler assignment for the Timer0 module means that there is no prescaler for the Watchdog Timer and vice-versa.

The PSA and PS<2:0> bits (OPTION<3:0>) determine the prescaler assignment and prescale ratio.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (i.e., `CLRF 1, MOVWF 1, BSF 1,x`....etc.) will clear the prescaler. When assigned to WDT, a `CLRWDT` instruction will clear the prescaler along with the Watchdog Timer. The prescaler is not readable or writable.

FIGURE 7-6: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER


CLKOUT (= Fosc/4)
Data Bus
8
T0CKI pin
0
M U X
1
T0SE
T0CS
1
M U X
0
PSA
SYNC 2 Cycles
TMR0 reg
Set flag bit T0IF on Overflow
0
M U X
1
8-bit Prescaler
8
8-to-1MUX
PS<2:0>
Watchdog Timer
PSA
WDT Enable bit
0
1
M U X
PSA
WDT Time-out

Note: T0SE, T0CS, PSA, PS<2:0> are bits in the OPTION register.

7.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed “on-the-fly” during program execution). To avoid an unintended device RESET, the following instruction sequence (Example 7-1) must be executed when changing the prescaler assignment from Timer0 to WDT.

EXAMPLE 7-1: CHANGING PRESCALER (TIMER0→WDT)

```
 1.BCF    STATUS, RP0  ;Skip if already in
                       ; Bank 0
 2.CLRWDT              ;Clear WDT
 3.CLRF   TMR0         ;Clear TMR0 & Prescaler
 4.BSF    STATUS, RP0  ;Bank 1
 5.MOVLW  '00101111'b  ;These 3 lines (5, 6, 7)
 6.MOVWF  OPTION       ; are required only if
                       ; desired PS<2:0> are
 7.CLRWDT              ; 000 or 001
 8.MOVLW  '00101xxx'b  ;Set Postscaler to
 9.MOVWF  OPTION       ; desired WDT rate
10.BCF    STATUS, RP0  ;Return to Bank 0
```

To change prescaler from the WDT to the TMR0 module, use the sequence shown in Example 7-2. This precaution must be taken even if the WDT is disabled.

EXAMPLE 7-2: CHANGING PRESCALER (WDT→TIMER0)

```
CLRWDT                 ;Clear WDT and
                       ;prescaler
BSF      STATUS, RP0
MOVLW    b'xxxx0xxx'   ;Select TMR0, new
                       ;prescale value and
                       ;clock source
MOVWF    OPTION_REG
BCF      STATUS, RP0
```

TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR	Value on All Other Resets
01h	TMR0	Timer0 module register								`xxxx xxxx`	`uuuu uuuu`
0Bh/8Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	`0000 000x`	`0000 000u`
81h	OPTION	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	`1111 1111`	`1111 1111`
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	`---1 1111`	`---1 1111`

Legend: — = Unimplemented locations, read as ‘0’, x = unknown, u = unchanged.

Note: Shaded bits are not used by TMR0 module.

NOTES:

8.0 COMPARATOR MODULE

The comparator module contains two analog comparators. The inputs to the comparators are multiplexed with the RA0 through RA3 pins. The on-chip voltage reference (Section 9.0) can also be an input to the comparators.

The CMCON register, shown in Register 8-1, controls the comparator input and output multiplexers. A block diagram of the comparator is shown in Figure 8-1.

REGISTER 8-1: CMCON REGISTER (ADDRESS 1Fh)

R-0	R-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0
bit7							bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
- n = Value at POR reset

bit 7: **C2OUT**: Comparator 2 output
1 = C2 VIN+ > C2 VIN–
0 = C2 VIN+ < C2 VIN–

bit 6: **C1OUT**: Comparator 1 output
1 = C1 VIN+ > C1 VIN–
0 = C1 VIN+ < C1 VIN–

bit 5-4: **Unimplemented:** Read as '0'

bit 3: **CIS**: Comparator Input Switch
When CM<2:0>: = 001:
1 = C1 VIN– connects to RA3
0 = C1 VIN– connects to RA0
When CM<2:0> = 010:
1 = C1 VIN– connects to RA3
C2 VIN– connects to RA2
0 = C1 VIN– connects to RA0
C2 VIN– connects to RA1

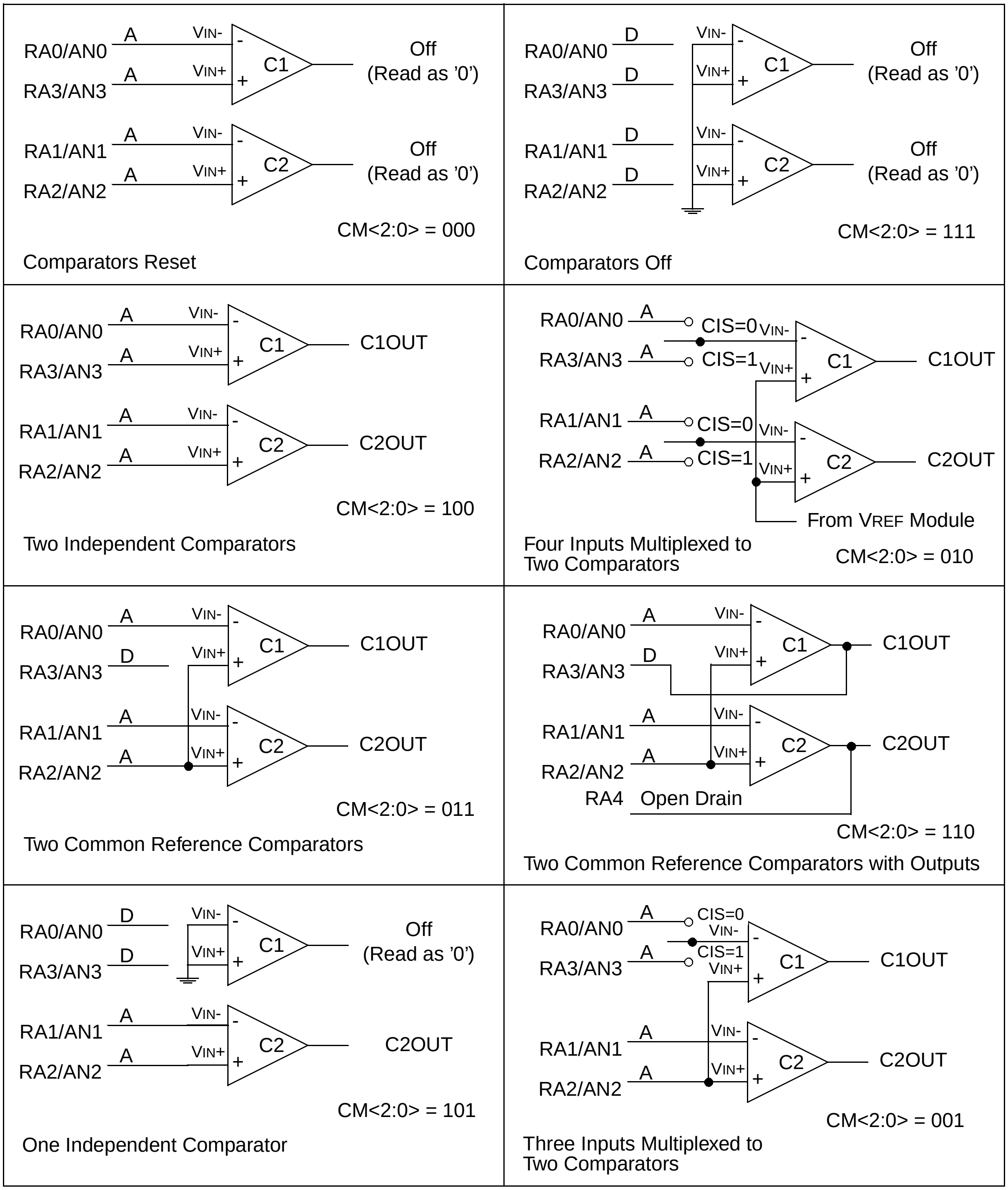
bit 2-0: **CM<2:0>**: Comparator mode
Figure 8-1.

8.1 Comparator Configuration

There are eight modes of operation for the comparators. The CMCON register is used to select the mode. Figure 8-1 shows the eight possible modes. The TRISA register controls the data direction of the comparator pins for each mode. If the comparator mode is changed, the comparator output level may not be valid for the specified mode change delay shown in Table 13-1.

Note:	Comparator interrupts should be disabled during a comparator mode change, otherwise a false interrupt may occur.

FIGURE 8-1: COMPARATOR I/O OPERATING MODES


RA0/AN0
A
VIN-
RA3/AN3
A
VIN+
C1
Off
(Read as '0')
RA1/AN1
A
VIN-
RA2/AN2
A
VIN+
C2
Off
(Read as '0')
CM<2:0> = 000
Comparators Reset
RA0/AN0
D
VIN-
RA3/AN3
D
VIN+
C1
Off
(Read as '0')
RA1/AN1
D
VIN-
RA2/AN2
D
VIN+
C2
Off
(Read as '0')
CM<2:0> = 111
Comparators Off
RA0/AN0
A
VIN-
RA3/AN3
A
VIN+
C1
C1OUT
RA1/AN1
A
VIN-
RA2/AN2
A
VIN+
C2
C2OUT
CM<2:0> = 100
Two Independent Comparators
RA0/AN0
A
CIS=0
VIN-
RA3/AN3
A
CIS=1
VIN+
C1
C1OUT
RA1/AN1
A
CIS=0
VIN-
RA2/AN2
A
CIS=1
VIN+
C2
C2OUT
From VREF Module
Four Inputs Multiplexed to Two Comparators
CM<2:0> = 010
RA0/AN0
A
VIN-
RA3/AN3
D
VIN+
C1
C1OUT
RA1/AN1
A
VIN-
RA2/AN2
A
VIN+
C2
C2OUT
CM<2:0> = 011
Two Common Reference Comparators
RA0/AN0
A
VIN-
RA3/AN3
D
VIN+
C1
C1OUT
RA1/AN1
A
VIN-
RA2/AN2
A
VIN+
C2
C2OUT
RA4
Open Drain
CM<2:0> = 110
Two Common Reference Comparators with Outputs
RA0/AN0
D
VIN-
RA3/AN3
D
VIN+
C1
Off
(Read as '0')
RA1/AN1
A
VIN-
RA2/AN2
A
VIN+
C2
C2OUT
CM<2:0> = 101
One Independent Comparator
RA0/AN0
A
CIS=0
VIN-
RA3/AN3
A
CIS=1
VIN+
C1
C1OUT
RA1/AN1
A
VIN-
RA2/AN2
A
VIN+
C2
C2OUT
CM<2:0> = 001
Three Inputs Multiplexed to Two Comparators

A = Analog Input, Port Reads Zeros Always
D = Digital Input
CIS = CMCON<3>, Comparator Input Switch

The code example in Example 8-1 depicts the steps required to configure the comparator module. RA3 and RA4 are configured as digital output. RA0 and RA1 are configured as the V- inputs and RA2 as the V+ input to both comparators.

EXAMPLE 8-1: INITIALIZING COMPARATOR MODULE

```
FLAG_REG EQU         0X20
CLRF     FLAG_REG    ;Init flag register
CLRF     PORTA       ;Init PORTA
MOVF     CMCON,W     ;Move comparator contents to W
ANDLW    0xC0        ;Mask comparator bits
IORWF    FLAG_REG,F  ;Store bits in flag register
MOVLW    0x03        ;Init comparator mode
MOVWF    CMCON       ;CM<2:0> = 011
BSF      STATUS,RP0  ;Select Bank1
MOVLW    0x07        ;Initialize data direction
MOVWF    TRISA       ;Set RA<2:0> as inputs
                     ;RA<4:3> as outputs
                     ;TRISA<7:5> always read ‘0’
BCF      STATUS,RP0  ;Select Bank 0
CALL     DELAY 10    ;10µs delay
MOVF     CMCON,F     ;Read CMCON to end change condition
BCF      PIR1,CMIF   ;Clear pending interrupts
BSF      STATUS,RP0  ;Select Bank 1
BSF      PIE1,CMIE   ;Enable comparator interrupts
BCF      STATUS,RP0  ;Select Bank 0
BSF      INTCON,PEIE ;Enable peripheral interrupts
BSF      INTCON,GIE  ;Global interrupt enable
```

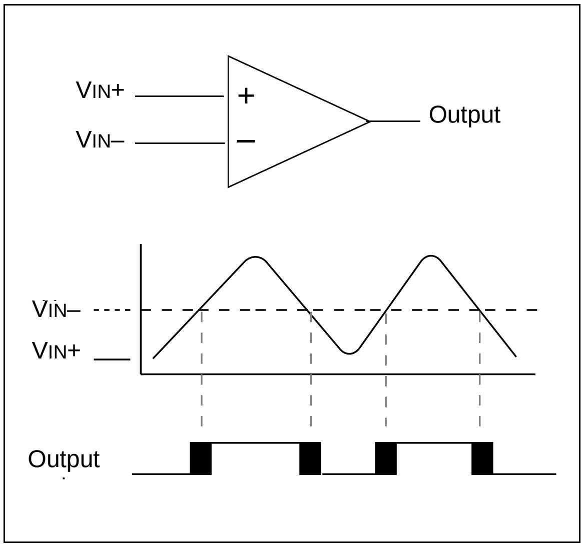
8.2 Comparator Operation

A single comparator is shown in Figure 8-2 along with the relationship between the analog input levels and the digital output. When the analog input at VIN+ is less than the analog input VIN–, the output of the comparator is a digital low level. When the analog input at VIN+ is greater than the analog input VIN–, the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 8-2 represent the uncertainty due to input offsets and response time.

8.3 Comparator Reference

An external or internal reference signal may be used depending on the comparator operating mode. The analog signal that is present at VIN– is compared to the signal at VIN+, and the digital output of the comparator is adjusted accordingly (Figure 8-2).

FIGURE 8-2: SINGLE COMPARATOR



8.3.1 EXTERNAL REFERENCE SIGNAL

When external voltage references are used, the comparator module can be configured to have the comparators operate from the same or different reference sources. However, threshold detector applications may require the same reference. The reference signal must be between VSS and VDD and can be applied to either pin of the comparator(s).

8.3.2 INTERNAL REFERENCE SIGNAL

The comparator module also allows the selection of an internally generated voltage reference for the comparators. Section 13, Instruction Sets, contains a detailed description of the Voltage Reference Module that provides this signal. The internal reference signal is used when the comparators are in mode CM<2:0>=010 (Figure 8-1). In this mode, the internal voltage reference is applied to the VIN+ pin of both comparators.

8.4 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output has a valid level. If the internal reference is changed, the maximum delay of the internal voltage reference must be considered when using the comparator outputs, otherwise the maximum delay of the comparators should be used (Table 13-1 **).**

8.5 Comparator Outputs

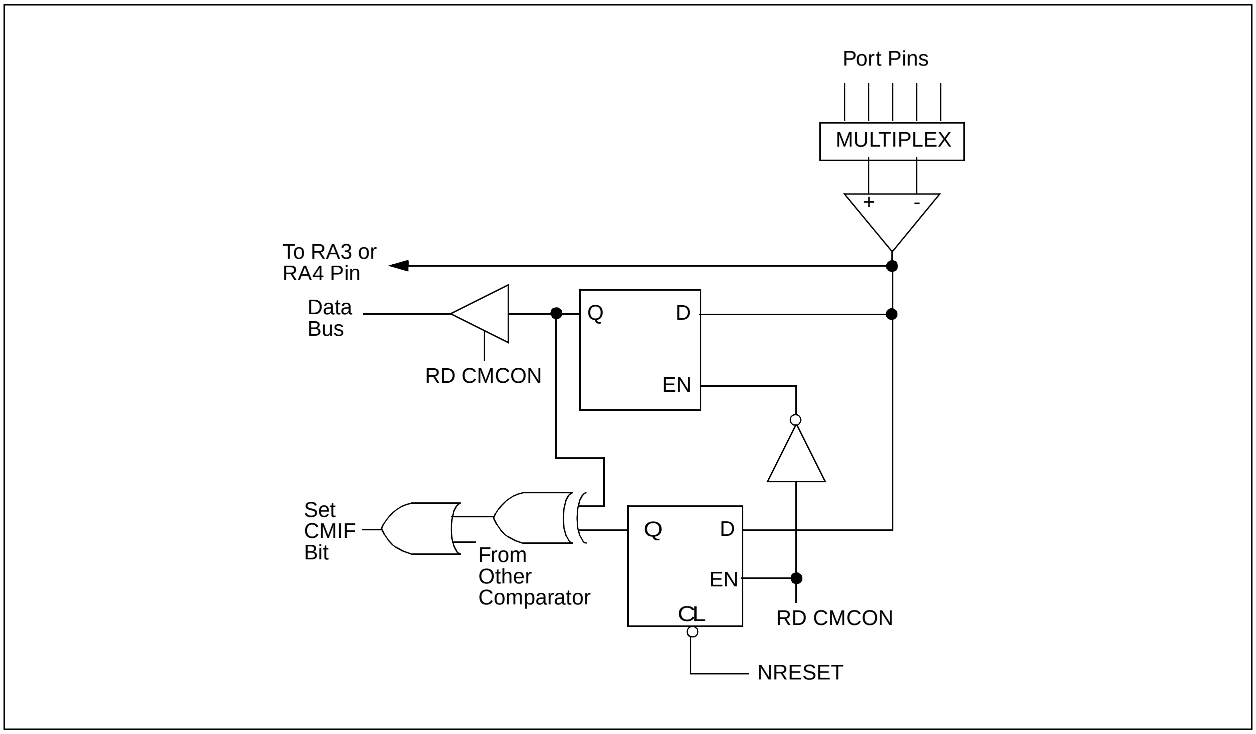
The comparator outputs are read through the CMCON register. These bits are read only. The comparator outputs may also be directly output to the RA3 and RA4 I/O pins. When the CM<2:0> = 110, multiplexors in the output path of the RA3 and RA4 pins will switch and the output of each pin will be the unsynchronized output of the comparator. The uncertainty of each of the comparators is related to the input offset voltage and the response time given in the specifications. Figure 8-3 shows the comparator output block diagram.

The TRISA bits will still function as an output enable/disable for the RA3 and RA4 pins while in this mode.

Note 1: When reading the PORT register, all pins configured as analog inputs will read as a ‘0’. Pins configured as digital inputs will convert an analog input according to the Schmitt Trigger input specification.

2: Analog levels on any pin that is defined as a digital input may cause the input buffer to consume more current than is specified.

FIGURE 8-3: COMPARATOR OUTPUT BLOCK DIAGRAM



8.6 Comparator Interrupts

The comparator interrupt flag is set whenever there is a change in the output value of either comparator. Software will need to maintain information about the status of the output bits, as read from CMCON<7:6>, to determine the actual change that has occurred. The CMIF bit, PIR1<6>, is the comparator interrupt flag. The CMIF bit must be reset by clearing ‘0’. Since it is also possible to write a '1' to this register, a simulated interrupt may be initiated.

The CMIE bit (PIE1<6>) and the PEIE bit (INTCON<6>) must be set to enable the interrupt. In addition, the GIE bit must also be set. If any of these bits are clear, the interrupt is not enabled, though the CMIF bit will still be set if an interrupt condition occurs.

Note: If a change in the CMCON register (C1OUT or C2OUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CMIF (PIR1<6>) interrupt flag may not get set.

The user, in the interrupt service routine, can clear the interrupt in the following manner:

a) Any read or write of CMCON. This will end the mismatch condition.

b) Clear flag bit CMIF.

A mismatch condition will continue to set flag bit CMIF. Reading CMCON will end the mismatch condition, and allow flag bit CMIF to be cleared.

8.7 Comparator Operation During SLEEP

When a comparator is active and the device is placed in SLEEP mode, the comparator remains active and the interrupt is functional if enabled. This interrupt will wake-up the device from SLEEP mode when enabled. While the comparator is powered-up, higher sleep currents than shown in the power down current specification will occur. Each comparator that is operational will consume additional current as shown in the comparator specifications. To minimize power consumption while in SLEEP mode, turn off the comparators, CM<2:0> = 111, before entering sleep. If the device wakes-up from sleep, the contents of the CMCON register are not affected.

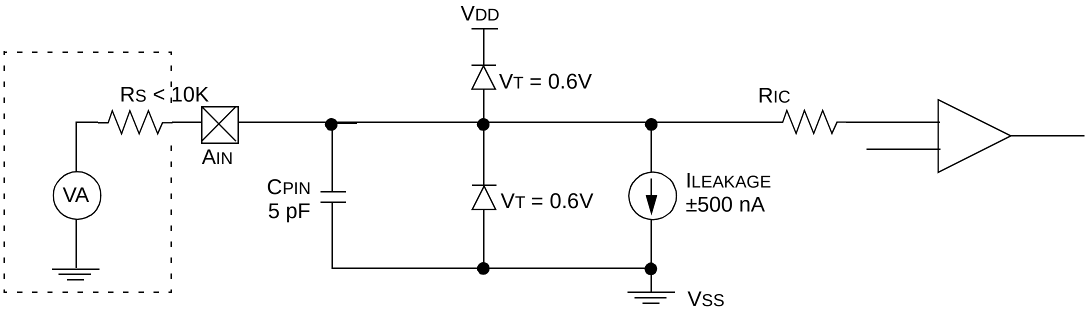
8.8 Effects of a RESET

A device reset forces the CMCON register to its reset state. This forces the comparator module to be in the comparator reset mode, CM<2:0> = 000. This ensures that all potential inputs are analog inputs. Device current is minimized when analog inputs are present at reset time. The comparators will be powered-down during the reset interval.

8.9 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 8-4. Since the analog pins are connected to a digital output, they have reverse biased diodes to VDD and VSS. The analog input therefore, must be between VSS and VDD. If the input voltage deviates from this range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up may occur. A maximum source impedance of 10 kΩ is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.

FIGURE 8-4: ANALOG INPUT MODEL



Legend
CPIN = Input capacitance
VT = Threshold voltage
ILEAKAGE = Leakage current at the pin due to various junctions
RIC = Interconnect resistance
RS = Source impedance
VA = Analog voltage

TABLE 8-1: REGISTERS ASSOCIATED WITH COMPARATOR MODULE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR	Value on All Other Resets
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	`00-- 0000`	`00-- 0000`
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	`000- 0000`	`000- 0000`
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	`0000 000x`	`0000 000u`
0Ch	PIR1	—	CMIF	—	—	—	—	—	—	`-0-- ----`	`-0-- ----`
8Ch	PIE1	—	CMIE	—	—	—	—	—	—	`-0-- ----`	`-0-- ----`
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	`---1 1111`	`---1 1111`

Legend: - = Unimplemented, read as "0", x = Unknown, u = unchanged

9.0 VOLTAGE REFERENCE MODULE

The Voltage Reference is a 16-tap resistor ladder network that provides a selectable voltage reference. The resistor ladder is segmented to provide two ranges of VREF values and has a power-down function to conserve power when the reference is not being used. The VRCON register controls the operation of the reference as shown in Register 9-1. The block diagram is given in Figure 9-1.

9.1 Configuring the Voltage Reference

The Voltage Reference can output 16 distinct voltage levels for each range.

The equations used to calculate the output of the Voltage Reference are as follows:

if VRR = 1: VREF = (VR<3:0>/24) x VDD

if VRR = 0: VREF = (VDD x 1/4) + (VR<3:0>/32) x VDD

The setting time of the Voltage Reference must be considered when changing the VREF output (Table 13-1). Example 9-1 shows an example of how to configure the Voltage Reference for an output voltage of 1.25V with VDD = 5.0V.

REGISTER 9-1: VRCON REGISTER (ADDRESS 9Fh)

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
VREN	VROE	VRR	—	VR3	VR2	VR1	VR0

bit7 bit0

R = Readable bit
W = Writable bit
U = Unimplemented bit, read as ‘0’
- n = Value at POR reset

bit 7: **VREN:** VREF Enable
`1` = VREF circuit powered on
`0` = VREF circuit powered down, no IDD drain

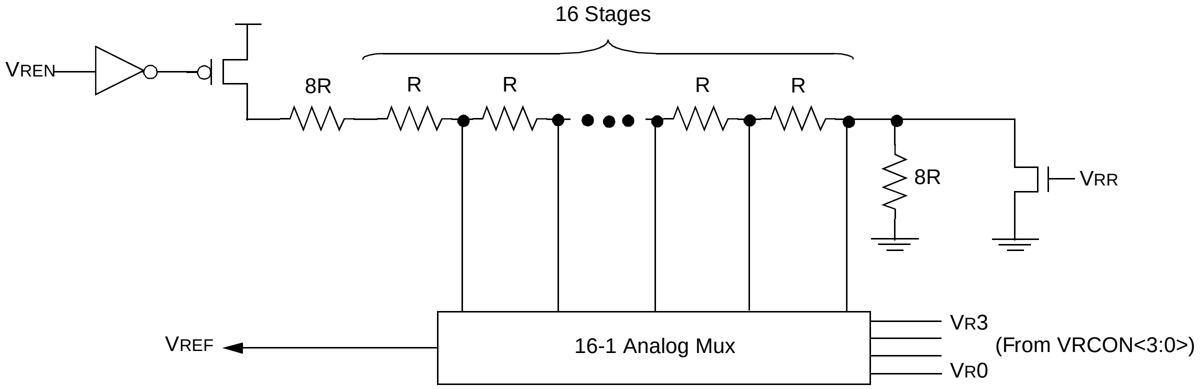
bit 6: **VROE:** VREF Output Enable
`1` = VREF is output on RA2 pin
`0` = VREF is disconnected from RA2 pin

bit 5: **VRR:** VREF Range selection
`1` = Low Range
`0` = High Range

bit 4: **Unimplemented:** Read as ’0’

bit 3-0: **VR<3:0>**: VREF value selection $0 \leq$ VR [3:0] ≤ 15
when VRR = 1: VREF = (VR<3:0>/ 24) * VDD
when VRR = 0: VREF = 1/4 * VDD + (VR<3:0>/ 32) * VDD

FIGURE 9-1: VOLTAGE REFERENCE BLOCK DIAGRAM



Note: R is defined in Table 13-2.

EXAMPLE 9-1: VOLTAGE REFERENCE CONFIGURATION

```
MOVLW     0x02          ; 4 Inputs Muxed
MOVWF     CMCON         ; to 2 comps.
BSF       STATUS,RP0    ; go to Bank 1
MOVLW     0x07          ; RA3-RA0 are
MOVWF     TRISA         ; outputs
MOVLW     0xA6          ; enable VREF
MOVWF     VRCON         ; low range
                        ; set VR<3:0>=6
BCF       STATUS,RP0    ; go to Bank 0
CALL      DELAY10       ; 10µs delay
```

9.2 Voltage Reference Accuracy/Error

The full range of VSS to VDD cannot be realized due to the construction of the module. The transistors on the top and bottom of the resistor ladder network (Figure 9-1) keep VREF from approaching VSS or VDD. The Voltage Reference is VDD derived and therefore, the VREF output changes with fluctuations in VDD. The absolute accuracy of the Voltage Reference can be found in Table 13-2.

9.3 Operation During Sleep

When the device wakes up from sleep through an interrupt or a Watchdog Timer time-out, the contents of the VRCON register are not affected. To minimize current consumption in SLEEP mode, the Voltage Reference should be disabled.

9.4 Effects of a Reset

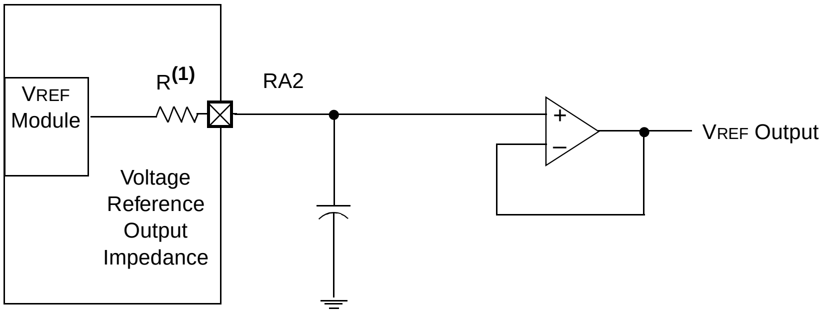
A device reset disables the Voltage Reference by clearing bit VREN (VRCON<7>). This reset also disconnects the reference from the RA2 pin by clearing bit VROE (VRCON<6>) and selects the high voltage range by clearing bit VRR (VRCON<5>). The VREF value select bits, VRCON<3:0>, are also cleared.

9.5 Connection Considerations

The Voltage Reference Module operates independently of the comparator module. The output of the reference generator may be connected to the RA2 pin if the TRISA<2> bit is set and the VROE bit, VRCON<6>, is set. Enabling the Voltage Reference output onto the RA2 pin with an input signal present will increase current consumption. Connecting RA2 as a digital output with VREF enabled will also increase current consumption.

The RA2 pin can be used as a simple D/A output with limited drive capability. Due to the limited drive capability, a buffer must be used in conjunction with the Voltage Reference output for external connections to VREF. Figure 9-2 shows an example buffering technique.

FIGURE 9-2: VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE



Note 1: R is dependent upon the Voltage Reference Configuration VRCON<3:0> and VRCON<5>.

TABLE 9-1: REGISTERS ASSOCIATED WITH VOLTAGE REFERENCE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value On POR / BOD	Value On All Other Resets
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	`000- 0000`	`000- 0000`
1Fh	CMCON	C2OUT	C1OUT	—	—	CIS	CM2	CM1	CM0	`00-- 0000`	`00-- 0000`
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	`---1 1111`	`---1 1111`

Legend: - = Unimplemented, read as "0"

10.0 SPECIAL FEATURES OF THE CPU

Special circuits to deal with the needs of real time applications are what sets a microcontroller apart from other processors. The PIC16CE62X family has a host of such features intended to maximize system reliability, minimize cost through elimination of external components, provide power saving operating modes and offer code protection.

These are:

1. OSC selection
2. Reset
   Power-on Reset (POR)
   Power-up Timer (PWRT)
   Oscillator Start-Up Timer (OST)
   Brown-out Reset (BOD)
3. Interrupts
4. Watchdog Timer (WDT)
5. SLEEP
6. Code protection
7. ID Locations
8. In-circuit serial programming

The PIC16CE62X has a Watchdog Timer which is controlled by configuration bits. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT), which provides a fixed delay of 72 ms (nominal) on power-up only, and is designed to keep the part in reset while the power supply stabilizes. There is also circuitry to reset the device if a brown-out occurs, which provides at least a 72 ms reset. With these three functions on-chip, most applications need no external reset circuitry.

The SLEEP mode is designed to offer a very low current power-down mode. The user can wake-up from SLEEP through external reset, Watchdog Timer wake-up or through an interrupt. Several oscillator options are also made available to allow the part to fit the application. The RC oscillator option saves system cost, while the LP crystal option saves power. A set of configuration bits are used to select various options.

10.1 Configuration Bits

The configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped in program memory location 2007h.

The user will note that address 2007h is beyond the user program memory space. In fact, it belongs to the special test/configuration memory space (2000h – 3FFFh), which can be accessed only during programming.

REGISTER 10-1: CONFIGURATION WORD

CP1	CP0(2)	CP1	CP0(2)	CP1	CP0(2)	—	BODEN(1)	CP1	CP0(2)	$\overline{\text{PWRTE}}$(1)	WDTE	F0SC1	F0SC0
bit13													bit0

CONFIG REGISTER: Address 2007h

bit 13-8, 5-4: **CP1:CP0 Pairs:** Code protection bit pairs(2)
Code protection for 2K program memory
11 = Program memory code protection off
10 = 0400h-07FFh code protected
01 = 0200h-07FFh code protected
00 = 0000h-07FFh code protected
Code protection for 1K program memory
11 = Program memory code protection off
10 =Program memory code protection on
01 = 0200h-03FFh code protected
00 = 0000h-03FFh code protected
Code protection for 0.5K program memory
11 = Program memory code protection off
10 = Program memory code protection off
01 = Program memory code protection off
00 = 0000h-01FFh code protected

bit 7: **Unimplemented**: Read as '1'

bit 6: **BODEN**: Brown-out Reset Enable bit (1)
1 = BOD enabled
0 = BOD disabled

bit 3: **$\overline{\text{PWRTE}}$**: Power-up Timer Enable bit (1)
1 = PWRT disabled
0 = PWRT enabled

bit 2: **WDTE**: Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 1-0: **FOSC1:FOSC0**: Oscillator Selection bits
11 = RC oscillator
10 = HS oscillator
01 = XT oscillator
00 = LP oscillator

Note 1: Enabling Brown-out Reset automatically enables Power-up Timer (PWRT), regardless of the value of bit $\overline{\text{PWRTE}}$. Ensure the Power-up Timer is enabled anytime Brown-out Reset is enabled.
2: All of the CP<1:0> pairs have to be given the same value to enable the code protection scheme listed.

10.2 Oscillator Configurations

10.2.1 OSCILLATOR TYPES

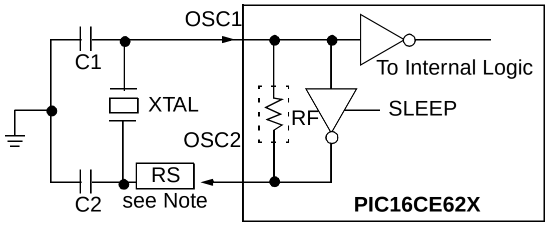
The PIC16CE62X can be operated in four different oscillator options. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

10.2.2 CRYSTAL OSCILLATOR / CERAMIC RESONATORS

In XT, LP or HS modes, a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation (Figure 10-1). The PIC16CE62X oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1 pin (Figure 10-2).

FIGURE 10-1: CRYSTAL OPERATION (OR CERAMIC RESONATOR) (HS, XT OR LP OSC CONFIGURATION)



See Table 10-1 and Table 10-2 for recommended values of C1 and C2.

Note: A series resistor may be required for AT strip cut crystals.

FIGURE 10-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

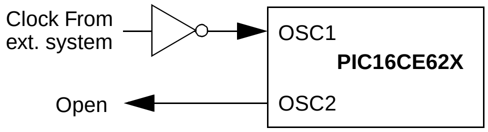


TABLE 10-1: CERAMIC RESONATORS, PIC16CE62X

Ranges Tested:			
Mode	**Freq**	**OSC1**	**OSC2**
XT	455 kHz 2.0 MHz 4.0 MHz	68 - 100 pF 15 - 68 pF 15 - 68 pF	68 - 100 pF 15 - 68 pF 15 - 68 pF
HS	8.0 MHz 16.0 MHz	10 - 68 pF 10 - 22 pF	10 - 68 pF 10 - 22 pF
These values are for design guidance only. See notes at bottom of page.			

TABLE 10-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR, PIC16CE62X

Osc Type	Crystal Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF
These values are for design guidance only. See notes at bottom of page.			

1. Recommended values of C1 and C2 are identical to the ranges tested table.
2. Higher capacitance increases the stability of oscillator, but also increases the start-up time.
3. Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.
4. Rs may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification.

10.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a prepackaged oscillator can be used or a simple oscillator circuit with TTL gates can be built. Prepackaged oscillators provide a wide operating range and better stability. A well-designed crystal oscillator will provide good performance with TTL gates. Two types of crystal oscillator circuits can be used; one with series resonance or one with parallel resonance.

Figure 10-3 shows implementation of a parallel resonant oscillator circuit. The circuit is designed to use the fundamental frequency of the crystal. The 74AS04 inverter performs the 180° phase shift that a parallel oscillator requires. The 4.7 kΩ resistor provides the negative feedback for stability. The 10 kΩ potentiometers bias the 74AS04 in the linear region. This could be used for external oscillator designs.

FIGURE 10-3: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

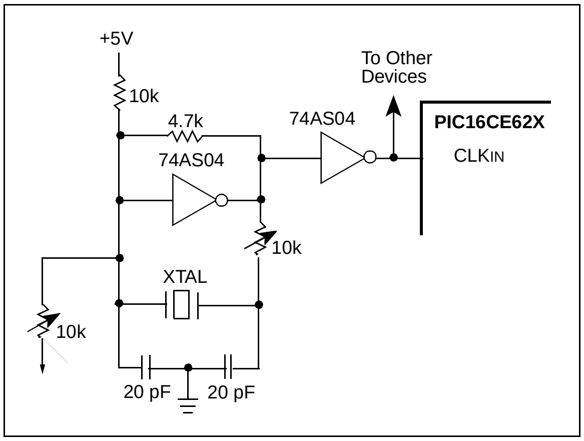
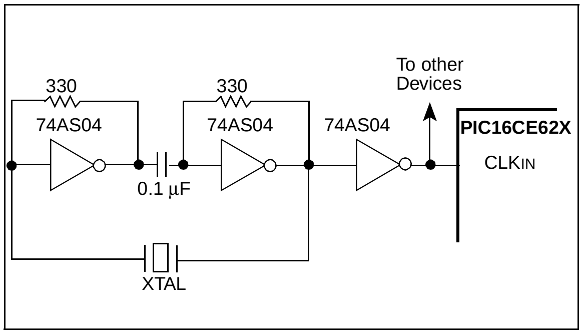


Figure 10-4 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180° phase shift in a series resonant oscillator circuit. The 330 kΩ resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 10-4: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



10.2.4 RC OSCILLATOR

For timing insensitive applications the “RC” device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (Rext) and capacitor (Cext) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low Cext values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 10-5 shows how the R/C combination is connected to the PIC16CE62X. For Rext values below 2.2 kΩ, the oscillator operation may become unstable, or stop completely. For very high Rext values (i.e., 1 MΩ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend to keep Rext between 3 kΩ and 100 kΩ.

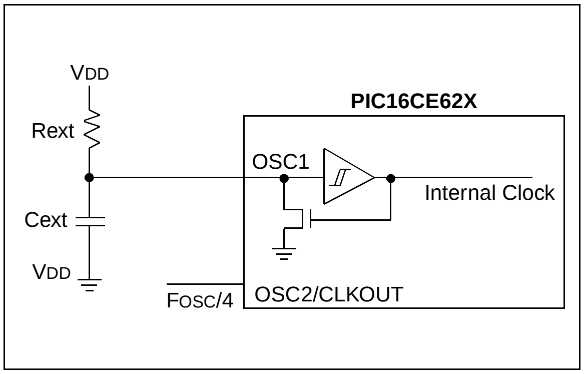
Although the oscillator will operate with no external capacitor (Cext = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With no or small external capacitance, the oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance or package lead frame capacitance.

See Section 14.0 for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See Section 14.0 for variation of oscillator frequency due to VDD for given Rext/Cext values, as well as frequency variation due to operating temperature for given R, C, and VDD values.

The oscillator frequency, divided by 4, is available on the OSC2/CLKOUT pin and can be used for test purposes or to synchronize other logic (Figure 3-2 for waveform).

FIGURE 10-5: RC OSCILLATOR MODE



10.3 Reset

The PIC16CE62X differentiates between various kinds of reset:

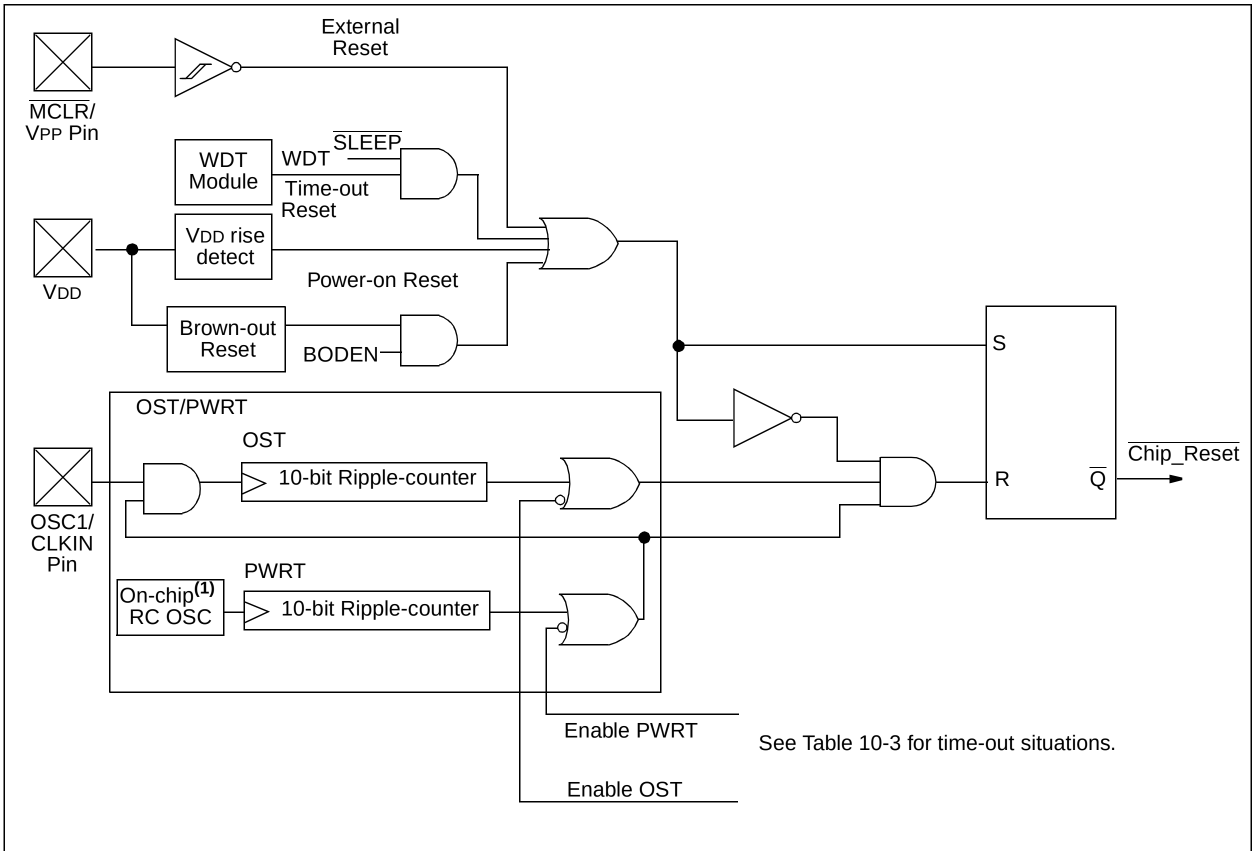
a) Power-on reset (POR)
b) $\overline{\text{MCLR}}$ reset during normal operation
c) $\overline{\text{MCLR}}$ reset during SLEEP
d) WDT reset (normal operation)
e) WDT wake-up (SLEEP)
f) Brown-out Reset (BOD)

Some registers are not affected in any reset condition. Their status is unknown on POR and unchanged in any other reset. Most other registers are reset to a “reset state” on Power-on reset, $\overline{\text{MCLR}}$ reset, WDT reset and $\overline{\text{MCLR}}$ reset during SLEEP. They are not affected by a WDT wake-up, since this is viewed as the resumption of normal operation. $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different reset situations as indicated in Table 10-4. These bits are used in software to determine the nature of the reset. See Table 10-6 for a full description of reset states of all registers.

A simplified block diagram of the on-chip reset circuit is shown in Figure 10-6.

The $\overline{\text{MCLR}}$ reset path has a noise filter to detect and ignore small pulses. See Table 13-5 for pulse width specification.

FIGURE 10-6: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT


External Reset
MCLR/
VPP Pin
WDT Module
WDT
SLEEP
Time-out Reset
VDD rise detect
VDD
Power-on Reset
Brown-out Reset
BODEN
OST/PWRT
OST
10-bit Ripple-counter
OSC1/
CLKIN
Pin
PWRT
On-chip(1) RC OSC
10-bit Ripple-counter
S
R
Q
Chip_Reset
Enable PWRT
Enable OST

See Table 10-3 for time-out situations.

Note 1: This is a separate oscillator from the RC oscillator of the CLKIN pin.

10.4 Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST) and Brown-out Reset (BOD)

10.4.1 POWER-ON RESET (POR)

The on-chip POR circuit holds the chip in reset until VDD has reached a high enough level for proper operation. To take advantage of the POR, just tie the $\overline{\text{MCLR}}$ pin through a resistor to VDD. This will eliminate external RC components usually needed to create Power-on Reset. A maximum rise time for VDD is required. See electrical specifications for details.

The POR circuit does not produce an internal reset when VDD declines.

When the device starts normal operation (exits the reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in reset until the operating conditions are met.

For additional information, refer to Application Note AN607, "Power-up Trouble Shooting".

10.4.2 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 72 ms (nominal) time-out on power-up only, from POR or Brown-out Reset. The Power-up Timer operates on an internal RC oscillator. The chip is kept in reset as long as PWRT is active. The PWRT delay allows the VDD to rise to an acceptable level. A configuration bit, $\overline{\text{PWRTE}}$, can disable (if set) or enable (if cleared or programmed) the Power-up Timer. The Power-up Timer should always be enabled when Brown-out Reset is enabled.

The Power-Up Time delay will vary from chip-to-chip and due to VDD, temperature and process variation. See DC parameters for details.

10.4.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-Up Timer (OST) provides a 1024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over. This ensures that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP and HS modes and only on power-on reset or wake-up from SLEEP.

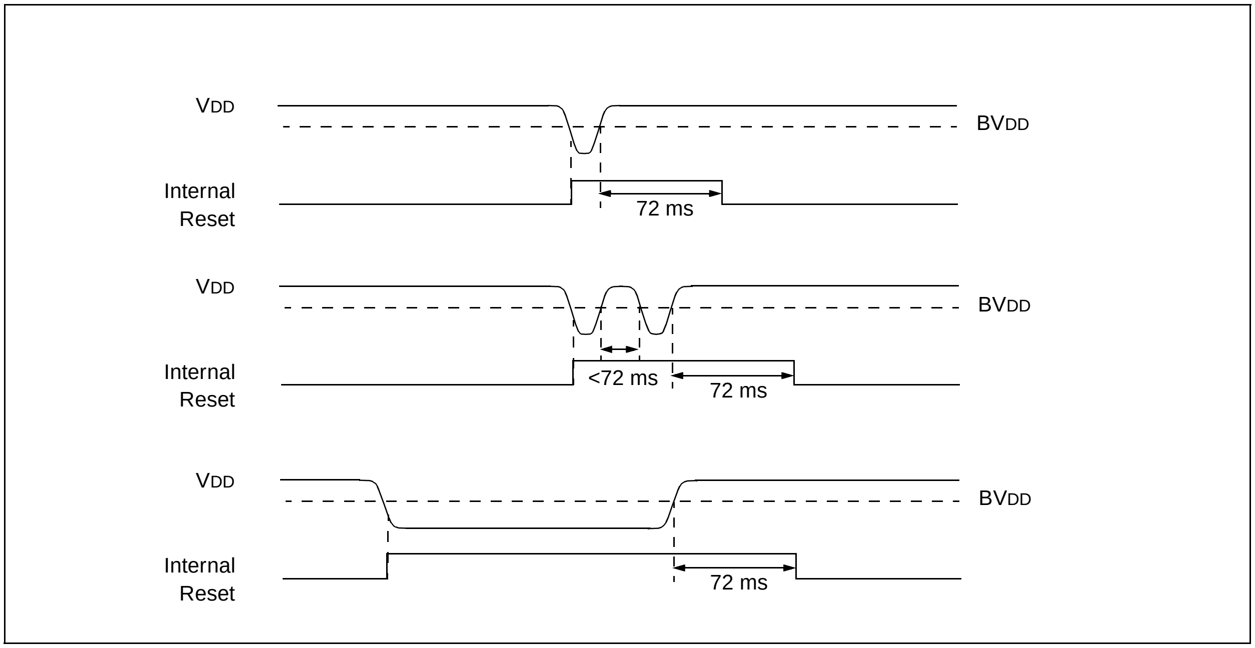
10.4.4 BROWN-OUT RESET (BOD)

The PIC16CE62X members have on-chip Brown-out Reset circuitry. A configuration bit, BOREN, can disable (if clear/programmed) or enable (if set) the Brown-out Reset circuitry. If VDD falls below 4.0V (refer to BVDD parameter D005) for greater than parameter (TBOR) in Table 13-5, the brown-out situation will reset the chip. A reset won't occur if VDD falls below 4.0V for less than parameter (TBOR).

On any reset (Power-on, Brown-out, Watch-dog, etc.) the chip will remain in reset until VDD rises above BVDD. The Power-up Timer will then be invoked and will keep the chip in reset an additional 72 ms.

If VDD drops below BVDD while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be re-initialized. Once VDD rises above BVDD, the Power-Up Timer will execute a 72 ms reset. The Power-up Timer should always be enabled when Brown-out Reset is enabled. Figure 10-7 shows typical Brown-out situations.

FIGURE 10-7: BROWN-OUT SITUATIONS



10.4.5 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows: First PWRT time-out is invoked after POR has expired, then OST is activated. The total time-out will vary based on oscillator configuration and $\overline{\text{PWRTE}}$ bit status. For example, in RC mode with $\overline{\text{PWRTE}}$ bit erased (PWRT disabled), there will be no time-out at all. Figure 10-8, Figure 10-9 and Figure 10-10 depict time-out sequences.

Since the time-outs occur from the POR pulse, if $\overline{\text{MCLR}}$ is kept low long enough, the time-outs will expire. Then bringing $\overline{\text{MCLR}}$ high will begin execution immediately (see Figure 10-9). This is useful for testing purposes or to synchronize more than one PICmicro® device operating in parallel.

Table 10-5 shows the reset conditions for some special registers, while Table 10-6 shows the reset conditions for all the registers.

10.4.6 POWER CONTROL (PCON)/STATUS REGISTER

The power control/status register, PCON (address 8Eh) has two bits.

Bit0 is $\overline{\text{BOR}}$ (Brown-out). $\overline{\text{BOR}}$ is unknown on power-on-reset. It must then be set by the user and checked on subsequent resets to see if $\overline{\text{BOR}}$ = 0 indicating that a brown-out has occurred. The $\overline{\text{BOR}}$ status bit is a don't care and is not necessarily predictable if the brown-out circuit is disabled (by setting BODEN bit = 0 in the Configuration word).

Bit1 is $\overline{\text{POR}}$ (Power-on-reset). It is a '0' on power-on-reset and unaffected otherwise. The user must write a '1' to this bit following a power-on-reset. On a subsequent reset, if $\overline{\text{POR}}$ is '0', it will indicate that a power-on-reset must have occurred (VDD may have gone too low).

TABLE 10-3: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up		Brown-out Reset	Wake-up from SLEEP
	$\overline{\text{PWRTE}}$ = 0	$\overline{\text{PWRTE}}$ = 1		
XT, HS, LP	72 ms + 1024 TOSC	1024 TOSC	72 ms + 1024 TOSC	1024 TOSC
RC	72 ms	—	72 ms	—

TABLE 10-4: STATUS/PCON BITS AND THEIR SIGNIFICANCE

$\overline{\text{POR}}$	$\overline{\text{BOR}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	
0	X	1	1	Power-on-reset
0	X	0	X	Illegal, $\overline{\text{TO}}$ is set on $\overline{\text{POR}}$
0	X	X	0	Illegal, $\overline{\text{PD}}$ is set on $\overline{\text{POR}}$
1	0	X	X	Brown-out Reset
1	1	0	u	WDT Reset
1	1	0	0	WDT Wake-up
1	1	u	u	$\overline{\text{MCLR}}$ reset during normal operation
1	1	1	0	$\overline{\text{MCLR}}$ reset during SLEEP

Legend: x = unknown, u = unchanged

TABLE 10-5: INITIALIZATION CONDITION FOR SPECIAL REGISTERS

Condition	Program Counter	STATUS Register	PCON Register
Power-on Reset	000h	`0001 1xxx`	`---- --0x`
$\overline{\text{MCLR}}$ reset during normal operation	000h	`000u uuuu`	`---- --uu`
$\overline{\text{MCLR}}$ reset during SLEEP	000h	`0001 0uuu`	`---- --uu`
WDT reset	000h	`0000 uuuu`	`---- --uu`
WDT Wake-up	PC + 1	`uuu0 0uuu`	`---- --uu`
Brown-out Reset	000h	`000x xuuu`	`---- --u0`
Interrupt Wake-up from SLEEP	PC + 1(1)	`uuu1 0uuu`	`---- --uu`

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as ‘0’.

Note 1: When the wake-up is due to an interrupt and global enable bit, GIE is set and the PC is loaded with the interrupt vector (0004h) after execution of PC+1.

TABLE 10-6: INITIALIZATION CONDITION FOR REGISTERS

Register	Address	Power-on Reset	• $\overline{\text{MCLR}}$ Reset during normal operation • $\overline{\text{MCLR}}$ Reset during SLEEP • WDT Reset • Brown-out Reset (1)	• Wake-up from SLEEP through interrupt • Wake-up from SLEEP through WDT time-out
W	-	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
INDF	00h	-	-	-
TMR0	01h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
PCL	02h	`0000 0000`	`0000 0000`	`PC + 1`(3)
STATUS	03h	`0001 1xxx`	`000q quuu`(4)	`uuuq quuu`(4)
FSR	04h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
PORTA	05h	`---x xxxx`	`---u uuuu`	`---u uuuu`
PORTB	06h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
CMCON	1Fh	`00-- 0000`	`00-- 0000`	`uu-- uuuu`
PCLATH	0Ah	`---0 0000`	`---0 0000`	`---u uuuu`
INTCON	0Bh	`0000 000x`	`0000 000u`	`uuuu uqqq`(2)
PIR1	0Ch	`-0-- ----`	`-0-- ----`	`-q-- ----`(2,5)
OPTION	81h	`1111 1111`	`1111 1111`	`uuuu uuuu`
TRISA	85h	`---1 1111`	`---1 1111`	`---u uuuu`
TRISB	86h	`1111 1111`	`1111 1111`	`uuuu uuuu`
PIE1	8Ch	`-0-- ----`	`-0-- ----`	`-u-- ----`
PCON	8Eh	`---- --0x`	`---- --uq`(1,6)	`---- --uu`
EEINTF	90h	`---- -111`	`---- -111`	`---- -111`
VRCON	9Fh	`000- 0000`	`000- 0000`	`uuu- uuuu`

Legend: u = unchanged, x = unknown, - = unimplemented bit, reads as ‘0’,q = value depends on condition.

Note 1: If VDD goes too low, Power-on Reset will be activated and registers will be affected differently.

2: One or more bits in INTCON, PIR1 and/or PIR2 will be affected (to cause wake-up).

3: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

4: See Table 10-5 for reset value for specific condition.

5: If wake-up was due to comparator input changing , then bit 6 = 1. All other interrupts generating a wake-up will cause bit 6 = u.

6: If reset was due to brown-out, then PCON bit 0 = 0. All other resets will cause bit 0 = u.

FIGURE 10-8: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD}): CASE 1

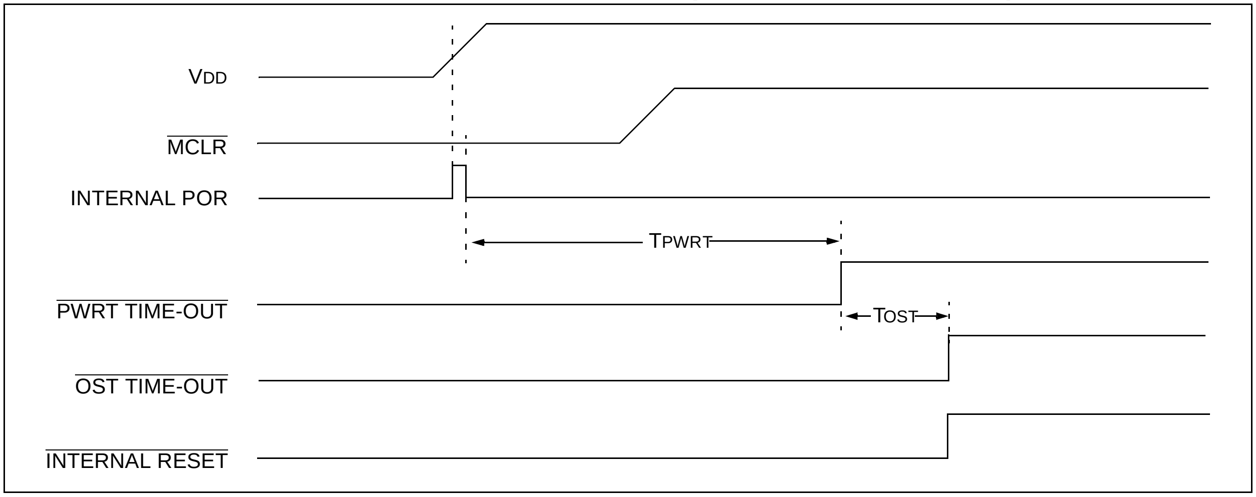

VDD
MCLR
INTERNAL POR
TPWRT
PWRT TIME-OUT
TOST
OST TIME-OUT
INTERNAL RESET

FIGURE 10-9: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD}): CASE 2

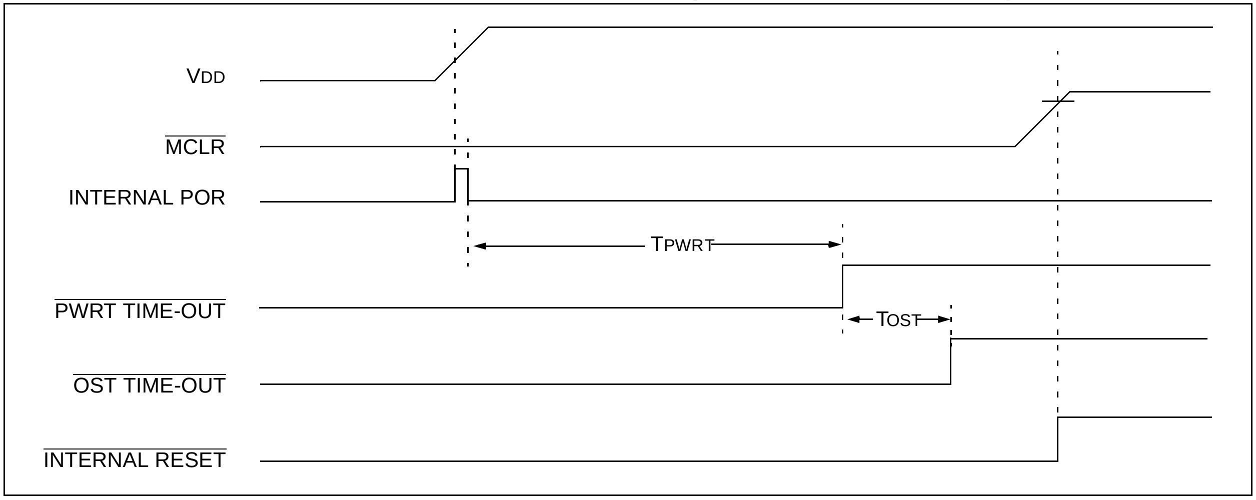

VDD
MCLR
INTERNAL POR
TPWRT
PWRT TIME-OUT
TOST
OST TIME-OUT
INTERNAL RESET

FIGURE 10-10: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ TIED TO V_{DD})

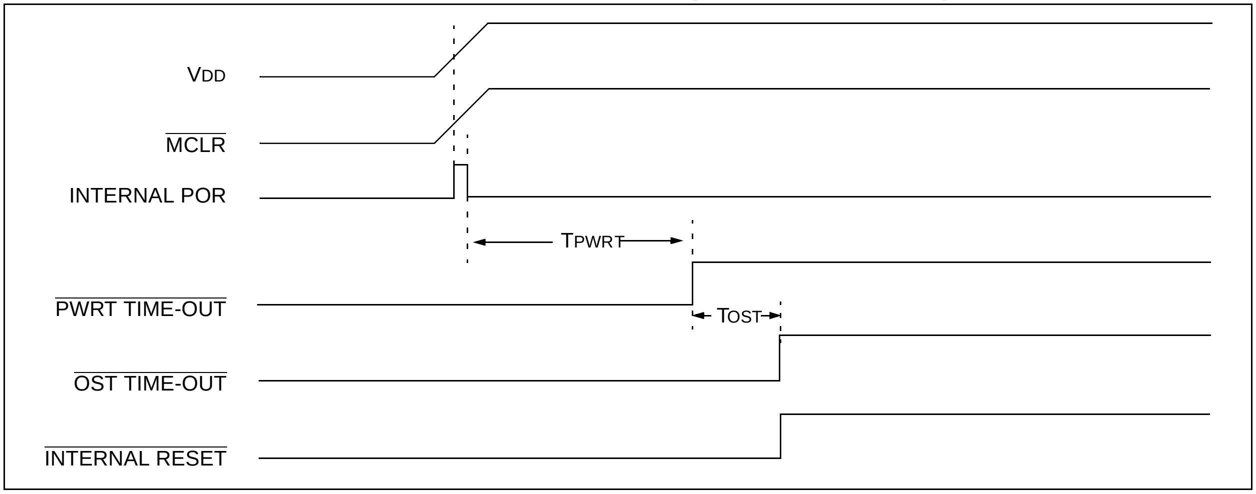
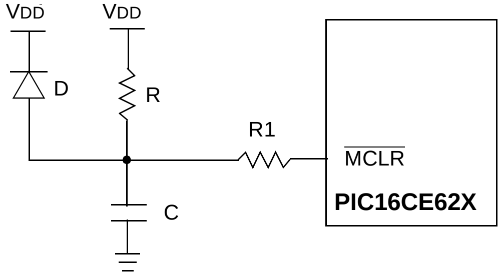

VDD
MCLR
INTERNAL POR
TPWRT
PWRT TIME-OUT
TOST
OST TIME-OUT
INTERNAL RESET

FIGURE 10-11: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)

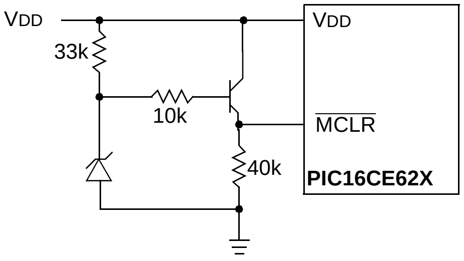


Note 1: External power-on reset circuit is required only if VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

2: < 40 kΩ is recommended to make sure that voltage drop across R does not violate the device's electrical specification.

3: R1 = 100Ω to 1 kΩ will limit any current flowing into $\overline{\text{MCLR}}$ from external capacitor C in the event of $\overline{\text{MCLR}}$/VPP pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

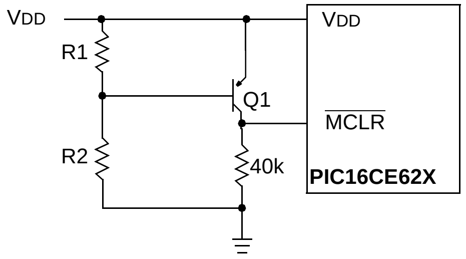
FIGURE 10-12: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 1



Note 1: This circuit will activate reset when VDD goes below (Vz + 0.7V) where Vz = Zener voltage.

2: Internal Brown-out Reset circuitry should be disabled when using this circuit.

FIGURE 10-13: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 2



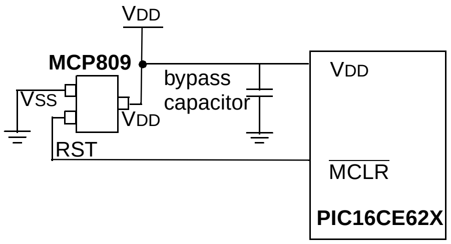
Note 1: This brown-out circuit is less expensive, albeit less accurate. Transistor Q1 turns off when VDD is below a certain level such that:

$$V_{DD} \times \frac{R1}{R1 + R2} = 0.7\ V$$

2: Internal brown-out detection should be disabled when using this circuit.

3: Resistors should be adjusted for the characteristics of the transistor.

FIGURE 10-14: EXTERNAL BROWN-OUT PROTECTION CIRCUIT 3



This brown-out protection circuit employs Microchip Technology's MCP809 microcontroller supervisor. The MCP8XX and MCP1XX families of supervisors provide push-pull and open collector outputs with both high and low active reset pins. There are 7 different trip point selections to accommodate 5V and 3V systems.

10.5 Interrupts

The PIC16CE62X has 4 sources of interrupt:

- External interrupt RB0/INT
- TMR0 overflow interrupt
- PortB change interrupts (pins RB<7:4>)
- Comparator interrupt

The interrupt control register (INTCON) records individual interrupt requests in flag bits. It also has individual and global interrupt enable bits.

A global interrupt enable bit, GIE (INTCON<7>) enables (if set) all un-masked interrupts or disables (if cleared) all interrupts. Individual interrupts can be disabled through their corresponding enable bits in INTCON register. GIE is cleared on reset.

The "return from interrupt" instruction, `RETFIE`, exits interrupt routine, as well as sets the GIE bit, which re-enable RB0/INT interrupts.

The INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

The peripheral interrupt flag is contained in the special register PIR1. The corresponding interrupt enable bit is contained in special registers PIE1.

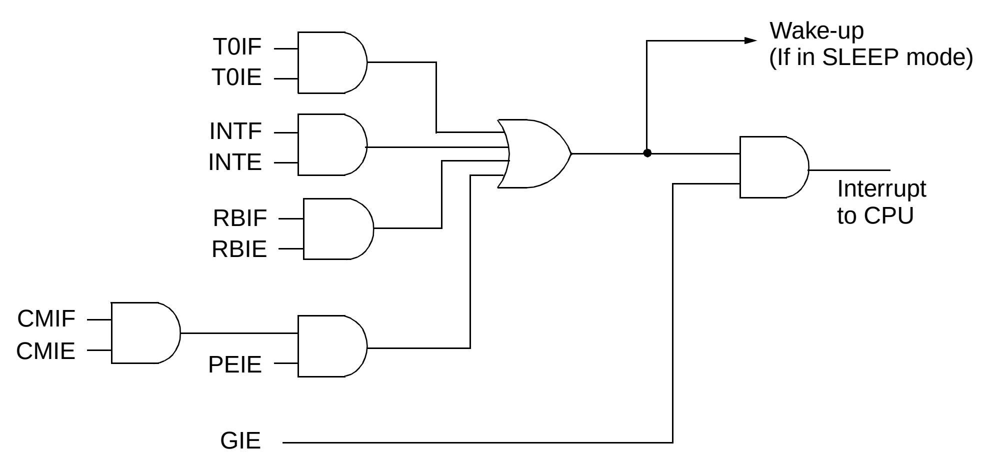
When an interrupt is responded to, the GIE is cleared to disable any further interrupt, the return address is pushed into the stack and the PC is loaded with 0004h. Once in the interrupt service routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid RB0/INT recursive interrupts.

For external interrupt events, such as the INT pin or PORTB change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends on when the interrupt event occurs (Figure 10-16). The latency is the same for one or two cycle instructions. Once in the interrupt service routine the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid multiple interrupt requests.

Note 1: Individual interrupt flag bits are set, regardless of the status of their corresponding mask bit or the GIE bit.

2: When an instruction that clears the GIE bit is executed, any interrupts that were pending for execution in the next cycle are ignored. The CPU will execute a `NOP` in the cycle immediately following the instruction which clears the GIE bit. The interrupts which were ignored are still pending to be serviced when the GIE bit is set again.

FIGURE 10-15: INTERRUPT LOGIC



10.5.1 RB0/INT INTERRUPT

External interrupt on RB0/INT pin is edge triggered; either rising if INTEDG bit (OPTION<6>) is set, or falling, if INTEDG bit is clear. When a valid edge appears on the RB0/INT pin, the INTF bit (INTCON<1>) is set. This interrupt can be disabled by clearing the INTE control bit (INTCON<4>). The INTF bit must be cleared in software in the interrupt service routine before re-enabling this interrupt. The RB0/INT interrupt can wake-up the processor from SLEEP, if the INTE bit was set prior to going into SLEEP. The status of the GIE bit decides whether or not the processor branches to the interrupt vector following wake-up. See Section 10.8 for details on SLEEP and Figure 10-19 for timing of wake-up from SLEEP through RB0/INT interrupt.

10.5.2 TMR0 INTERRUPT

An overflow (FFh → 00h) in the TMR0 register will set the T0IF (INTCON<2>) bit. The interrupt can be enabled/disabled by setting/clearing T0IE (INTCON<5>) bit. For operation of the Timer0 module, see Section 7.0.

10.5.3 PORTB INTERRUPT

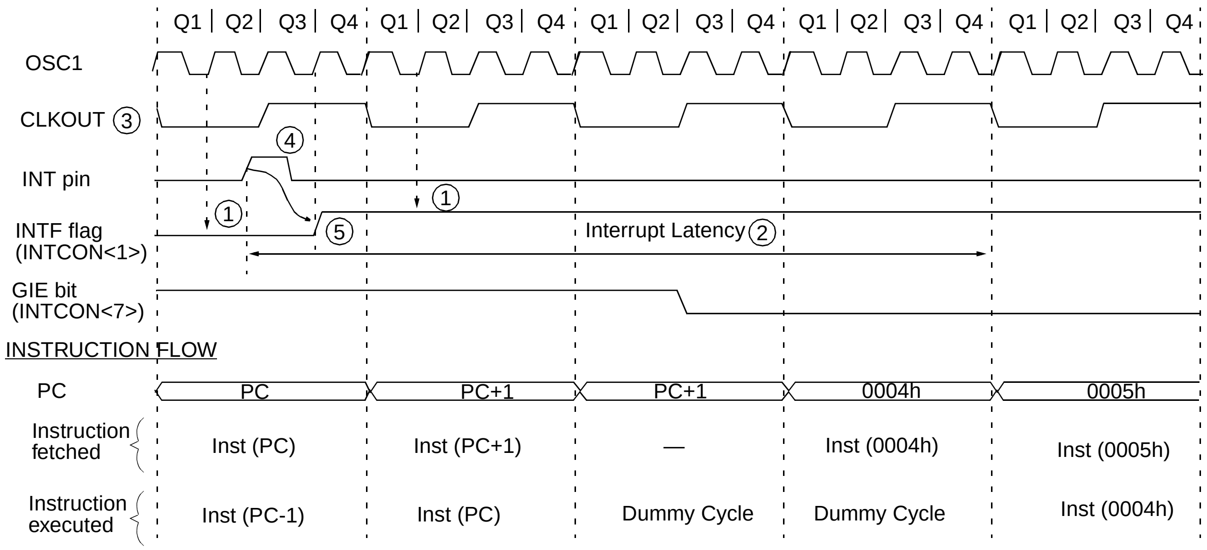
An input change on PORTB <7:4> sets the RBIF (INTCON<0>) bit. The interrupt can be enabled/disabled by setting/clearing the RBIE (INTCON<4>) bit. For operation of PORTB (Section 5.2).

Note:	If a change on the I/O pin should occur when the read operation is being executed (start of the Q2 cycle), then the RBIF interrupt flag may not get set.

10.5.4 COMPARATOR INTERRUPT

See Section 8.6 for complete description of comparator interrupts.

FIGURE 10-16: INT PIN INTERRUPT TIMING


Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4
OSC1
CLKOUT ③
④
INT pin
①
①
INTF flag (INTCON<1>)
⑤
Interrupt Latency ②
GIE bit (INTCON<7>)
INSTRUCTION FLOW
PC
PC
PC+1
PC+1
0004h
0005h
Instruction fetched
Inst (PC)
Inst (PC+1)
—
Inst (0004h)
Inst (0005h)
Instruction executed
Inst (PC-1)
Inst (PC)
Dummy Cycle
Dummy Cycle
Inst (0004h)

Note 1: INTF flag is sampled here (every Q1).

2: Interrupt latency = 3-4 Tcy where TCY = instruction cycle time.
Latency is the same whether Inst (PC) is a single cycle or a 2-cycle instruction.

3: CLKOUT is available only in RC oscillator mode.

4: For minimum width of INT pulse, refer to AC specs.

5: INTF is enabled to be set anytime during the Q4-Q1 cycles.

10.6 Context Saving During Interrupts

During an interrupt, only the return PC value is saved on the stack. Typically, users may wish to save key registers during an interrupt (i.e. W register and STATUS register). This will have to be implemented in software.

Example 10-1 stores and restores the STATUS and W registers. The user register, W_TEMP, must be defined in both banks and must be defined at the same offset from the bank base address (i.e., W_TEMP is defined at 0x70 in Bank 0 and it must also be defined at 0xF0 in Bank 1). The user register, STATUS_TEMP, must be defined in Bank 0. The Example 10-1:

- Stores the W register
- Stores the STATUS register in Bank 0
- Executes the ISR code
- Restores the STATUS (and bank select bit register)
- Restores the W register

EXAMPLE 10-1: SAVING THE STATUS AND W REGISTERS IN RAM

```
MOVWF   W_TEMP          ;copy W to temp register,
                        ;could be in either bank
SWAPF   STATUS,W        ;swap status to be saved into W
BCF     STATUS,RP0      ;change to bank 0 regardless
                        ;of current bank
MOVWF   STATUS_TEMP     ;save status to bank 0
                        ;register
     :
     :  (ISR)
     :
SWAPF   STATUS_TEMP,W   ;swap STATUS_TEMP register
                        ;into W, sets bank to original
                        ;state
MOVWF   STATUS          ;move W into STATUS register
SWAPF   W_TEMP,F        ;swap W_TEMP
SWAPF   W_TEMP,W        ;swap W_TEMP into W
```

10.7 Watchdog Timer (WDT)

The Watchdog Timer is a free running on-chip RC oscillator which does not require any external components. This RC oscillator is separate from the RC oscillator of the CLKIN pin. That means that the WDT will run, even if the clock on the OSC1 and OSC2 pins of the device have been stopped, for example, by execution of a `SLEEP` instruction. During normal operation, a WDT time-out generates a device RESET. If the device is in SLEEP mode, a WDT time-out causes the device to wake-up and continue with normal operation. The WDT can be permanently disabled by programming the configuration bit WDTE as clear (Section 10.1).

10.7.1 WDT PERIOD

The WDT has a nominal time-out period of 18 ms, (with no prescaler). The time-out periods vary with temperature, VDD and process variations from part to part (see DC specs). If longer time-out periods are desired, a prescaler with a division ratio of up to 1:128 can be assigned to the WDT under software control by writing to the OPTION register. Thus, time-out periods up to 2.3 seconds can be realized.

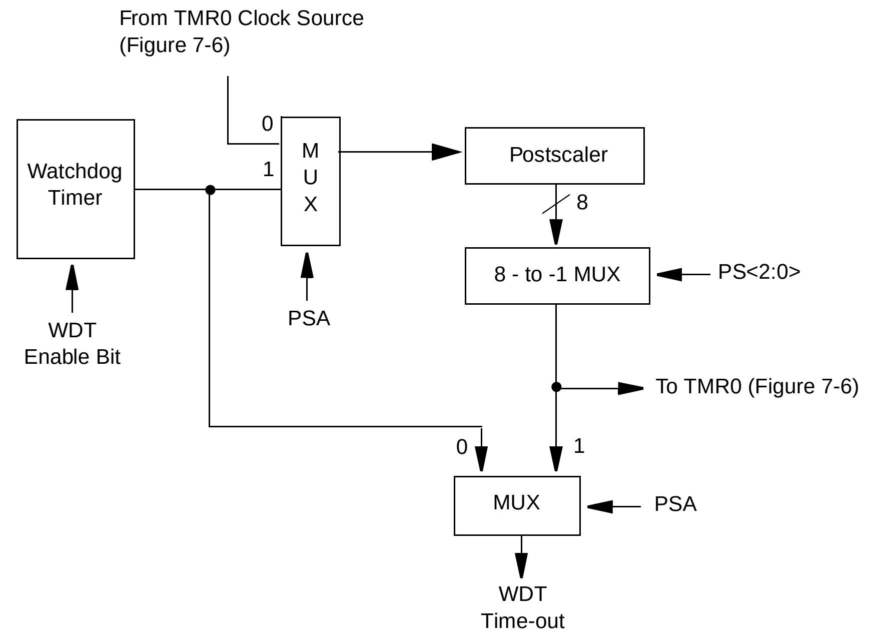
The `CLRWDT` and `SLEEP` instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET.

The $\overline{\text{TO}}$ bit in the STATUS register will be cleared upon a Watchdog Timer time-out.

10.7.2 WDT PROGRAMMING CONSIDERATIONS

It should also be taken in account that under worst case conditions (VDD = Min., Temperature = Max., max. WDT prescaler), it may take several seconds before a WDT time-out occurs.

FIGURE 10-17: WATCHDOG TIMER BLOCK DIAGRAM


From TMR0 Clock Source
(Figure 7-6)
Watchdog Timer
0
1
M U X
Postscaler
8
8 - to -1 MUX
PS<2:0>
WDT Enable Bit
PSA
To TMR0 (Figure 7-6)
0
1
MUX
PSA
WDT Time-out

Note: T0SE, T0CS, PSA, PS<2:0> are bits in the OPTION register.

FIGURE 10-18: SUMMARY OF WATCHDOG TIMER REGISTERS

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2007h	Config. bits	—	BOREN	CP1	CP0	$\overline{\text{PWRTE}}$	WDTE	FOSC1	FOSC0
81h	OPTION	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

Legend: – = Unimplemented location, read as “0”, + = Reserved for future use

Note: Shaded cells are not used by the Watchdog Timer.

10.8 Power-Down Mode (SLEEP)

The Power-down mode is entered by executing a `SLEEP` instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{PD}$ bit in the STATUS register is cleared, the $\overline{TO}$ bit is set and the oscillator driver is turned off. The I/O ports maintain the status they had before `SLEEP` was executed (driving high, low, or hi-impedance).

For lowest current consumption in this mode, all I/O pins should be either at VDD or VSS, with no external circuitry drawing current from the I/O pin, and the comparators and VREF should be disabled. I/O pins that are hi-impedance inputs should be pulled high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at VDD or VSS for lowest current consumption. The contribution from on chip pull-ups on PORTB should be considered.

The $\overline{MCLR}$ pin must be at a logic high level (VIHMC).

Note: It should be noted that a RESET generated by a WDT time-out does not drive $\overline{MCLR}$ pin low.

10.8.1 WAKE-UP FROM SLEEP

The device can wake-up from SLEEP through one of the following events:

1. External reset input on $\overline{MCLR}$ pin
2. Watchdog Timer Wake-up (if WDT was enabled)
3. Interrupt from RB0/INT pin, RB Port change, or the Peripheral Interrupt (Comparator).

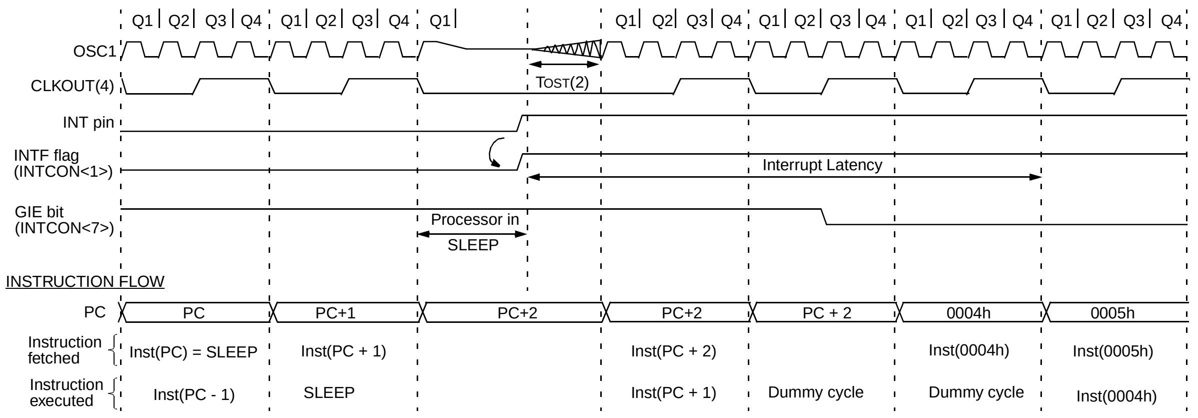
The first event will cause a device reset. The two latter events are considered a continuation of program execution. The $\overline{TO}$ and $\overline{PD}$ bits in the STATUS register can be used to determine the cause of device reset. $\overline{PD}$ bit, which is set on power-up is cleared when SLEEP is invoked. $\overline{TO}$ bit is cleared if WDT wake-up occurred.

When the `SLEEP` instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the `SLEEP` instruction. If the GIE bit is set (enabled), the device executes the instruction after the `SLEEP` instruction and then branches to the interrupt address (0004h). In cases where the execution of the instruction following `SLEEP` is not desirable, the user should have an `NOP` after the `SLEEP` instruction.

Note: If the global interrupts are disabled (GIE is cleared), but any interrupt source has both its interrupt enable bit and the corresponding interrupt flag bits set, the device will immediately wake-up from sleep. The sleep instruction is completely executed.

The WDT is cleared when the device wakes-up from sleep, regardless of the source of wake-up.

FIGURE 10-19: WAKE-UP FROM SLEEP THROUGH INTERRUPT


Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1
Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4 Q1 Q2 Q3 Q4
OSC1
CLKOUT(4)
TOST(2)
INT pin
INTF flag (INTCON<1>)
Interrupt Latency
GIE bit (INTCON<7>)
Processor in SLEEP
INSTRUCTION FLOW
PC
PC
PC+1
PC+2
PC+2
PC + 2
0004h
0005h
Instruction fetched
Inst(PC) = SLEEP
Inst(PC + 1)
Inst(PC + 2)
Inst(0004h)
Inst(0005h)
Instruction executed
Inst(PC - 1)
SLEEP
Inst(PC + 1)
Dummy cycle
Dummy cycle
Inst(0004h)

Note 1: XT, HS or LP oscillator mode assumed.
2: TOST = 1024TOSC (drawing not to scale) This delay does not occur for RC osc mode.
3: GIE = ’1’ assumed. In this case after wake- up, the processor jumps to the interrupt routine. If GIE = ’0’, execution will continue in-line.
4: CLKOUT is not available in these osc modes, but shown here for timing reference.

10.9 Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out for verification purposes.

Note:	Microchip does not recommend code protecting windowed devices.

10.10 ID Locations

Four memory locations (2000h-2003h) are designated as ID locations where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution but are readable and writable during program/verify. Only the least significant 4 bits of the ID locations are used.

10.11 In-Circuit Serial Programming

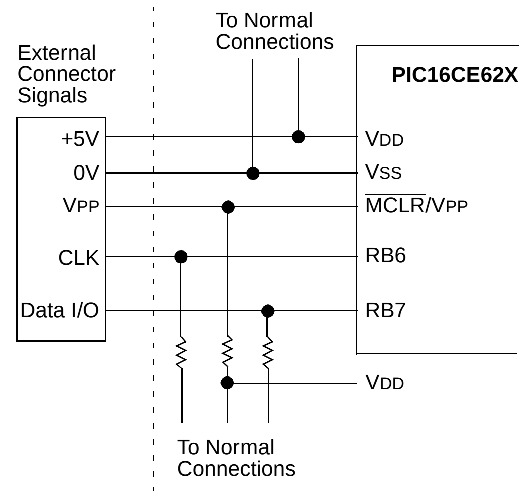
The PIC16CE62X microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

The device is placed into a program/verify mode by holding the RB6 and RB7 pins low, while raising the $\overline{\text{MCLR}}$ (V_{PP}) pin from V_{IL} to V_{IHH} (see programming specification). RB6 becomes the programming clock and RB7 becomes the programming data. Both RB6 and RB7 are Schmitt Trigger inputs in this mode.

After reset, to place the device into programming/verify mode, the program counter (PC) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14-bits of program data are then supplied to or from the device, depending if the command was a load or a read. For complete details of serial programming, please refer to the PIC16C6X/7X/9XX Programming Specifications (Literature #DS30228).

A typical in-circuit serial programming connection is shown in Figure 10-20.

FIGURE 10-20: TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION



11.0 INSTRUCTION SET SUMMARY

Each PIC16CE62X instruction is a 14-bit word divided into an OPCODE which specifies the instruction type and one or more operands which further specify the operation of the instruction. The PIC16CE62X instruction set summary in Table 11-2 lists **byte-oriented**, **bit-oriented**, and **literal and control** operations. Table 11-1 shows the opcode field descriptions.

For **byte-oriented** instructions, ’f’ represents a file register designator and ’d’ represents a destination designator. The file register designator specifies which file register is to be used by the instruction.

The destination designator specifies where the result of the operation is to be placed. If ’d’ is zero, the result is placed in the W register. If ’d’ is one, the result is placed in the file register specified in the instruction.

For **bit-oriented** instructions, ’b’ represents a bit field designator which selects the number of the bit affected by the operation, while ’f’ represents the number of the file in which the bit is located.

For **literal and control** operations, ’k’ represents an eight or eleven bit constant or literal value.

TABLE 11-1: OPCODE FIELD DESCRIPTIONS

Field	Description
`f`	Register file address (0x00 to 0x7F)
`W`	Working register (accumulator)
`b`	Bit address within an 8-bit file register
`k`	Literal field, constant data or label
`x`	Don’t care location (= 0 or 1) The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
`d`	Destination select; d = 0: store result in W, d = 1: store result in file register f. Default is d = 1
`label`	Label name
`TOS`	Top of Stack
`PC`	Program Counter
`PCLATH`	Program Counter High Latch
`GIE`	Global Interrupt Enable bit
`WDT`	Watchdog Timer/Counter
$\overline{\text{TO}}$	Time-out bit
$\overline{\text{PD}}$	Power-down bit
`dest`	Destination either the W register or the specified register file location
[ ]	Options
( )	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
italics	User defined term (font is courier)

The instruction set is highly orthogonal and is grouped into three basic categories:

- **Byte-oriented** operations
- **Bit-oriented** operations
- **Literal and control** operations

All instructions are executed within one single instruction cycle, unless a conditional test is true or the program counter is changed as a result of an instruction. In this case, the execution takes two instruction cycles with the second cycle executed as a NOP. One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μs. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 2 μs.

Table 11-1 lists the instructions recognized by the MPASM assembler.

Figure 11-1 shows the three general formats that the instructions can have.

Note:	To maintain upward compatibility with future PICmicro® products, <u>do not use</u> the `OPTION` and `TRIS` instructions.

All examples use the following format to represent a hexadecimal number:

0xhh

where h signifies a hexadecimal digit.

FIGURE 11-1: GENERAL FORMAT FOR INSTRUCTIONS

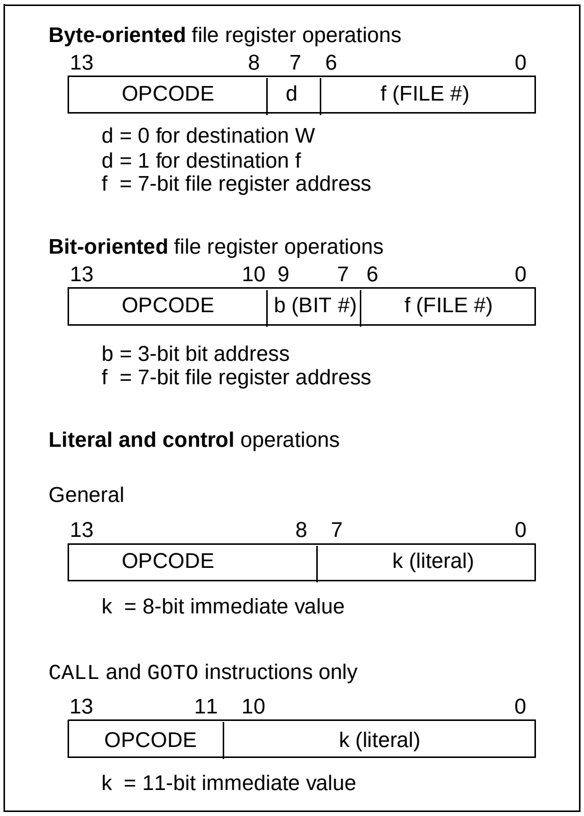

Byte-oriented file register operations
13
8
7
6
0
OPCODE
d
f (FILE #)
d = 0 for destination W
d = 1 for destination f
f = 7-bit file register address
Bit-oriented file register operations
13
10
9
7
6
0
OPCODE
b (BIT #)
f (FILE #)
b = 3-bit bit address
f = 7-bit file register address
Literal and control operations
General
13
8
7
0
OPCODE
k (literal)
k = 8-bit immediate value
CALL and GOTO instructions only
13
11
10
0
OPCODE
k (literal)
k = 11-bit immediate value

TABLE 11-2: PIC16CE62X INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb			LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	**f, d**	Add W and f	1	`00`	`0111`	`dfff`	`ffff`	C,DC,Z	1,2
ANDWF	**f, d**	AND W with f	1	`00`	`0101`	`dfff`	`ffff`	Z	1,2
CLRF	**f**	Clear f	1	`00`	`0001`	`lfff`	`ffff`	Z	2
CLRW	**-**	Clear W	1	`00`	`0001`	`0000`	`0011`	Z	
COMF	**f, d**	Complement f	1	`00`	`1001`	`dfff`	`ffff`	Z	1,2
DECF	**f, d**	Decrement f	1	`00`	`0011`	`dfff`	`ffff`	Z	1,2
DECFSZ	**f, d**	Decrement f, Skip if 0	1(2)	`00`	`1011`	`dfff`	`ffff`		1,2,3
INCF	**f, d**	Increment f	1	`00`	`1010`	`dfff`	`ffff`	Z	1,2
INCFSZ	**f, d**	Increment f, Skip if 0	1(2)	`00`	`1111`	`dfff`	`ffff`		1,2,3
IORWF	**f, d**	Inclusive OR W with f	1	`00`	`0100`	`dfff`	`ffff`	Z	1,2
MOVF	**f, d**	Move f	1	`00`	`1000`	`dfff`	`ffff`	Z	1,2
MOVWF	**f**	Move W to f	1	`00`	`0000`	`lfff`	`ffff`		
NOP	**-**	No Operation	1	`00`	`0000`	`0xx0`	`0000`		
RLF	**f, d**	Rotate Left f through Carry	1	`00`	`1101`	`dfff`	`ffff`	C	1,2
RRF	**f, d**	Rotate Right f through Carry	1	`00`	`1100`	`dfff`	`ffff`	C	1,2
SUBWF	**f, d**	Subtract W from f	1	`00`	`0010`	`dfff`	`ffff`	C,DC,Z	1,2
SWAPF	**f, d**	Swap nibbles in f	1	`00`	`1110`	`dfff`	`ffff`		1,2
XORWF	**f, d**	Exclusive OR W with f	1	`00`	`0110`	`dfff`	`ffff`	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	**f, b**	Bit Clear f	1	`01`	`00bb`	`bfff`	`ffff`		1,2
BSF	**f, b**	Bit Set f	1	`01`	`01bb`	`bfff`	`ffff`		1,2
BTFSC	**f, b**	Bit Test f, Skip if Clear	1 (2)	`01`	`10bb`	`bfff`	`ffff`		3
BTFSS	**f, b**	Bit Test f, Skip if Set	1 (2)	`01`	`11bb`	`bfff`	`ffff`		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	**k**	Add literal and W	1	`11`	`111x`	`kkkk`	`kkkk`	C,DC,Z	
ANDLW	**k**	AND literal with W	1	`11`	`1001`	`kkkk`	`kkkk`	Z	
CALL	**k**	Call subroutine	2	`10`	`0kkk`	`kkkk`	`kkkk`		
CLRWDT	**-**	Clear Watchdog Timer	1	`00`	`0000`	`0110`	`0100`	$\overline{TO}$,$\overline{PD}$	
GOTO	**k**	Go to address	2	`10`	`1kkk`	`kkkk`	`kkkk`		
IORLW	**k**	Inclusive OR literal with W	1	`11`	`1000`	`kkkk`	`kkkk`	Z	
MOVLW	**k**	Move literal to W	1	`11`	`00xx`	`kkkk`	`kkkk`		
RETFIE	**-**	Return from interrupt	2	`00`	`0000`	`0000`	`1001`		
RETLW	**k**	Return with literal in W	2	`11`	`01xx`	`kkkk`	`kkkk`		
RETURN	**-**	Return from Subroutine	2	`00`	`0000`	`0000`	`1000`		
SLEEP	**-**	Go into standby mode	1	`00`	`0000`	`0110`	`0011`	$\overline{TO}$,$\overline{PD}$	
SUBLW	**k**	Subtract W from literal	1	`11`	`110x`	`kkkk`	`kkkk`	C,DC,Z	
XORLW	**k**	Exclusive OR literal with W	1	`11`	`1010`	`kkkk`	`kkkk`	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

11.1 Instruction Descriptions

ADDLW	**Add Literal and W**
Syntax:	[*label*] ADDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) + k → (W)
Status Affected:	C, DC, Z
Encoding:	`11` `111x` `kkkk` `kkkk`
Description:	The contents of the W register are added to the eight bit literal 'k' and the result is placed in the W register.
Words:	1
Cycles:	1
Example	`ADDLW 0x15` Before Instruction W = 0x10 After Instruction W = 0x25

ADDWF	**Add W and f**
Syntax:	[*label*] ADDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) + (f) → (dest)
Status Affected:	C, DC, Z
Encoding:	`00` `0111` `dfff` `ffff`
Description:	Add the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.
Words:	1
Cycles:	1
Example	`ADDWF FSR, 0` Before Instruction W = 0x17 FSR = 0xC2 After Instruction W = 0xD9 FSR = 0xC2

ANDLW	**AND Literal with W**
Syntax:	[*label*] ANDLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .AND. (k) → (W)
Status Affected:	Z
Encoding:	`11` `1001` `kkkk` `kkkk`
Description:	The contents of W register are AND'ed with the eight bit literal 'k'. The result is placed in the W register.
Words:	1
Cycles:	1
Example	`ANDLW 0x5F` Before Instruction W = 0xA3 After Instruction W = 0x03

ANDWF	**AND W with f**
Syntax:	[*label*] ANDWF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(W) .AND. (f) → (dest)
Status Affected:	Z
Encoding:	`00` `0101` `dfff` `ffff`
Description:	AND the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.
Words:	1
Cycles:	1
Example	`ANDWF FSR, 1` Before Instruction W = 0x17 FSR = 0xC2 After Instruction W = 0x17 FSR = 0x02

BCF Bit Clear f

Syntax: [*label*] BCF f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: $0 \rightarrow (f<b>)$

Status Affected: None

Encoding:

`01`	`00bb`	`bfff`	`ffff`

Description: Bit 'b' in register 'f' is cleared.

Words: 1

Cycles: 1

Example

```
BCF      FLAG_REG, 7
```

Before Instruction
FLAG_REG = 0xC7
After Instruction
FLAG_REG = 0x47

BSF Bit Set f

Syntax: [*label*] BSF f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: $1 \rightarrow (f<b>)$

Status Affected: None

Encoding:

`01`	`01bb`	`bfff`	`ffff`

Description: Bit 'b' in register 'f' is set.

Words: 1

Cycles: 1

Example

```
BSF      FLAG_REG,   7
```

Before Instruction
FLAG_REG = 0x0A
After Instruction
FLAG_REG = 0x8A

BTFSC Bit Test, Skip if Clear

Syntax: [*label*] BTFSC f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

`01`	`10bb`	`bfff`	`ffff`

Description: If bit 'b' in register 'f' is '0', then the next instruction is skipped.
If bit 'b' is '0', then the next instruction fetched during the current instruction execution is discarded, and a `NOP` is executed instead, making this a two-cycle instruction.

Words: 1

Cycles: 1(2)

Example

```
HERE     BTFSC   FLAG,1
FALSE    GOTO    PROCESS_CODE
TRUE     •
         •
         •
```

Before Instruction
PC = address `HERE`
After Instruction
if FLAG<1> = 0,
PC = address `TRUE`
if FLAG<1>=1,
PC = address `FALSE`

BTFSS	Bit Test f, Skip if Set
Syntax:	[*label*] BTFSS f,b
Operands:	0 ≤ f ≤ 127 0 ≤ b < 7
Operation:	skip if (f<b>) = 1
Status Affected:	None
Encoding:	`01` | `11bb` | `bfff` | `ffff`
Description:	If bit ’b’ in register ’f’ is ’1’ then the next instruction is skipped. If bit ’b’ is ’1’, then the next instruction fetched during the current instruction execution, is discarded and a `NOP` is executed instead, making this a two-cycle instruction.
Words:	1
Cycles:	1(2)

Example

```
HERE    BTFSS  FLAG,1
FALSE   GOTO   PROCESS_CODE
TRUE    •
        •
        •
```

Before Instruction

PC = address `HERE`

After Instruction

if FLAG<1> = 0,
PC = address `FALSE`
if FLAG<1> = 1,
PC = address `TRUE`

CALL	Call Subroutine
Syntax:	[*label*] CALL k
Operands:	0 ≤ k ≤ 2047
Operation:	(PC)+ 1→ TOS, k → PC<10:0>, (PCLATH<4:3>) → PC<12:11>
Status Affected:	None
Encoding:	`10` | `0kkk` | `kkkk` | `kkkk`
Description:	Call Subroutine. First, return address (PC+1) is pushed onto the stack. The eleven bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. `CALL` is a two-cycle instruction.
Words:	1
Cycles:	2

Example

```
HERE     CALL   THERE
```

Before Instruction

PC = Address `HERE`

After Instruction

PC = Address `THERE`
TOS = Address `HERE+1`

CLRF	Clear f
Syntax:	[*label*] CLRF f
Operands:	0 ≤ f ≤ 127
Operation:	00h → (f) 1 → Z
Status Affected:	Z
Encoding:	`00` | `0001` | `1fff` | `ffff`
Description:	The contents of register ’f’ are cleared and the Z bit is set.
Words:	1
Cycles:	1

Example

```
CLRF        FLAG_REG
```

Before Instruction

FLAG_REG = 0x5A

After Instruction

FLAG_REG = 0x00
Z = 1

CLRW	Clear W
Syntax:	[*label*] CLRW
Operands:	None
Operation:	00h → (W) 1 → Z
Status Affected:	Z
Encoding:	`00` | `0001` | `0000` | `0011`
Description:	W register is cleared. Zero bit (Z) is set.
Words:	1
Cycles:	1

Example

```
CLRW
```

Before Instruction

W = 0x5A

After Instruction

W = 0x00
Z = 1

CLRWDT — Clear Watchdog Timer

Syntax: [*label*] CLRWDT

Operands: None

Operation: 00h → WDT
0 → WDT prescaler,
1 → $\overline{TO}$
1 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

`00`	`0000`	`0110`	`0100`

Description: `CLRWDT` instruction resets the Watchdog Timer. It also resets the prescaler of the WDT. Status bits $\overline{TO}$ and $\overline{PD}$ are set.

Words: 1

Cycles: 1

Example

```
CLRWDT
```

Before Instruction
WDT counter = ?
After Instruction
WDT counter = 0x00
WDT prescaler = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

COMF — Complement f

Syntax: [*label*] COMF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: ($\bar{f}$) → (dest)

Status Affected: Z

Encoding:

`00`	`1001`	`dfff`	`ffff`

Description: The contents of register ’f’ are complemented. If ’d’ is 0, the result is stored in W. If ’d’ is 1, the result is stored back in register ’f’.

Words: 1

Cycles: 1

Example

```
COMF       REG1,0
```

Before Instruction
REG1 = 0x13
After Instruction
REG1 = 0x13
W = 0xEC

DECF — Decrement f

Syntax: [*label*] DECF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) - 1 → (dest)

Status Affected: Z

Encoding:

`00`	`0011`	`dfff`	`ffff`

Description: Decrement register ’f’. If ’d’ is 0, the result is stored in the W register. If ’d’ is 1, the result is stored back in register ’f’.

Words: 1

Cycles: 1

Example

```
DECF    CNT, 1
```

Before Instruction
CNT = 0x01
Z = 0
After Instruction
CNT = 0x00
Z = 1

DECFSZ — Decrement f, Skip if 0

Syntax: [*label*] DECFSZ f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

Operation: (f) - 1 → (dest); skip if result = 0

Status Affected: None

Encoding:

`00`	`1011`	`dfff`	`ffff`

Description: The contents of register ’f’ are decremented. If ’d’ is 0, the result is placed in the W register. If ’d’ is 1, the result is placed back in register ’f’.
If the result is 0, the next instruction, which is already fetched, is discarded. A `NOP` is executed instead making it a two-cycle instruction.

Words: 1

Cycles: 1(2)

Example

```
HERE      DECFSZ   CNT, 1
          GOTO     LOOP
CONTINUE  •
          •
          •
```

Before Instruction
PC = address `HERE`
After Instruction
CNT = CNT - 1
if CNT = 0,
PC = address `CONTINUE`
if CNT ≠ 0,
PC = address `HERE+1`

GOTO Unconditional Branch

Syntax:	[*label*] GOTO k
Operands:	0 ≤ k ≤ 2047
Operation:	k → PC<10:0> PCLATH<4:3> → PC<12:11>
Status Affected:	None
Encoding:	`10` `1kkk` `kkkk` `kkkk`
Description:	`GOTO` is an unconditional branch. The eleven bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. `GOTO` is a two-cycle instruction.
Words:	1
Cycles:	2

Example

```
GOTO THERE
```

After Instruction

PC = Address `THERE`

INCF Increment f

Syntax:	[*label*] INCF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (dest)
Status Affected:	Z
Encoding:	`00` `1010` `dfff` `ffff`
Description:	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.
Words:	1
Cycles:	1

Example

```
INCF    CNT, 1
```

Before Instruction

CNT = 0xFF
Z = 0

After Instruction

CNT = 0x00
Z = 1

INCFSZ Increment f, Skip if 0

Syntax:	[*label*] INCFSZ f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (dest), skip if result = 0
Status Affected:	None
Encoding:	`00` `1111` `dfff` `ffff`
Description:	The contents of register 'f' are incremented. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'. If the result is 0, the next instruction, which is already fetched, is discarded. A `NOP` is executed instead making it a two-cycle instruction.
Words:	1
Cycles:	1(2)

Example

```
HERE      INCFSZ     CNT,  1
          GOTO      LOOP
CONTINUE  •
          •
          •
```

Before Instruction

PC = address `HERE`

After Instruction

CNT = CNT + 1
if CNT= 0,
PC = address `CONTINUE`
if CNT≠ 0,
PC = address `HERE +1`

IORLW Inclusive OR Literal with W

Syntax:	[*label*] IORLW k
Operands:	0 ≤ k ≤ 255
Operation:	(W) .OR. k → (W)
Status Affected:	Z
Encoding:	`11` `1000` `kkkk` `kkkk`
Description:	The contents of the W register are OR'ed with the eight bit literal 'k'. The result is placed in the W register.
Words:	1
Cycles:	1

Example

```
IORLW   0x35
```

Before Instruction

W = 0x9A

After Instruction

W = 0xBF
Z = 1

IORWF	**Inclusive OR W with f**
Syntax:	[*label*] IORWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .OR. (f) → (dest)
Status Affected:	Z
Encoding:	`00` `0100` `dfff` `ffff`
Description:	Inclusive OR the W register with register ’f’. If ’d’ is 0, the result is placed in the W register. If ’d’ is 1, the result is placed back in register ’f’.
Words:	1
Cycles:	1
Example	`IORWF RESULT, 0` Before Instruction RESULT = 0x13 W = 0x91 After Instruction RESULT = 0x13 W = 0x93 Z = 1

MOVLW	**Move Literal to W**
Syntax:	[*label*] MOVLW k
Operands:	$0 \le k \le 255$
Operation:	k → (W)
Status Affected:	None
Encoding:	`11` `00xx` `kkkk` `kkkk`
Description:	The eight bit literal ’k’ is loaded into W register. The don’t cares will assemble as 0’s.
Words:	1
Cycles:	1
Example	`MOVLW 0x5A` After Instruction W = 0x5A

MOVF	**Move f**
Syntax:	[*label*] MOVF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) → (dest)
Status Affected:	Z
Encoding:	`00` `1000` `dfff` `ffff`
Description:	The contents of register f are moved to a destination dependant upon the status of d. If d = 0, destination is W register. If d = 1, the destination is file register f itself. d = 1 is useful to test a file register since status flag Z is affected.
Words:	1
Cycles:	1
Example	`MOVF FSR, 0` After Instruction W = value in FSR register Z = 1

MOVWF	**Move W to f**
Syntax:	[*label*] MOVWF f
Operands:	$0 \le f \le 127$
Operation:	(W) → (f)
Status Affected:	None
Encoding:	`00` `0000` `1fff` `ffff`
Description:	Move data from W register to register 'f'.
Words:	1
Cycles:	1
Example	`MOVWF OPTION` Before Instruction OPTION = 0xFF W = 0x4F After Instruction OPTION = 0x4F W = 0x4F

NOP	**No Operation**
Syntax:	[*label*] NOP
Operands:	None
Operation:	No operation
Status Affected:	None
Encoding:	00 0000 0xx0 0000
Description:	No operation.
Words:	1
Cycles:	1
Example	`NOP`

OPTION	**Load Option Register**
Syntax:	[*label*] OPTION
Operands:	None
Operation:	(W) → OPTION
Status Affected:	None
Encoding:	00 0000 0110 0010
Description:	The contents of the W register are loaded in the OPTION register. This instruction is supported for code compatibility with PIC16C5X products. Since OPTION is a readable/writable register, the user can directly address it.
Words:	1
Cycles:	1
Example	
	To maintain upward compatibility with future PICmicro® products, do not use this instruction.

RETFIE	**Return from Interrupt**
Syntax:	[*label*] RETFIE
Operands:	None
Operation:	TOS → PC, 1 → GIE
Status Affected:	None
Encoding:	00 0000 0000 1001
Description:	Return from Interrupt. Stack is POPed and Top of Stack (TOS) is loaded in the PC. Interrupts are enabled by setting Global Interrupt Enable bit, GIE (INTCON<7>). This is a two-cycle instruction.
Words:	1
Cycles:	2
Example	`RETFIE`

After Interrupt

PC = TOS

GIE = 1

RETLW	**Return with Literal in W**
Syntax:	[*label*] RETLW k
Operands:	$0 \le k \le 255$
Operation:	k → (W); TOS → PC
Status Affected:	None
Encoding:	11 01xx kkkk kkkk
Description:	The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two-cycle instruction.
Words:	1
Cycles:	2

Example

```
        CALL TABLE    ;W contains table
                      ;offset value
        •             ;W now has table
        value
        •
TABLE   •
        ADDWF PC      ;W = offset
        RETLW k1      ;Begin table
        RETLW k2      ;
        •
        •
        •
        RETLW kn      ; End of table
```

Before Instruction

W = 0x07

After Instruction

W = value of k8

RETURN — Return from Subroutine

Syntax: [*label*] RETURN

Operands: None

Operation: TOS → PC

Status Affected: None

Encoding:

`00`	`0000`	`0000`	`1000`

Description: Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two cycle instruction.

Words: 1

Cycles: 2

Example `RETURN`

After Interrupt

```
PC  =   TOS
```

RLF — Rotate Left f through Carry

Syntax: [*label*] RLF f,d

Operands: $0 \leq f \leq 127$
$d \in [0,1]$

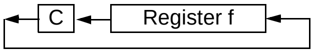
Operation: See description below

Status Affected: C

Encoding:

`00`	`1101`	`dfff`	`ffff`

Description: The contents of register 'f' are rotated one bit to the left through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is stored back in register 'f'.



Words: 1

Cycles: 1

Example `RLF REG1,0`

Before Instruction

```
REG1   =   1110 0110
C      =   0
```

After Instruction

```
REG1   =   1110 0110
W      =   1100 1100
C      =   1
```

RRF — Rotate Right f through Carry

Syntax: [*label*] RRF f,d

Operands: $0 \leq f \leq 127$
$d \in [0,1]$

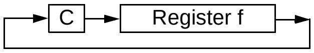
Operation: See description below

Status Affected: C

Encoding:

`00`	`1100`	`dfff`	`ffff`

Description: The contents of register 'f' are rotated one bit to the right through the Carry Flag. If 'd' is 0, the result is placed in the W register. If 'd' is 1, the result is placed back in register 'f'.



Words: 1

Cycles: 1

Example `RRF REG1,0`

Before Instruction

```
REG1   =   1110 0110
C      =   0
```

After Instruction

```
REG1   =   1110 0110
W      =   0111 0011
C      =   0
```

SLEEP

Syntax: [*label*] SLEEP

Operands: None

Operation: 00h → WDT,
0 → WDT prescaler,
1 → $\overline{\text{TO}}$,
0 → $\overline{\text{PD}}$

Status Affected: $\overline{\text{TO}}$, $\overline{\text{PD}}$

Encoding:

`00`	`0000`	`0110`	`0011`

Description: The power-down status bit, $\overline{\text{PD}}$ is cleared. Time-out status bit, $\overline{\text{TO}}$ is set. Watchdog Timer and its prescaler are cleared.
The processor is put into `SLEEP` mode with the oscillator stopped. See Section 10.8 for more details.

Words: 1

Cycles: 1

Example: SLEEP

SUBLW Subtract W from Literal

Syntax:	[*label*] SUBLW k
Operands:	$0 \le k \le 255$
Operation:	$k - (W) \rightarrow (W)$
Status Affected:	C, DC, Z
Encoding:	`11` | `110x` | `kkkk` | `kkkk`
Description:	The W register is subtracted (2's complement method) from the eight bit literal 'k'. The result is placed in the W register.
Words:	1
Cycles:	1
Example 1:	`SUBLW 0x02` Before Instruction W = 1 C = ? After Instruction W = 1 C = 1; result is positive
Example 2:	Before Instruction W = 2 C = ? After Instruction W = 0 C = 1; result is zero
Example 3:	Before Instruction W = 3 C = ? After Instruction W = 0xFF C = 0; result is negative

SUBWF Subtract W from f

Syntax:	[*label*] SUBWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	$(f) - (W) \rightarrow (dest)$
Status Affected:	C, DC, Z
Encoding:	`00` | `0010` | `dfff` | `ffff`
Description:	Subtract (2's complement method) W register from register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.
Words:	1
Cycles:	1
Example 1:	`SUBWF REG1,1` Before Instruction REG1 = 3 W = 2 C = ? After Instruction REG1 = 1 W = 2 C = 1; result is positive
Example 2:	Before Instruction REG1 = 2 W = 2 C = ? After Instruction REG1 = 0 W = 2 C = 1; result is zero
Example 3:	Before Instruction REG1 = 1 W = 2 C = ? After Instruction REG1 = 0xFF W = 2 C = 0; result is negative

SWAPF	**Swap Nibbles in f**
Syntax:	[*label*] SWAPF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f<3:0>) → (dest<7:4>), (f<7:4>) → (dest<3:0>)
Status Affected:	None
Encoding:	`00` `1110` `dfff` `ffff`
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0, the result is placed in W register. If 'd' is 1, the result is placed in register 'f'.
Words:	1
Cycles:	1
Example	`SWAPF REG, 0` Before Instruction REG1 = 0xA5 After Instruction REG1 = 0xA5 W = 0x5A

TRIS	**Load TRIS Register**
Syntax:	[*label*] TRIS f
Operands:	$5 \le f \le 7$
Operation:	(W) → TRIS register f;
Status Affected:	None
Encoding:	`00` `0000` `0110` `0fff`
Description:	The instruction is supported for code compatibility with the PIC16C5X products. Since TRIS registers are readable and writable, the user can directly address them.
Words:	1
Cycles:	1
Example	
	To maintain upward compatibility with future PICmicro® products, do not use this instruction.

XORLW	**Exclusive OR Literal with W**
Syntax:	[*label*] XORLW k
Operands:	$0 \le k \le 255$
Operation:	(W) .XOR. k → (W)
Status Affected:	Z
Encoding:	`11` `1010` `kkkk` `kkkk`
Description:	The contents of the W register are XOR'ed with the eight bit literal 'k'. The result is placed in the W register.
Words:	1
Cycles:	1
Example:	`XORLW 0xAF` Before Instruction W = 0xB5 After Instruction W = 0x1A

XORWF	**Exclusive OR W with f**
Syntax:	[*label*] XORWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(W) .XOR. (f) → (dest)
Status Affected:	Z
Encoding:	`00` `0110` `dfff` `ffff`
Description:	Exclusive OR the contents of the W register with register 'f'. If 'd' is 0, the result is stored in the W register. If 'd' is 1, the result is stored back in register 'f'.
Words:	1
Cycles:	1
Example	`XORWF REG 1` Before Instruction REG = 0xAF W = 0xB5 After Instruction REG = 0x1A W = 0xB5

12.0 DEVELOPMENT SUPPORT

The PICmicro® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM Assembler
  - MPLAB-C17 and MPLAB-C18 C Compilers
  - MPLINK/MPLIB Linker/Librarian
- Simulators
  - MPLAB-SIM Software Simulator
- Emulators
  - MPLAB-ICE Real-Time In-Circuit Emulator
  - PICMASTER®/PICMASTER-CE In-Circuit Emulator
  - ICEPIC™
- In-Circuit Debugger
  - MPLAB-ICD for PIC16F877
- Device Programmers
  - PRO MATE® II Universal Programmer
  - PICSTART® Plus Entry-Level Prototype Programmer
- Low-Cost Demonstration Boards
  - SIMICE
  - PICDEM-1
  - PICDEM-2
  - PICDEM-3
  - PICDEM-17
  - SEEVAL®
  - KEELOQ®

12.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8-bit microcontroller market. MPLAB is a Windows®-based application which contains:

- Multiple functionality
  - editor
  - simulator
  - programmer (sold separately)
  - emulator (sold separately)
- A full featured editor
- A project manager
- Customizable tool bar and key mapping
- A status bar
- On-line help

MPLAB allows you to:

- Edit your source files (either assembly or ‘C’)
- One touch assemble (or compile) and download to PICmicro tools (automatically updates all project information)
- Debug using:
  - source files
  - absolute listing file
  - object code

The ability to use MPLAB with Microchip’s simulator, MPLAB-SIM, allows a consistent platform and the ability to easily switch from the cost-effective simulator to the full featured emulator with minimal retraining.

12.2 MPASM Assembler

MPASM is a full featured universal macro assembler for all PICmicro MCU’s. It can produce absolute code directly in the form of HEX files for device programmers, or it can generate relocatable objects for MPLINK.

MPASM has a command line interface and a Windows shell and can be used as a standalone application on a Windows 3.x or greater system. MPASM generates relocatable object files, Intel standard HEX files, MAP files to detail memory usage and symbol reference, an absolute LST file which contains source lines and generated machine code, and a COD file for MPLAB debugging.

MPASM features include:

- MPASM and MPLINK are integrated into MPLAB projects.
- MPASM allows user defined macros to be created for streamlined assembly.
- MPASM allows conditional assembly for multi purpose source files.
- MPASM directives allow complete control over the assembly process.

12.3 MPLAB-C17 and MPLAB-C18 C Compilers

The MPLAB-C17 and MPLAB-C18 Code Development Systems are complete ANSI ‘C’ compilers and integrated development environments for Microchip’s PIC17CXXX and PIC18CXXX family of microcontrollers, respectively. These compilers provide powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compilers provide symbol information that is compatible with the MPLAB IDE memory display.

12.4 MPLINK/MPLIB Linker/Librarian

MPLINK is a relocatable linker for MPASM and MPLAB-C17 and MPLAB-C18. It can link relocatable objects from assembly or C source files along with precompiled libraries using directives from a linker script.

MPLIB is a librarian for pre-compiled code to be used with MPLINK. When a routine from a library is called from another source file, only the modules that contains that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications. MPLIB manages the creation and modification of library files.

MPLINK features include:

- MPLINK works with MPASM and MPLAB-C17 and MPLAB-C18.
- MPLINK allows all memory areas to be defined as sections to provide link-time flexibility.

MPLIB features include:

- MPLIB makes linking easier because single libraries can be included instead of many smaller files.
- MPLIB helps keep code maintainable by grouping related modules together.
- MPLIB commands allow libraries to be created and modules to be added, listed, replaced, deleted, or extracted.

12.5 MPLAB-SIM Software Simulator

The MPLAB-SIM Software Simulator allows code development in a PC host environment by simulating the PICmicro series microcontrollers on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a file or user-defined key press to any of the pins. The execution can be performed in single step, execute until break, or trace mode.

MPLAB-SIM fully supports symbolic debugging using MPLAB-C17 and MPLAB-C18 and MPASM. The Software Simulator offers the flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

12.6 MPLAB-ICE High Performance Universal In-Circuit Emulator with MPLAB IDE

The MPLAB-ICE Universal In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PICmicro microcontrollers (MCUs). Software control of MPLAB-ICE is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, “make” and download, and source debugging from a single environment.

Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The universal architecture of the MPLAB-ICE allows expansion to support new PICmicro microcontrollers.

The MPLAB-ICE Emulator System has been designed as a real-time emulation system with advanced features that are generally found on more expensive development tools. The PC platform and Microsoft® Windows 3.x/95/98 environment were chosen to best make these features available to you, the end user.

MPLAB-ICE 2000 is a full-featured emulator system with enhanced trace, trigger, and data monitoring features. Both systems use the same processor modules and will operate across the full operating speed range of the PICmicro MCU.

12.7 PICMASTER/PICMASTER CE

The PICMASTER system from Microchip Technology is a full-featured, professional quality emulator system. This flexible in-circuit emulator provides a high-quality, universal platform for emulating Microchip 8-bit PICmicro microcontrollers (MCUs). PICMASTER systems are sold worldwide, with a CE compliant model available for European Union (EU) countries.

12.8 ICEPIC

ICEPIC is a low-cost in-circuit emulation solution for the Microchip Technology PIC16C5X, PIC16C6X, PIC16C7X, and PIC16CXXX families of 8-bit one-time-programmable (OTP) microcontrollers. The modular system can support different subsets of PIC16C5X or PIC16CXXX products through the use of interchangeable personality modules or daughter boards. The emulator is capable of emulating without target application circuitry being present.

12.9 MPLAB-ICD In-Circuit Debugger

Microchip’s In-Circuit Debugger, MPLAB-ICD, is a powerful, low-cost run-time development tool. This tool is based on the flash PIC16F877 and can be used to develop for this and other PICmicro microcontrollers from the PIC16CXXX family. MPLAB-ICD utilizes the In-Circuit Debugging capability built into the PIC16F87X. This feature, along with Microchip’s In-Circuit Serial Programming protocol, offers cost-effective in-circuit flash programming and debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by watching variables, single-stepping and setting break points. Running at full speed enables testing hardware in real-time. The MPLAB-ICD is also a programmer for the flash PIC16F87X family.

12.10 PRO MATE II Universal Programmer

The PRO MATE II Universal Programmer is a full-featured programmer capable of operating in stand-alone mode as well as PC-hosted mode. PRO MATE II is CE compliant.

The PRO MATE II has programmable V_{DD} and V_{PP} supplies which allows it to verify programmed memory at V_{DD} min and V_{DD} max for maximum reliability. It has an LCD display for instructions and error messages, keys to enter commands and a modular detachable socket assembly to support various package types. In stand-alone mode the PRO MATE II can read, verify or program PICmicro devices. It can also set code-protect bits in this mode.

12.11 PICSTART Plus Entry Level Development System

The PICSTART programmer is an easy-to-use, low-cost prototype programmer. It connects to the PC via one of the COM (RS-232) ports. MPLAB Integrated Development Environment software makes using the programmer simple and efficient.

PICSTART Plus supports all PICmicro devices with up to 40 pins. Larger pin count devices such as the PIC16C92X, and PIC17C76X may be supported with an adapter socket. PICSTART Plus is CE compliant.

12.12 SIMICE Entry-Level Hardware Simulator

SIMICE is an entry-level hardware development system designed to operate in a PC-based environment with Microchip's simulator MPLAB-SIM. Both SIMICE and MPLAB-SIM run under Microchip Technology's MPLAB Integrated Development Environment (IDE) software. Specifically, SIMICE provides hardware simulation for Microchip's PIC12C5XX, PIC12CE5XX, and PIC16C5X families of PICmicro 8-bit microcontrollers. SIMICE works in conjunction with MPLAB-SIM to provide non-real-time I/O port emulation. SIMICE enables a developer to run simulator code for driving the target system. In addition, the target system can provide input to the simulator code. This capability allows for simple and interactive debugging without having to manually generate MPLAB-SIM stimulus files. SIMICE is a valuable debugging tool for entry-level system development.

12.13 PICDEM-1 Low-Cost PICmicro Demonstration Board

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The users can program the sample microcontrollers provided with the PICDEM-1 board, on a PRO MATE II or PICSTART-Plus programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the MPLAB-ICE emulator and download the firmware to the emulator for testing. Additional prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push-button switches and eight LEDs connected to PORTB.

12.14 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE II programmer or PICSTART-Plus, and easily test firmware. The MPLAB-ICE emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I^2C bus and separate headers for connection to an LCD module and a keypad.

12.15 PICDEM-3 Low-Cost PIC16CXXX Demonstration Board

The PICDEM-3 is a simple demonstration board that supports the PIC16C923 and PIC16C924 in the PLCC package. It will also support future 44-pin PLCC microcontrollers with a LCD Module. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-3 board, on a PRO MATE II programmer or PICSTART Plus with an adapter socket, and easily test firmware. The MPLAB-ICE emulator may also be used with the PICDEM-3 board to test firmware. Additional prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features include an RS-232 interface, push-button switches, a potentiometer for simulated analog input, a thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM-3 board is an LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM-3 provides an additional RS-232 interface and Windows 3.1 software for showing the demultiplexed LCD signals on a PC. A simple serial interface allows the user to construct a hardware demultiplexer for the LCD signals.

12.16 PICDEM-17

The PICDEM-17 is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756, PIC17C762, and PIC17C766. All necessary hardware is included to run basic demo programs, which are supplied on a 3.5-inch disk. A programmed sample is included, and the user may erase it and program it with the other sample programs using the PRO MATE II or PICSTART Plus device programmers and easily debug and test the sample code. In addition, PICDEM-17 supports down-loading of programs to and executing out of external FLASH memory on board. The PICDEM-17 is also usable with the MPLAB-ICE or PICMASTER emulator, and all of the sample programs can be run and modified using either emulator. Additionally, a generous prototype area is available for user hardware.

12.17 SEEVAL Evaluation and Programming System

The SEEVAL SEEPROM Designer's Kit supports all Microchip 2-wire and 3-wire Serial EEPROMs. The kit includes everything necessary to read, write, erase or program special features of any Microchip SEEPROM product including Smart Serials™ and secure serials. The Total Endurance™ Disk is included to aid in trade-off analysis and reliability calculations. The total kit can significantly reduce time-to-market and result in an optimized system.

12.18 KEELOQ Evaluation and Programming Tools

KEELOQ evaluation and programming tools support Microchips HCS Secure Data Products. The HCS evaluation kit includes an LCD display to show changing codes, a decoder to decode transmissions, and a programming interface to program test transmitters.

TABLE 12-1: DEVELOPMENT TOOLS FROM MICROCHIP

		PIC12CXXX	PIC14000	PIC16C5X	PIC16C6X	PIC16CXXX	PIC16F62X	PIC16C7X	PIC16C7XX	PIC16C8X	PIC16F8XX	PIC16C9XX	PIC17C4X	PIC17C7XX	PIC18CXX2	24CXX/ 25CXX/ 93CXX	HCSXXX	MCRFXXX	MCP2510
Software Tools	**MPLAB® Integrated Development Environment**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				
	MPLAB® C17 Compiler												✓	✓					
	MPLAB® C18 Compiler														✓				
	MPASM/MPLINK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
Emulators	**MPLAB®-ICE**	✓	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓				
	PICMASTER/PICMASTER-CE	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	✓					
	ICEPIC™ Low-Cost In-Circuit Emulator	✓		✓	✓	✓		✓	✓	✓		✓							
Debugger	**MPLAB®-ICD In-Circuit Debugger**				✓*			✓*			✓								
Programmers	**PICSTART®Plus Low-Cost Universal Dev. Kit**	✓	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓				
	PRO MATE® II Universal Programmer	✓	✓	✓	✓	✓	✓**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
Demo Boards and Eval Kits	**SIMICE**	✓		✓															
	PICDEM-1			✓		✓		✓†		✓			✓						
	PICDEM-2				✓†			✓†							✓				
	PICDEM-3											✓							
	PICDEM-14A		✓																
	PICDEM-17													✓					
	KEELOQ® Evaluation Kit																✓		
	KEELOQ Transponder Kit																✓		
	microID™ Programmer's Kit																	✓	
	125 kHz microID Developer's Kit																	✓	
	125 kHz Anticollision microID Developer's Kit																	✓	
	13.56 MHz Anticollision microID Developer's Kit																	✓	
	MCP2510 CAN Developer's Kit																		✓

* Contact the Microchip Technology Inc. web site at www.microchip.com for information on how to use the MPLAB®-ICD In-Circuit Debugger (DV164001) with PIC16C62, 63, 64, 65, 72, 73, 74, 76, 77

** Contact Microchip Technology Inc. for availability date.

† Development tool is available on select devices.

NOTES:

13.0 ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings †

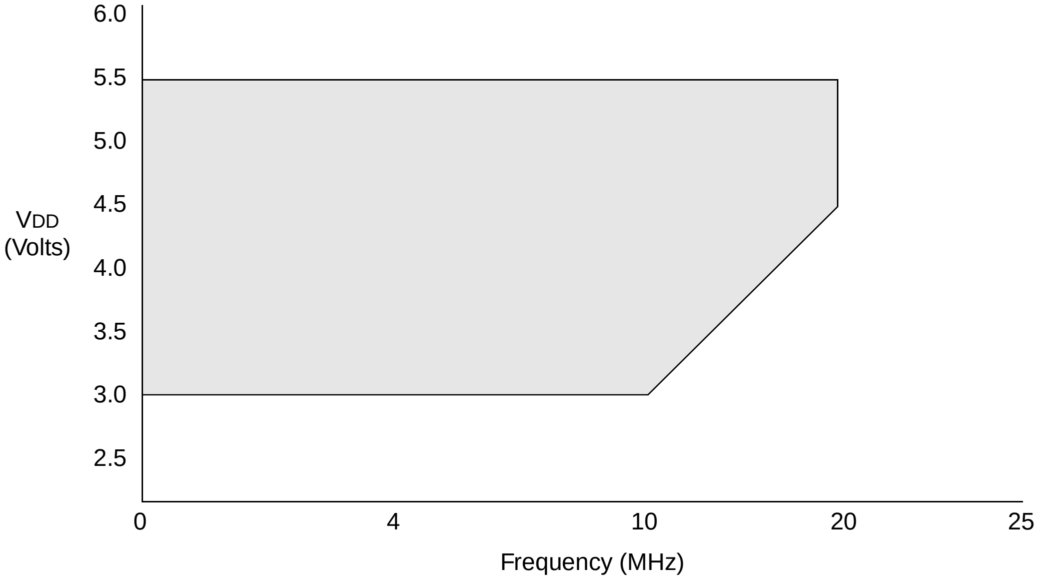
Ambient Temperature under bias	-40° to +125°C
Storage Temperature	-65° to +150°C
Voltage on any pin with respect to V_{SS} (except V_{DD} and $\overline{MCLR}$)	-0.6V to V_{DD} +0.6V
Voltage on V_{DD} with respect to V_{SS}	0 to +7.0V
Voltage on RA4 with respect to V_{SS}	8.5V
Voltage on $\overline{MCLR}$ with respect to V_{SS} (Note 2)	0 to +14V
Voltage on RA4 with respect to V_{SS}	8.5V
Total power Dissipation (Note 1)	1.0W
Maximum Current out of V_{SS} pin	300 mA
Maximum Current into V_{DD} pin	250 mA
Input Clamp Current, I_{IK} (V_I <0 or V_I> V_{DD})	±20 mA
Output Clamp Current, I_{OK} (V_O <0 or V_O>V_{DD})	±20 mA
Maximum Output Current sunk by any I/O pin	25 mA
Maximum Output Current sourced by any I/O pin	25 mA
Maximum Current sunk by PORTA and PORTB	200 mA
Maximum Current sourced by PORTA and PORTB	200 mA

Note 1: Power dissipation is calculated as follows: $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD}-V_{OH}) \times I_{OH}\} + \sum(V_{Ol} \times I_{OL})$

2: Voltage spikes below V_{SS} at the $\overline{MCLR}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100¾ should be used when applying a "low" level to the $\overline{MCLR}$ pin rather than pulling this pin directly to V_{SS}.

† **NOTICE**: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

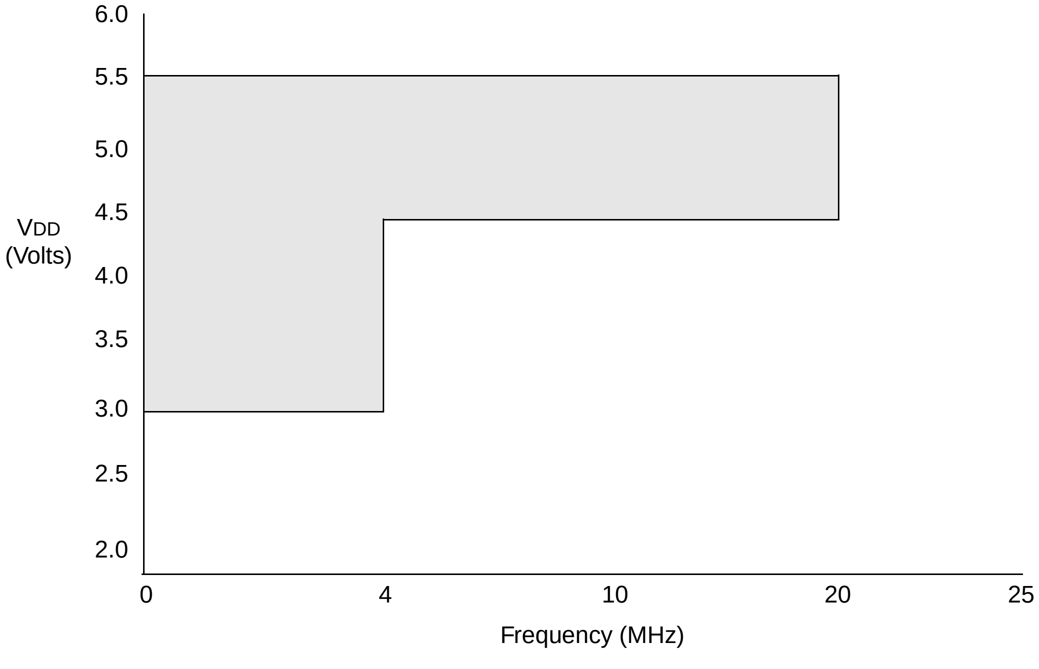
FIGURE 13-1: PIC16CE62X VOLTAGE-FREQUENCY GRAPH, 0°C ≤ T_A ≤ +70°C



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

2: The maximum rated speed of the part limits the permissible combinations of voltage and frequency. Please reference the Product Identification System section for the maximum rated speed of the parts.

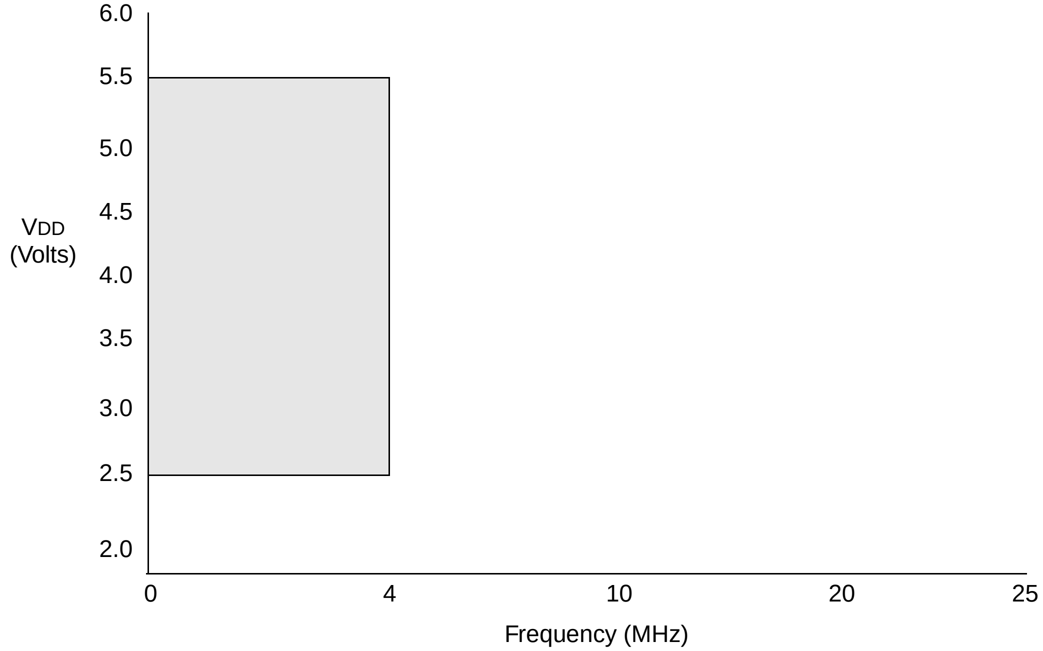
FIGURE 13-2: PIC16CE62X VOLTAGE-FREQUENCY GRAPH, -40°C ≤ T_A < 0°C, +70°C < T_A ≤ +125°C



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

2: The maximum rated speed of the part limits the permissible combinations of voltage and frequency. Please reference the Product Identification System section for the maximum rated speed of the parts.

FIGURE 13-3: PIC16LCE62X VOLTAGE-FREQUENCY GRAPH, -40°C $\leq$ T_A $<$ +125°C



Note 1: The shaded region indicates the permissible combinations of voltage and frequency.

2: The maximum rated speed of the part limits the permissible combinations of voltage and frequency. Please reference the Product Identification System section for the maximum rated speed of the parts.

13.1 DC CHARACTERISTICS: PIC16CE62X-04 (Commercial, Industrial, Extended)
PIC16CE62X-20 (Commercial, Industrial, Extended)

DC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise stated)** Operating temperature –40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial and –40°C ≤ TA ≤ +125°C for extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
D001	VDD	Supply Voltage	3.0	–	5.5	V	See Figure 13-1 through Figure 13-3
D002	VDR	RAM Data Retention Voltage (Note 1)	–	1.5*	–	V	Device in SLEEP mode
D003	VPOR	VDD start voltage to ensure Power-on Reset	–	VSS	–	V	See section on power-on reset for details
D004	SVDD	VDD rise rate to ensure Power-on Reset	0.05*	–	–	V/ms	See section on power-on reset for details
D005	VBOR	Brown-out Detect Voltage	3.7	4.0	4.35	V	BOREN configuration bit is cleared
D010	IDD	Supply Current (Note 2, 4)	–	1.2	2.0	mA	FOSC = 4 MHz, VDD = 5.5V, WDT disabled, XT osc mode, (Note 4)*
			–	0.4	1.2	mA	FOSC = 4 MHz, VDD = 3.0V, WDT disabled, XT osc mode, (Note 4)
			–	1.0	2.0	mA	FOSC = 10 MHz, VDD = 3.0V, WDT disabled, HS osc mode, (Note 6)
			–	4.0	6.0	mA	FOSC = 20 MHz, VDD = 4.5V, WDT disabled, HS osc mode
			–	4.0	7.0	mA	FOSC = 20 MHz, VDD = 5.5V, WDT disabled*, HS osc mode
			–	35	70	µA	FOSC = 32 kHz, VDD = 3.0V, WDT disabled, LP osc mode
D020	IPD	Power Down Current (Note 3)	–	–	2.2	µA	VDD = 3.0V
			–	–	5.0	µA	VDD = 4.5V*
			–	–	9.0	µA	VDD = 5.5V
			–	–	15	µA	VDD = 5.5V Extended
D022	ΔIWDT	WDT Current (Note 5)	–	6.0	10	µA	VDD = 4.0V
					12	µA	(125°C)
D022A	ΔIBOR	Brown-out Reset Current (Note 5)	–	75	125	µA	$\overline{\text{BOD}}$ enabled, VDD = 5.0V
D023	ΔICOMP	Comparator Current for each Comparator (Note 5)	–	30	60	µA	VDD = 4.0V
D023A	ΔIVREF	VREF Current (Note 5)	–	80	135	µA	VDD = 4.0V
	ΔIEE Write	Operating Current	–		3	mA	VCC = 5.5V, SCL = 400 kHz
	ΔIEE Read	Operating Current	–		1	mA	
	ΔIEE	Standby Current	–		30	µA	VCC = 3.0V, EE VDD = VCC
	ΔIEE	Standby Current	–		100	µA	VCC = 3.0V, EE VDD = VCC
1A	FOSC	LP Oscillator Operating Frequency	0	–	200	kHz	All temperatures
		RC Oscillator Operating Frequency	0	–	4	MHz	All temperatures
		XT Oscillator Operating Frequency	0	–	4	MHz	All temperatures
		HS Oscillator Operating Frequency	0	–	20	MHz	All temperatures

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C, unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail to rail; all I/O pins tri-stated, pulled to VDD,
$\overline{\text{MCLR}}$ = VDD; WDT enabled/disabled as specified.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD or VSS.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kΩ.

5: The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

6: Commercial temperature range only.

13.2 DC CHARACTERISTICS: PIC16LCE62X-04 (Commercial, Industrial)

DC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise stated)** Operating temperature –40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial and –40°C ≤ TA ≤ +125°C for extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
D001	VDD	Supply Voltage	2.5	–	5.5	V	See Figure 13-1 through Figure 13-3
D002	VDR	RAM Data Retention Voltage (Note 1)	–	1.5*	–	V	Device in SLEEP mode
D003	VPOR	VDD start voltage to ensure Power-on Reset	–	VSS	–	V	See section on power-on reset for details
D004	SVDD	VDD rise rate to ensure Power-on Reset	.05*	–	–	V/ms	See section on power-on reset for details
D005	VBOR	Brown-out Detect Voltage	3.7	4.0	4.35	V	BOREN configuration bit is cleared
D010	IDD	Supply Current (Note 2)	–	1.2	2.0	mA	FOSC = 4 MHz, VDD = 5.5V, WDT disabled, XT osc mode, (Note 4)*
			–	–	1.1	mA	FOSC = 4 MHz, VDD = 2.5V, WDT disabled, XT osc mode, (Note 4)
			–	35	70	µA	FOSC = 32 kHz, VDD = 2.5V, WDT disabled, LP osc mode
D020	IPD	Power Down Current (Note 3)	–	–	2.0	µA	VDD = 2.5V
			–	–	2.2	µA	VDD = 3.0V*
			–	–	9.0	µA	VDD = 5.5V
			–	–	15	µA	VDD = 5.5V Extended
D022	ΔIWDT	WDT Current (Note 5)	–	6.0	10	µA	VDD=4.0V
					12	µA	(125°C)
D022A	ΔIBOR	Brown-out Reset Current (Note 5)	–	75	125	µA	$\overline{\text{BOD}}$ enabled, VDD = 5.0V
D023	ΔICOMP	Comparator Current for each Comparator (Note 5)	–	30	60	µA	VDD = 4.0V
D023A	ΔIVREF	VREF Current (Note 5)	–	80	135	µA	VDD = 4.0V
	ΔIEE Write	Operating Current	–		3	mA	VCC = 5.5V, SCL = 400 kHz
	ΔIEE Read	Operating Current	–		1	mA	
	ΔIEE	Standby Current	–		30	µA	VCC = 3.0V, EE VDD = VCC
	ΔIEE	Standby Current	–		100	µA	VCC = 3.0V, EE VDD = VCC
1A	FOSC	LP Oscillator Operating Frequency	0	—	200	kHz	All temperatures
		RC Oscillator Operating Frequency	0	—	4	MHz	All temperatures
		XT Oscillator Operating Frequency	0	—	4	MHz	All temperatures
		HS Oscillator Operating Frequency	0	—	20	MHz	All temperatures

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C, unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2: The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail to rail; all I/O pins tri-stated, pulled to VDD,
$\overline{\text{MCLR}}$ = VDD; WDT enabled/disabled as specified.

3: The power down current in SLEEP mode does not depend on the oscillator type. Power down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD or VSS.

4: For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kΩ.

5: The Δ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

6: Commercial temperature range only.

13.3 DC CHARACTERISTICS: PIC16CE62X-04 (Commercial, Industrial, Extended)
PIC16CE62X-20 (Commercial, Industrial, Extended)
PIC16LCE62X (Commercial, Industrial)

DC CHARACTERISTICS			**Standard Operating Conditions (unless otherwise stated)** Operating temperature –40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial and –40°C ≤ TA ≤ +125°C for extended Operating voltage VDD range as described in DC spec Table 13-1				
Parm No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Unit**	**Conditions**
	VIL	**Input Low Voltage**					
		I/O ports					
D030		with TTL buffer	VSS	–	0.8V 0.15VDD	V	VDD = 4.5V to 5.5V, Otherwise
D031		with Schmitt Trigger input	VSS		0.2VDD	V	
D032		$\overline{\text{MCLR}}$, RA4/T0CKI,OSC1 (in RC mode)	VSS	–	0.2VDD	V	Note1
D033		OSC1 (in XT and HS)	VSS	–	0.3VDD	V	
		OSC1 (in LP)	VSS	–	0.6VDD - 1.0	V	
	VIH	**Input High Voltage**					
		I/O ports					
D040		with TTL buffer	2.0V .25VDD + 0.8V	–	VDD VDD	V	VDD = 4.5V to 5.5V, Otherwise
D041		with Schmitt Trigger input	0.8VDD		VDD		
D042		$\overline{\text{MCLR}}$ RA4/T0CKI	0.8VDD	–	VDD	V	
D043		OSC1 (XT, HS and LP)	0.7VDD	–	VDD	V	
D043A		OSC1 (in RC mode)	0.9VDD				Note1
D070	IPURB	PORTB weak pull-up current	50	200	400	µA	VDD = 5.0V, VPIN = VSS
		Input Leakage Current					
	IIL	(Notes 2, 3)					
		I/O ports (Except PORTA)			±1.0	µA	VSS ≤ VPIN ≤ VDD, pin at hi-impedance
D060		PORTA	–	–	±0.5	µA	Vss ≤ VPIN ≤ VDD, pin at hi-impedance
D061		RA4/T0CKI	–	–	±1.0	µA	Vss ≤ VPIN ≤ VDD
D063		OSC1, $\overline{\text{MCLR}}$	–	–	±5.0	µA	Vss ≤ VPIN ≤ VDD, XT, HS and LP osc configuration
	VOL	**Output Low Voltage**					
D080		I/O ports	–	–	0.6	V	IOL=8.5 mA, VDD=4.5V, -40° to +85°C
			–	–	0.6	V	IOL=7.0 mA, VDD=4.5V, +125°C
D083		OSC2/CLKOUT (RC only)	–	–	0.6	V	IOL=1.6 mA, VDD=4.5V, -40° to +85°C
			–	–	0.6	V	IOL=1.2 mA, VDD=4.5V, +125°C
	VOH	**Output High Voltage** (Note 3)					
D090		I/O ports (Except RA4)	VDD-0.7	–	–	V	IOH=-3.0 mA, VDD=4.5V, -40° to +85°C
			VDD-0.7	–	–	V	IOH=-2.5 mA, VDD=4.5V, +125°C
D092		OSC2/CLKOUT (RC only)	VDD-0.7	–	–	V	IOH=-1.3 mA, VDD=4.5V, -40° to +85°C
			VDD-0.7	–	–	V	IOH=-1.0 mA, VDD=4.5V, +125°C
*D150	VOD	**Open-Drain High Voltage**			8.5	V	RA4 pin
		Capacitive Loading Specs on Output Pins					
D100	COSC2	OSC2 pin			15	pF	In XT, HS and LP modes when external clock used to drive OSC1.
D101	Cio	All I/O pins/OSC2 (in RC mode)			50	pF	

* These parameters are characterized but not tested.

† Data in “Typ” column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt Trigger input. It is not recommended that the PIC16CE62X be driven with external clock in RC mode.

2: The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as coming out of the pin.

TABLE 13-1: COMPARATOR SPECIFICATIONS

Operating Conditions: VDD range as described in Table 12-1, -40°C<TA<+125°C. .

Param No.	Characteristics	Sym	Min	Typ	Max	Units	Comments
D300	Input offset voltage	VIOFF		± 5.0	± 10	mV	
D301	Input common mode voltage	VICM	0		VDD - 1.5	V	
D302	CMRR	CMRR	+55*			db	
300	Response Time(1)	TRESP		150*	400*	ns	PIC16CE62X
301	Comparator Mode Change to Output Valid	TMC2OV			10*	µs	

* These parameters are characterized but not tested.

Note 1: Response time measured with one comparator input at (VDD - 1.5)/2 while the other input transitions from VSS to VDD.

TABLE 13-2: VOLTAGE REFERENCE SPECIFICATIONS

Operating Conditions: VDD range as described in Table 12-1, -40°C<TA<+125°C.

Param No.	Characteristics	Sym	Min	Typ	Max	Units	Comments
D310	Resolution	VRES	VDD/24		VDD/32	LSB	
D311	Absolute Accuracy	VRAA			±1/4 ±1/2	LSB LSB	Low Range (VRR=1) High Range (VRR=0)
D312	Unit Resistor Value (R)	VRUR		2K*		Ω	Figure 9-1
310	Settling Time(1)	TSET			10*	µs	

* These parameters are characterized but not tested.

Note 1: Settling time measured while VRR = 1 and VR<3:0> transitions from `0000` to `1111`.

13.4 Timing Parameter Symbology

The timing parameter symbols have been created with one of the following formats:

1. TppS2ppS
2. TppS

T			
F	Frequency	T	Time

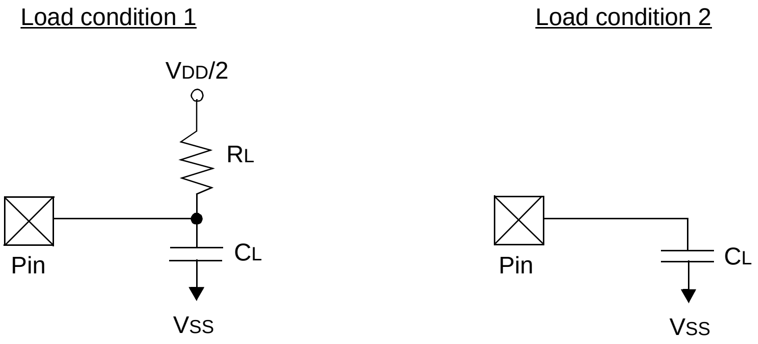
Lowercase subscripts (pp) and their meanings:

pp			
ck	CLKOUT	osc	OSC1
io	I/O port	t0	T0CKI
mc	$\overline{\text{MCLR}}$		

Uppercase letters and their meanings:

S			
F	Fall	P	Period
H	High	R	Rise
I	Invalid (Hi-impedance)	V	Valid
L	Low	Z	Hi-Impedance

FIGURE 13-4: LOAD CONDITIONS


Load condition 1
Load condition 2
VDD/2
RL
Pin
CL
VSS
Pin
CL
VSS

R_L = 464Ω

C_L = 50 pF for all pins except OSC2

15 pF for OSC2 output

13.5 Timing Diagrams and Specifications

FIGURE 13-5: EXTERNAL CLOCK TIMING

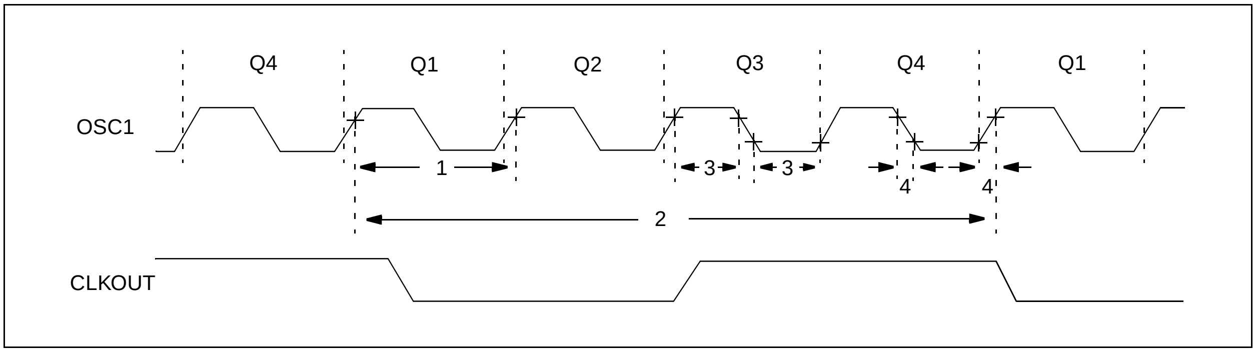


TABLE 13-3: EXTERNAL CLOCK TIMING REQUIREMENTS

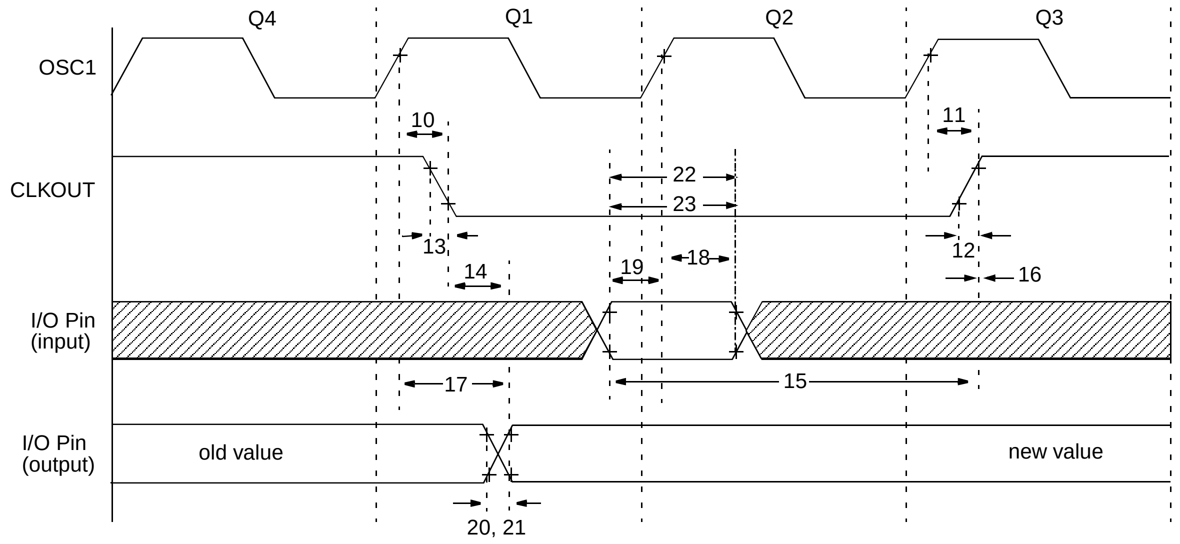
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
1A	Fosc	**External CLKIN Frequency (Note 1)**	DC	—	4	MHz	XT and RC osc mode, VDD=5.0V
			DC	—	20	MHz	HS osc mode
			DC	—	200	kHz	LP osc mode
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode, VDD=5.0V
			0.1	—	4	MHz	XT osc mode
			1	—	20	MHz	HS osc mode
			DC	–	200	kHz	LP osc mode
1	Tosc	**External CLKIN Period (Note 1)**	250	—	—	ns	XT and RC osc mode
			50	—	—	ns	HS osc mode
			5	—	—	µs	LP osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			250	—	10,000	ns	XT osc mode
			50	—	1,000	ns	HS osc mode
			5	—	—	µs	LP osc mode
2	TCY	**Instruction Cycle Time (Note 1)**	200	—	DC	ns	TCY=FOSC/4
3*	TosL, TosH	External Clock in (OSC1) High or Low Time	100*	—	—	ns	XT oscillator, TOSC L/H duty cycle
			2*	—	—	µs	LP oscillator, TOSC L/H duty cycle
			20*	—	—	ns	HS oscillator, TOSC L/H duty cycle
4*	TosR, TosF	External Clock in (OSC1) Rise or Fall Time	25*	—	—	ns	XT oscillator
			50*	—	—	ns	LP oscillator
			15*	—	—	ns	HS oscillator

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Instruction cycle period (TCY) equals four times the input oscillator time-base period. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at "min." values with an external clock applied to the OSC1 pin.
When an external clock input is used, the "Max." cycle time limit is "DC" (no clock) for all devices.

FIGURE 13-6: CLKOUT AND I/O TIMING



Note: All tests must be do with specified capacitance loads (Figure 13-4) 50 pF on I/O pins and CLKOUT

TABLE 13-4: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter #	Sym	Characteristic	Min	Typ†	Max	Units
10*	TosH2ckL	OSC1↑ to CLKOUT↓ (1)	—	75	200	ns
11*	TosH2ckH	OSC1↑ to CLKOUT↑ (1)	—	75	200	ns
12*	TckR	CLKOUT rise time (1)	—	35	100	ns
13*	TckF	CLKOUT fall time (1)	—	35	100	ns
14*	TckL2ioV	CLKOUT ↓ to Port out valid (1)	—	—	20	ns
15*	TioV2ckH	Port in valid before CLKOUT ↑ (1)	Tosc +200 ns	—	—	ns
16*	TckH2ioI	Port in hold after CLKOUT ↑ (1)	0	—	—	ns
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid	—	50	150	ns
18*	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	100	—	—	ns
19*	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)	0	—	—	ns
20*	TioR	Port output rise time	—	10	40	ns
21*	TioF	Port output fall time	—	10	40	ns
22*	Tinp	RB0/INT pin high or low time	25	—	—	ns
23	Trbp	RB<7:4> change interrupt high or low time	TCY	—	—	ns

* These parameters are characterized but not tested

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x TOSC.

FIGURE 13-7: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING

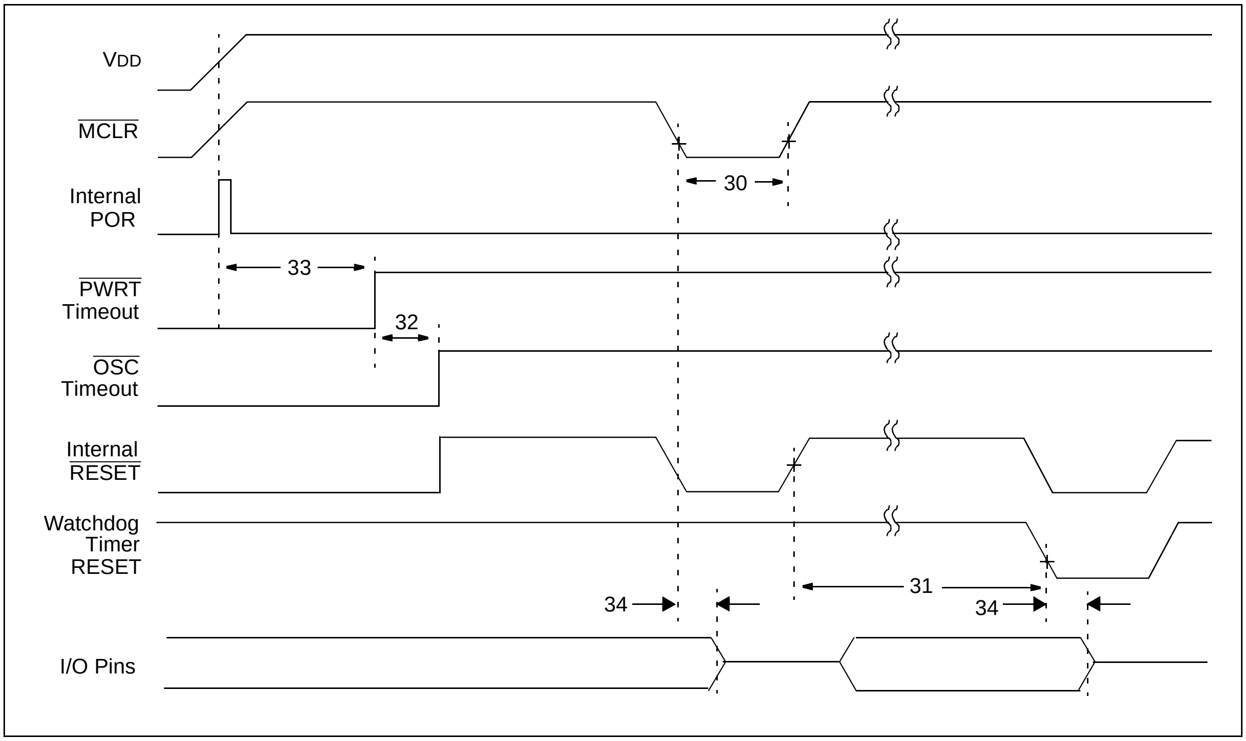

VDD
MCLR
Internal POR
PWRT Timeout
OSC Timeout
Internal RESET
Watchdog Timer RESET
I/O Pins
30
33
32
31
34
34

FIGURE 13-8: BROWN-OUT RESET TIMING

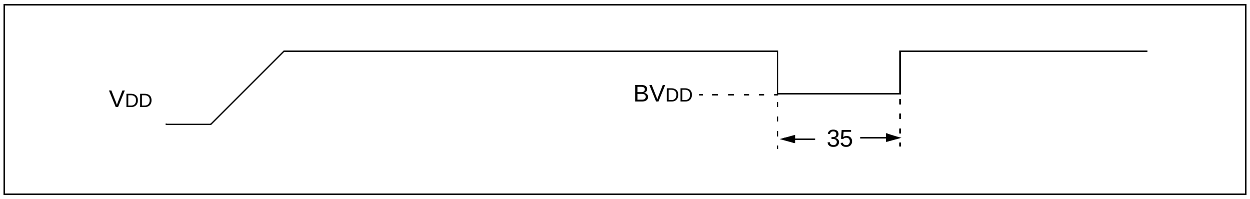


TABLE 13-5: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
30	TmcL	$\overline{\text{MCLR}}$ Pulse Width (low)	2000	—	—	ns	-40° to +85°C
31	Twdt	Watchdog Timer Time-out Period (No Prescaler)	7*	18	33*	ms	VDD = 5.0V, -40° to +85°C
32	Tost	Oscillation Start-up Timer Period	—	1024 TOSC	—	—	TOSC = OSC1 period
33	Tpwrt	Power-up Timer Period	28*	72	132*	ms	VDD = 5.0V, -40° to +85°C
34	TIOZ	I/O hi-impedance from $\overline{\text{MCLR}}$ low		—	2.0	μs	
35	TBOR	Brown-out Reset Pulse Width	100*	—	—	μs	3.7V ≤ VDD ≤ 4.3V

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

FIGURE 13-9: TIMER0 CLOCK TIMING

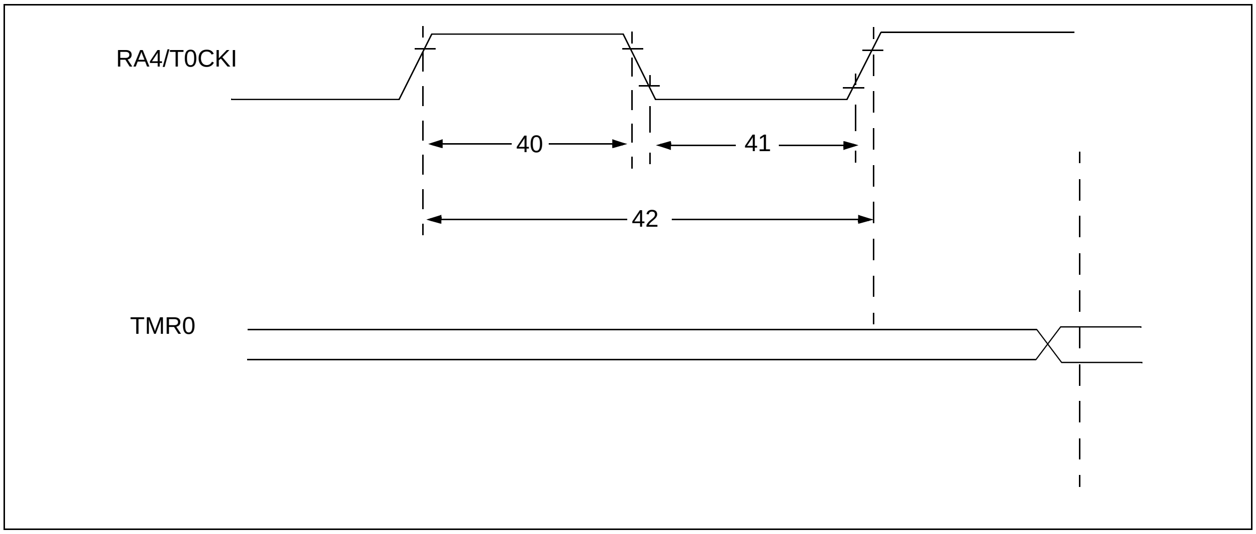


TABLE 13-6: TIMER0 CLOCK REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
40	Tt0H	T0CKI High Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
41	Tt0L	T0CKI Low Pulse Width	No Prescaler	0.5 TCY + 20*	—	—	ns	
			With Prescaler	10*	—	—	ns	
42	Tt0P	T0CKI Period		$\frac{T_{CY} + 40}{N}$*	—	—	ns	N = prescale value (1, 2, 4, ..., 256)

* These parameters are characterized but not tested.

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

13.6 EEPROM Timing

FIGURE 13-10: BUS TIMING DATA

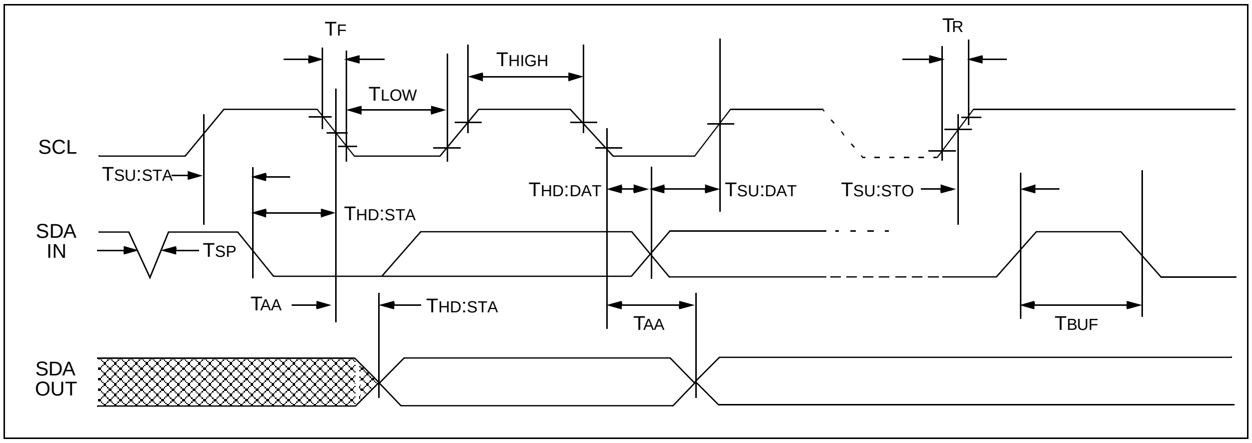


TABLE 13-7: AC CHARACTERISTICS

Parameter	Symbol	STANDARD MODE		Vcc = 4.5 - 5.5V FAST MODE		Units	Remarks
		Min.	Max.	Min.	Max.		
Clock frequency	FCLK	—	100	—	400	kHz	
Clock high time	THIGH	4000	—	600	—	ns	
Clock low time	TLOW	4700	—	1300	—	ns	
SDA and SCL rise time	TR	—	1000	—	300	ns	(Note 1)
SDA and SCL fall time	TF	—	300	—	300	ns	(Note 1)
START condition hold time	THD:STA	4000	—	600	—	ns	After this period the first clock pulse is generated
START condition setup time	TSU:STA	4700	—	600	—	ns	Only relevant for repeated START condition
Data input hold time	THD:DAT	0	—	0	—	ns	(Note 2)
Data input setup time	TSU:DAT	250	—	100	—	ns	
STOP condition setup time	TSU:STO	4000	—	600	—	ns	
Output valid from clock	TAA	—	3500	—	900	ns	(Note 2)
Bus free time	TBUF	4700	—	1300	—	ns	Time the bus must be free before a new transmission can start
Output fall time from VIH minimum to VIL maximum	TOF	—	250	20 + 0.1 CB	250	ns	(Note 1), CB ≤ 100 pF
Input filter spike suppression (SDA and SCL pins)	TSP	—	50	—	50	ns	(Note 3)
Write cycle time	TWR	—	10	—	10	ms	Byte or Page mode
Endurance	—	10M 1M	—	10M 1M	—	cycles	25°C, VCC = 5.0V, Block Mode (Note 4)

Note 1: Not 100% tested. CB = total capacitance of one bus line in pF.

2: As a transmitter, the device must provide an internal minimum delay time to bridge the undefined region (minimum 300 ns) of the falling edge of SCL to avoid unintended generation of START or STOP conditions.

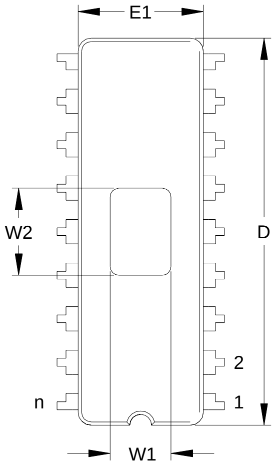
3: The combined TSP and VHYS specifications are due to new Schmitt trigger inputs which provide improved noise spike suppression. This eliminates the need for a TI specification for standard operation.

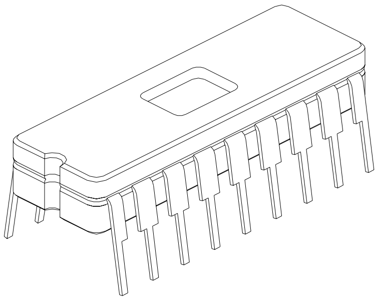
4: This parameter is not tested but guaranteed by characterization. For endurance estimates in a specific application, please consult the Total Endurance Model which can be obtained on our website.

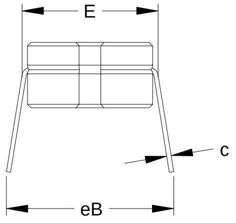
NOTES:

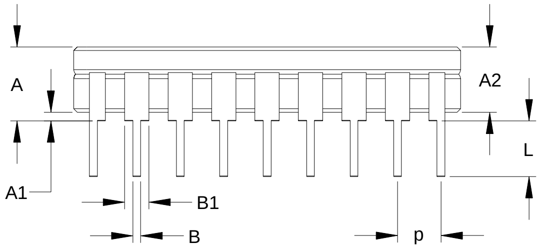
14.0 PACKAGING INFORMATION

18-Lead Ceramic Dual In-line with Window (JW) – 300 mil (CERDIP)





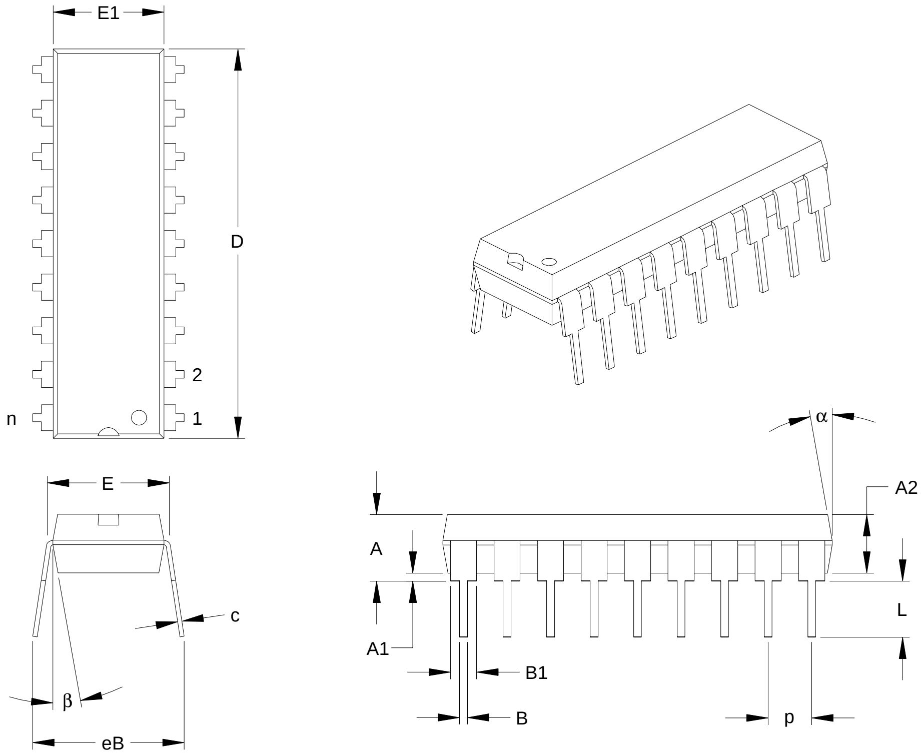




Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.170	.183	.195	4.32	4.64	4.95
Ceramic Package Height	A2	.155	.160	.165	3.94	4.06	4.19
Standoff	A1	.015	.023	.030	0.38	0.57	0.76
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Ceramic Pkg. Width	E1	.285	.290	.295	7.24	7.37	7.49
Overall Length	D	.880	.900	.920	22.35	22.86	23.37
Tip to Seating Plane	L	.125	.138	.150	3.18	3.49	3.81
Lead Thickness	c	.008	.010	.012	0.20	0.25	0.30
Upper Lead Width	B1	.050	.055	.060	1.27	1.40	1.52
Lower Lead Width	B	.016	.019	.021	0.41	0.47	0.53
Overall Row Spacing	eB	.345	.385	.425	8.76	9.78	10.80
Window Width	W1	.130	.140	.150	3.30	3.56	3.81
Window Length	W2	.190	.200	.210	4.83	5.08	5.33

*Controlling Parameter
JEDEC Equivalent: MO-036
Drawing No. C04-010

18-Lead Plastic Dual In-line (P) – 300 mil (PDIP)



	Units	INCHES*			MILLIMETERS		
	Dimension Limits	MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.155	.170	3.56	3.94	4.32
Molded Package Thickness	A2	.115	.130	.145	2.92	3.30	3.68
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.313	.325	7.62	7.94	8.26
Molded Package Width	E1	.240	.250	.260	6.10	6.35	6.60
Overall Length	D	.890	.898	.905	22.61	22.80	22.99
Tip to Seating Plane	L	.125	.130	.135	3.18	3.30	3.43
Lead Thickness	c	.008	.012	.015	0.20	0.29	0.38
Upper Lead Width	B1	.045	.058	.070	1.14	1.46	1.78
Lower Lead Width	B	.014	.018	.022	0.36	0.46	0.56
Overall Row Spacing	eB	.310	.370	.430	7.87	9.40	10.92
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

*Controlling Parameter

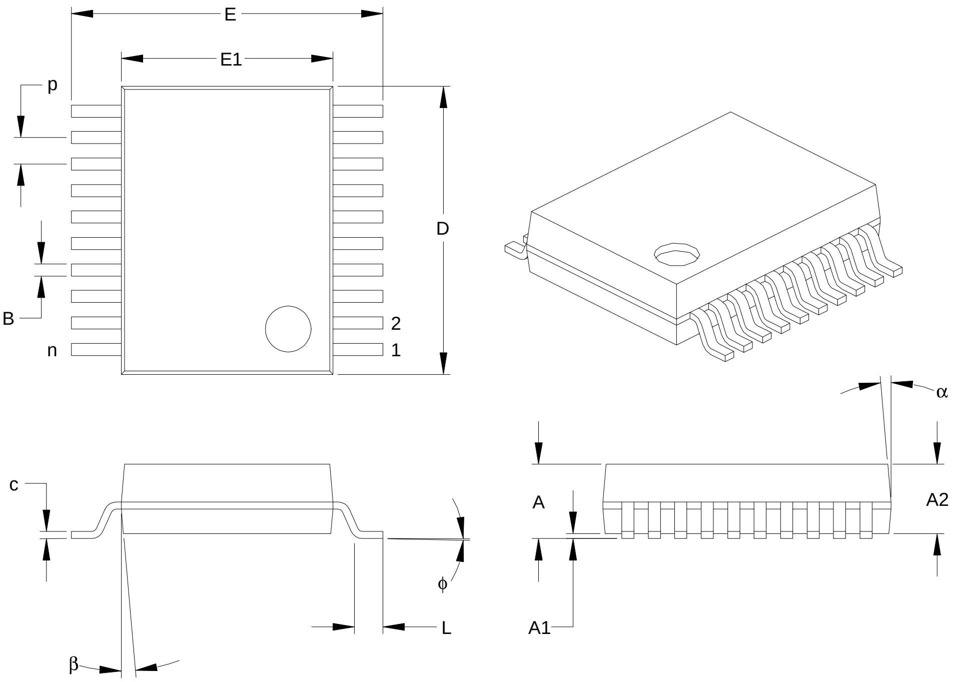
Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.

JEDEC Equivalent: MS-001

Drawing No. C04-007

20-Lead Plastic Shrink Small Outline (SS) – 209 mil, 5.30 mm (SSOP)



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		20			20	
Pitch	p		.026			0.66	
Overall Height	A	.068	.073	.078	1.73	1.85	1.98
Molded Package Thickness	A2	.064	.068	.072	1.63	1.73	1.83
Standoff	A1	.002	.006	.010	0.05	0.15	0.25
Overall Width	E	.299	.309	.322	7.59	7.85	8.18
Molded Package Width	E1	.201	.207	.212	5.11	5.25	5.38
Overall Length	D	.278	.284	.289	7.06	7.20	7.34
Foot Length	L	.022	.030	.037	0.56	0.75	0.94
Lead Thickness	c	.004	.007	.010	0.10	0.18	0.25
Foot Angle	φ	0	4	8	0.00	101.60	203.20
Lead Width	B	.010	.013	.015	0.25	0.32	0.38
Mold Draft Angle Top	α	0	5	10	0	5	10
Mold Draft Angle Bottom	β	0	5	10	0	5	10

*Controlling Parameter

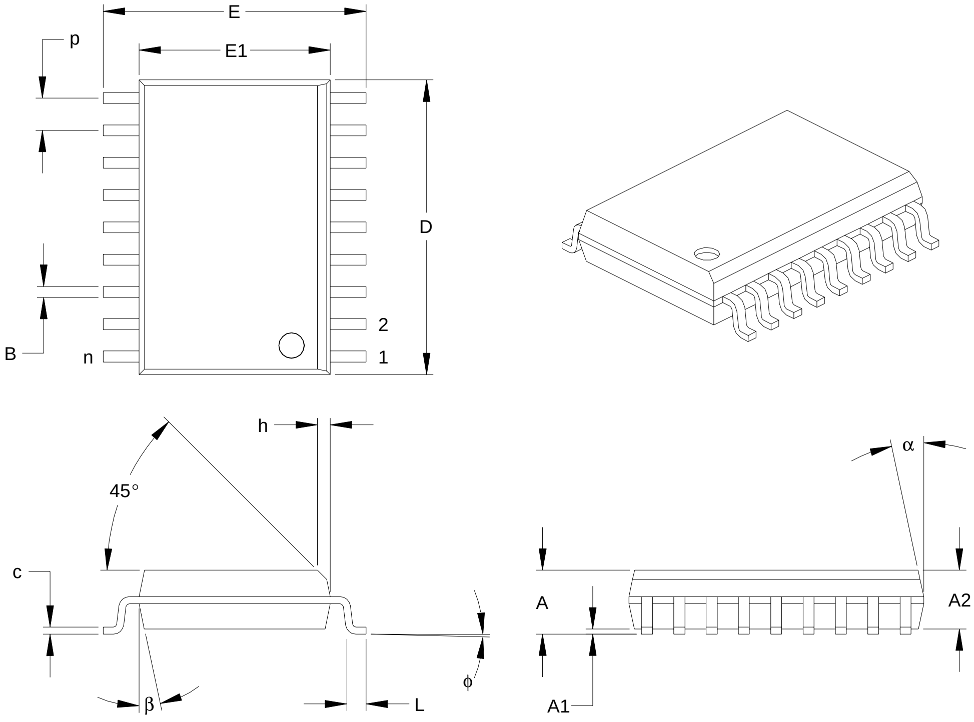
Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.

JEDEC Equivalent: MO-150

Drawing No. C04-072

18-Lead Plastic Small Outline (SO) – Wide, 300 mil (SOIC)



Units		INCHES*			MILLIMETERS		
Dimension Limits		MIN	NOM	MAX	MIN	NOM	MAX
Number of Pins	n		18			18	
Pitch	p		.050			1.27	
Overall Height	A	.093	.099	.104	2.36	2.50	2.64
Molded Package Thickness	A2	.088	.091	.094	2.24	2.31	2.39
Standoff	A1	.004	.008	.012	0.10	0.20	0.30
Overall Width	E	.394	.407	.420	10.01	10.34	10.67
Molded Package Width	E1	.291	.295	.299	7.39	7.49	7.59
Overall Length	D	.446	.454	.462	11.33	11.53	11.73
Chamfer Distance	h	.010	.020	.029	0.25	0.50	0.74
Foot Length	L	.016	.033	.050	0.41	0.84	1.27
Foot Angle	φ	0	4	8	0	4	8
Lead Thickness	c	.009	.011	.012	0.23	0.27	0.30
Lead Width	B	.014	.017	.020	0.36	0.42	0.51
Mold Draft Angle Top	α	0	12	15	0	12	15
Mold Draft Angle Bottom	β	0	12	15	0	12	15

*Controlling Parameter

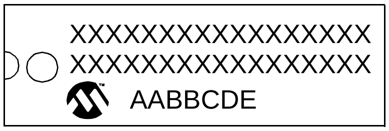
Notes:

Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010” (0.254mm) per side.

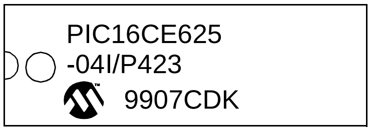
JEDEC Equivalent: MS-013
Drawing No. C04-051

14.1 Package Marking Information

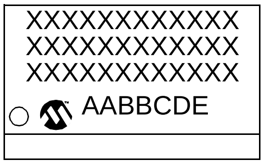
18-Lead PDIP


XXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX
AABBCDE

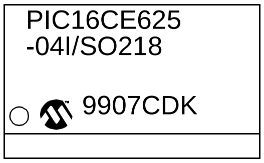
Example


PIC16CE625
-04I/P423
9907CDK

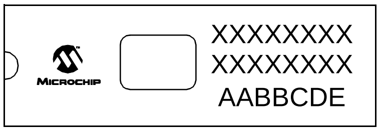
18-Lead SOIC (.300")


XXXXXXXXXXXX
XXXXXXXXXXXX
XXXXXXXXXXXX
AABBCDE

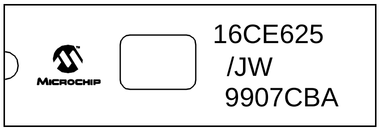
Example


PIC16CE625
-04I/SO218
9907CDK

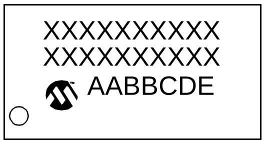
18-Lead CERDIP Windowed


MICROCHIP
XXXXXXXX
XXXXXXXX
AABBCDE

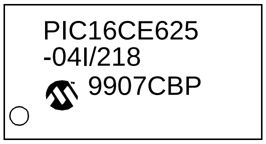
Example


MICROCHIP
16CE625
/JW
9907CBA

20-Lead SSOP


XXXXXXXXXX
XXXXXXXXXX
AABBCDE

Example


PIC16CE625
-04I/218
9907CBP

Legend: MM...M	Microchip part number information
XX...X	Customer specific information*
AA	Year code (last 2 digits of calendar year)
BB	Week code (week of January 1 is week '01')
C	Facility code of the plant at which wafer is manufactured O = Outside Vendor C = 5" Line S = 6" Line H = 8" Line
D	Mask revision number
E	Assembly code of the plant or country of origin in which part was assembled

Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line thus limiting the number of available characters for customer specific information.

* Standard OTP marking consists of Microchip part number, year code, week code, facility code, mask rev#, and assembly code. For OTP marking beyond this, certain price adders apply. Please check with your Microchip Sales Office. For QTP devices, any special marking adders are included in QTP price.

NOTES:

APPENDIX A: CODE FOR ACCESSING EEPROM DATA MEMORY

Please check our web site at www.microchip.com for code availability.

NOTES:

INDEX

A

ADDLW Instruction 67
ADDWF Instruction 67
ANDLW Instruction 67
ANDWF Instruction 67
Architectural Overview 7
Assembler
  MPASM Assembler 77

B

BCF Instruction 68
Block Diagram
  TIMER0 35
  TMR0/WDT PRESCALER 38
Brown-Out Detect (BOD) 54
BSF Instruction 68
BTFSC Instruction 68
BTFSS Instruction 69

C

CALL Instruction 69
Clocking Scheme/Instruction Cycle 10
CLRF Instruction 69
CLRW Instruction 69
CLRWDT Instruction 70
CMCON Register 41
Code Protection 64
COMF Instruction 70
Comparator Configuration 42
Comparator Interrupts 45
Comparator Module 41
Comparator Operation 43
Comparator Reference 43
Configuration Bits 50
Configuring the Voltage Reference 47
Crystal Operation 51

D

Data Memory Organization 12
DECF Instruction 70
DECFSZ Instruction 70
Development Support 77

E

EEPROM Peripheral Operation 29
Errata 2
External Crystal Oscillator Circuit 52

G

General purpose Register File 12
GOTO Instruction 71

I

I/O Ports 23
I/O Programming Considerations 28
ID Locations 64
INCF Instruction 71
INCFSZ Instruction 71
In-Circuit Serial Programming 64
Indirect Addressing, INDF and FSR Registers 21
Instruction Flow/Pipelining 10
Instruction Set
  ADDLW 67
  ADDWF 67
  ANDLW 67
  ANDWF 67
  BCF 68
  BSF 68
  BTFSC 68
  BTFSS 69
  CALL 69
  CLRF 69
  CLRW 69
  CLRWDT 70
  COMF 70
  DECF 70
  DECFSZ 70
  GOTO 71
  INCF 71
  INCFSZ 71
  IORLW 71
  IORWF 72
  MOVF 72
  MOVLW 72
  MOVWF 72
  NOP 73
  OPTION 73
  RETFIE 73
  RETLW 73
  RETURN 74
  RLF 74
  RRF 74
  SLEEP 74
  SUBLW 75
  SUBWF 75
  SWAPF 76
  TRIS 76
  XORLW 76
  XORWF 76
Instruction Set Summary 65
INT Interrupt 60
INTCON Register 17
Interrupts 59
IORLW Instruction 71
IORWF Instruction 72

K

KeeLoq® Evaluation and Programming Tools 80

M

MOVF Instruction 72
MOVLW Instruction 72
MOVWF Instruction 72
MPLAB Integrated Development Environment Software 77

N

NOP Instruction 73

O

One-Time-Programmable (OTP) Devices 5
OPTION Instruction 73
OPTION Register 16
Oscillator Configurations 51
Oscillator Start-up Timer (OST) 54

P

Package Marking Information 101
Packaging Information 97
PCL and PCLATH 20
PCON Register 19
PICDEM-1 Low-Cost PICmicro Demo Board 79
PICDEM-2 Low-Cost PIC16CXX Demo Board 79
PICDEM-3 Low-Cost PIC16CXXX Demo Board 79
PICSTART® Plus Entry Level Development System 79
PIE1 Register 18
Pinout Description 9
PIR1 Register 18

Port RB Interrupt 60
PORTA 23
PORTB 26
Power Control/Status Register (PCON) 55
Power-Down Mode (SLEEP) 63
Power-On Reset (POR) 54
Power-up Timer (PWRT) 54
Prescaler 38
PRO MATE® II Universal Programmer 79
Program Memory Organization 11

Q

Quick-Turnaround-Production (QTP) Devices 5

R

RC Oscillator 52
Reset 53
RETFIE Instruction 73
RETLW Instruction 73
RETURN Instruction 74
RLF Instruction 74
RRF Instruction 74

S

SEEVAL® Evaluation and Programming System 80
Serialized Quick-Turnaround-Production (SQTP) Devices ... 5
SLEEP Instruction 74
Software Simulator (MPLAB-SIM) 78
Special Features of the CPU 49
Special Function Registers 14
Stack 20
Status Register 15
SUBLW Instruction 75
SUBWF Instruction 75
SWAPF Instruction 76

T

Timer0
  TIMER0 35
  TIMER0 (TMR0) Interrupt 35
  TIMER0 (TMR0) Module 35
  TMR0 with External Clock 37
Timer1
  Switching Prescaler Assignment 39
Timing Diagrams and Specifications 91
TMR0 Interrupt 60
TRIS Instruction 76
TRISA 23
TRISB 26

V

Voltage Reference Module 47
VRCON Register 47

W

Watchdog Timer (WDT) 61
WWW, On-Line Support 2

X

XORLW Instruction 76
XORWF Instruction 76

ON-LINE SUPPORT

Microchip provides on-line support on the Microchip World Wide Web (WWW) site.

The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, the user must have access to the Internet and a web browser, such as Netscape or Microsoft Explorer. Files are also available for FTP download from our FTP site.

Connecting to the Microchip Internet Web Site

The Microchip web site is available by using your favorite Internet browser to attach to:

www.microchip.com

The file transfer site is available by using an FTP service to connect to:

ftp://ftp.microchip.com

The web site and file transfer site provide a variety of services. Users may download files for the latest Development Tools, Data Sheets, Application Notes, User's Guides, Articles and Sample Programs. A variety of Microchip specific business information is also available, including listings of Microchip sales offices, distributors and factory representatives. Other data available for consideration is:

- Latest Microchip Press Releases
- Technical Support Section with Frequently Asked Questions
- Design Tips
- Device Errata
- Job Postings
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip Products
- Conferences for products, Development Systems, technical information and more
- Listing of seminars and events

Systems Information and Upgrade Hot Line

The Systems Information and Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits.The Hot Line Numbers are:

1-800-755-2345 for U.S. and most of Canada, and

1-480-786-7302 for the rest of the world.

981103

Trademarks: The Microchip name, logo, PIC, PICmicro, PICSTART, PICMASTER, PRO MATE and MPLAB are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries. *Flex*ROM and *fuzzy*LAB are trademarks and SQTP is a service mark of Microchip in the U.S.A.

All other trademarks mentioned herein are the property of their respective companies.

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 786-7578.

Please list the following information, and use this outline to provide us with your comments about this Data Sheet.

To: Technical Publications Manager Total Pages Sent
RE: Reader Response
From: Name ______________________________
Company ______________________________
Address ______________________________
City / State / ZIP / Country ______________________________
Telephone: (_______) _________ - _________ FAX: (______) _________ - _________

Application (optional):

Would you like a reply? ____Y ____N

Device: **PIC16CE62X** Literature Number: **DS40182C**

Questions:

1. What are the best features of this document?

____

____

2. How does this document meet your hardware and software development needs?

____

____

3. Do you find the organization of this data sheet easy to follow? If not, why?

____

____

4. What additions to the data sheet do you think would enhance the structure and subject?

____

____

5. What deletions from the data sheet could be made without affecting the overall usefulness?

____

____

6. Is there any incorrect or misleading information (what and where)?

____

____

7. How would you improve this document?

____

____

8. How would you improve our software, systems, and silicon products?

____

____

PIC16CE62X PRODUCT IDENTIFICATION SYSTEM

To order or to obtain information, e.g., on pricing or delivery, please use the listed part numbers, and refer to the factory or the listed sales offices.

PART NO. -XX X /XX XXX

Pattern: 3-Digit Pattern Code for QTP (blank otherwise)

Package:
P = PDIP
SO = SOIC (Gull Wing, 300 mil body)
SS = SSOP (209 mil)
JW* = Windowed CERDIP

Temperature Range:
- = 0°C to +70°C
I = –40°C to +85°C
E = –40°C to +125°C

Frequency Range:
04 = 200kHz (LP osc)
04 = 4 MHz (XT and RC osc)
20 = 20 MHz (HS osc)

Device:
PIC16CE62X :VDD range 3.0V to 5.5V
PIC16CE62XT:VDD range 3.0V to 5.5V (Tape and Reel)

Examples:

a) PIC16CE623-04/P301 = Commercial temp., PDIP package, 4 MHz, normal VDD limits, QTP pattern #301.
b) PIC16CE623-04I/SO = Industrial temp., SOIC package, 4MHz, industrial VDD limits.

* JW Devices are UV erasable and can be programmed to any device configuration. JW Devices meet the electrical requirement of each oscillator type.

Sales and Support

Data Sheets
Products supported by a preliminary Data Sheet may have an errata sheet describing minor operational differences and recommended workarounds. To determine if an errata sheet exists for a particular device, please contact one of the following:

1. Your local Microchip sales office
2. The Microchip Corporate Literature Center U.S. FAX: (480) 786-7277
3. The Microchip Worldwide Site (www.microchip.com)

Please specify which device, revision of silicon and Data Sheet (include Literature #) you are using.

New Customer Notification System
Register on our web site (www.microchip.com/cn) to receive the most current information on our products.

NOTES:

NOTES:

Note the following details of the code protection feature on PICmicro® MCUs.

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable”.
- Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip’s products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, FilterLab, KEELOQ, microID, MPLAB, PIC, PICmicro, PICMASTER, PICSTART, PRO MATE, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

dsPIC, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, MXDEV, PICC, PICDEM, PICDEM.net, rfPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2002, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

♻ Printed on recycled paper.

DNV Certification, Inc.
USA

ANSI★RAB
QMS
ACCREDITED

DNV

DNV MSC
The Netherlands
Accredited by the RvA

ISO 9001 / QS-9000
REGISTERED FIRM

Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company’s quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001 certified.

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200 Fax: 480-792-7277
Technical Support: 480-792-7627
Web Address: http://www.microchip.com

Rocky Mountain
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966 Fax: 480-792-7456

Atlanta
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848 Fax: 978-692-3821

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423 Fax: 972-818-2924

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

Kokomo
2767 S. Albright Road
Kokomo, Indiana 46902
Tel: 765-864-8360 Fax: 765-864-8387

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

New York
150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 631-273-5305 Fax: 631-273-5335

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699 Fax: 905-673-6509

ASIA/PACIFIC

Australia
Microchip Technology Australia Pty Ltd
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

China - Beijing
Microchip Technology Consulting (Shanghai)
Co., Ltd., Beijing Liaison Office
Unit 915
Bei Hai Wan Tai Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

China - Chengdu
Microchip Technology Consulting (Shanghai)
Co., Ltd., Chengdu Liaison Office
Rm. 2401, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-6766200 Fax: 86-28-6766599

China - Fuzhou
Microchip Technology Consulting (Shanghai)
Co., Ltd., Fuzhou Liaison Office
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506 Fax: 86-591-7503521

China - Shanghai
Microchip Technology Consulting (Shanghai)
Co., Ltd.
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

China - Shenzhen
Microchip Technology Consulting (Shanghai)
Co., Ltd., Shenzhen Liaison Office
Rm. 1315, 13/F, Shenzhen Kerry Centre,
Renminnan Lu
Shenzhen 518001, China
Tel: 86-755-2350361 Fax: 86-755-2366086

Hong Kong
Microchip Technology Hongkong Ltd.
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

India
Microchip Technology Inc.
India Liaison Office
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

Japan
Microchip Technology Japan K.K.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5934

Singapore
Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-334-8870 Fax: 65-334-8850

Taiwan
Microchip Technology Taiwan
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Denmark
Microchip Technology Nordic ApS
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45 4420 9895 Fax: 45 4420 9910

France
Microchip Technology SARL
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany
Microchip Technology GmbH
Gustav-Heinemann Ring 125
D-81739 Munich, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy
Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-039-65791-1 Fax: 39-039-6899883

United Kingdom
Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5869 Fax: 44-118 921-5820

01/18/02