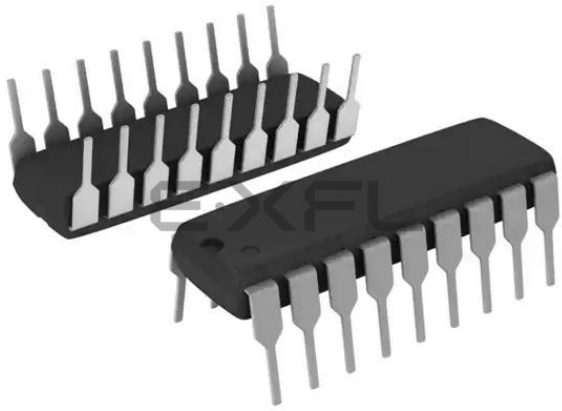


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, POR, WDT
Number of I/O	13
Program Memory Size	896B (512 x 14)
Program Memory Type	OTP
EEPROM Size	128 x 8
RAM Size	96 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	18-DIP (0.300", 7.62mm)
Supplier Device Package	18-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16ce623-04i-p

1.0 GENERAL DESCRIPTION

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

All PIC® 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.

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.

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

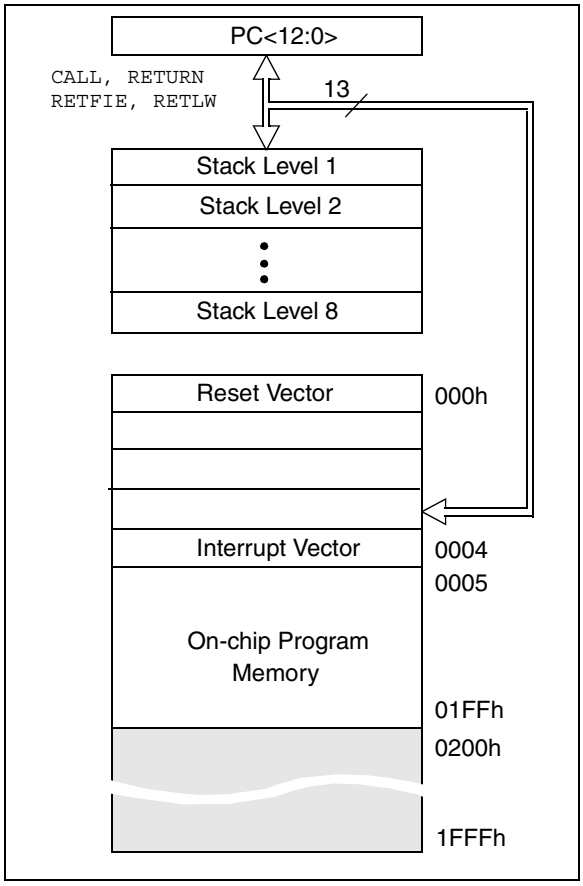


FIGURE 4-2: PROGRAM MEMORY MAP AND STACK FOR THE PIC16CE624

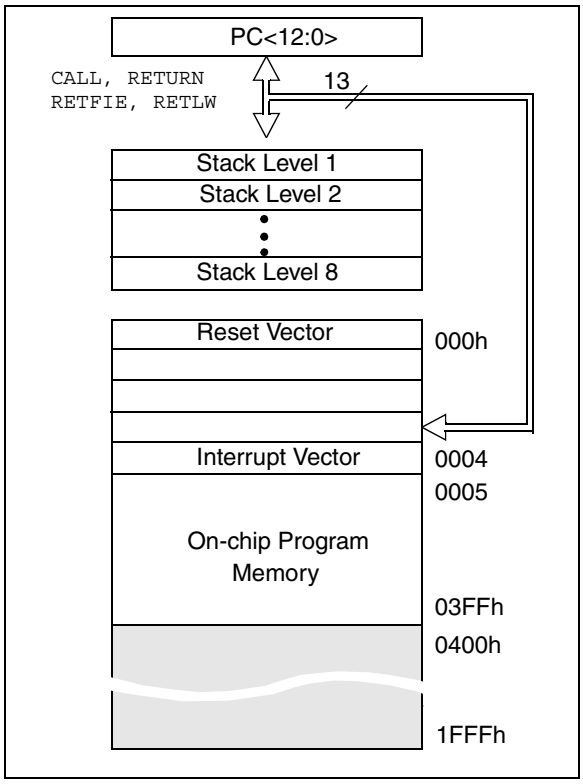
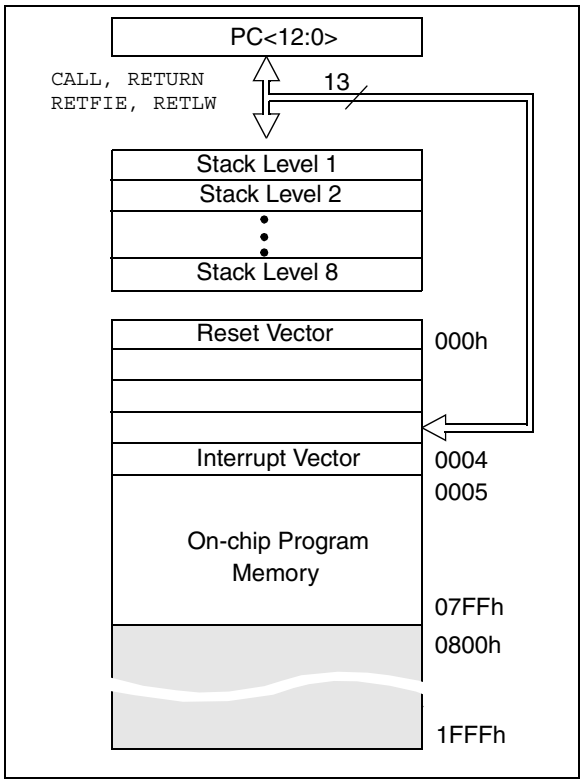


FIGURE 4-3: PROGRAM MEMORY MAP AND STACK FOR THE PIC16CE625



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

4.2.2.6 PCON REGISTER

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

Note: $\overline{BOD}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent resets to see if $\overline{BOD}$ is cleared, indicating a brown-out has occurred. The $\overline{BOD}$ 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
—	—	—	—	—	—	$\overline{POR}$	$\overline{BOD}$
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: **$\overline{POR}$**: 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: **$\overline{BOD}$**: 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)

NOTES:

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.

FIGURE 6-7: CURRENT ADDRESS READ

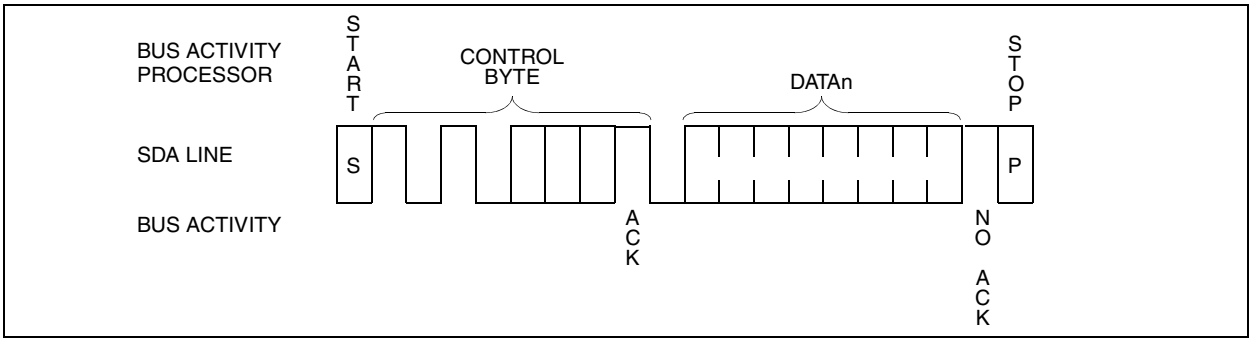


FIGURE 6-8: RANDOM READ

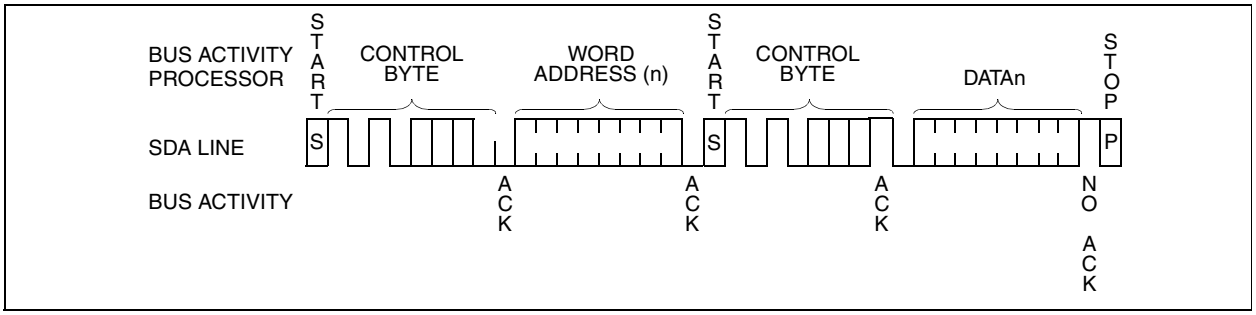
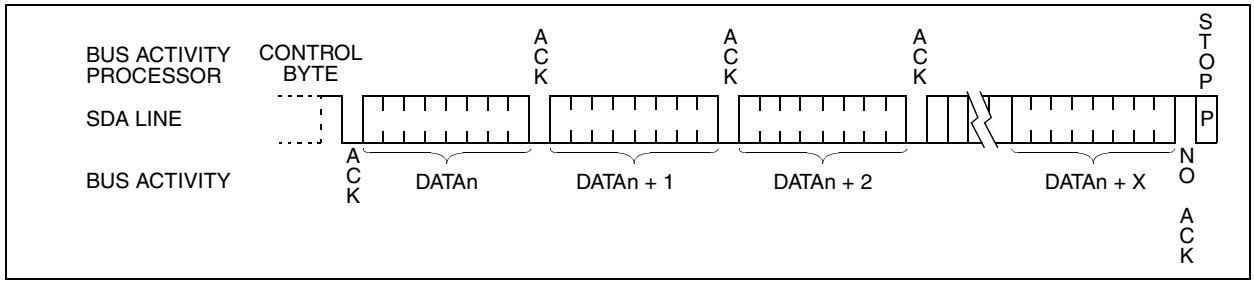


FIGURE 6-9: SEQUENTIAL READ



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.

8.4 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 **).**

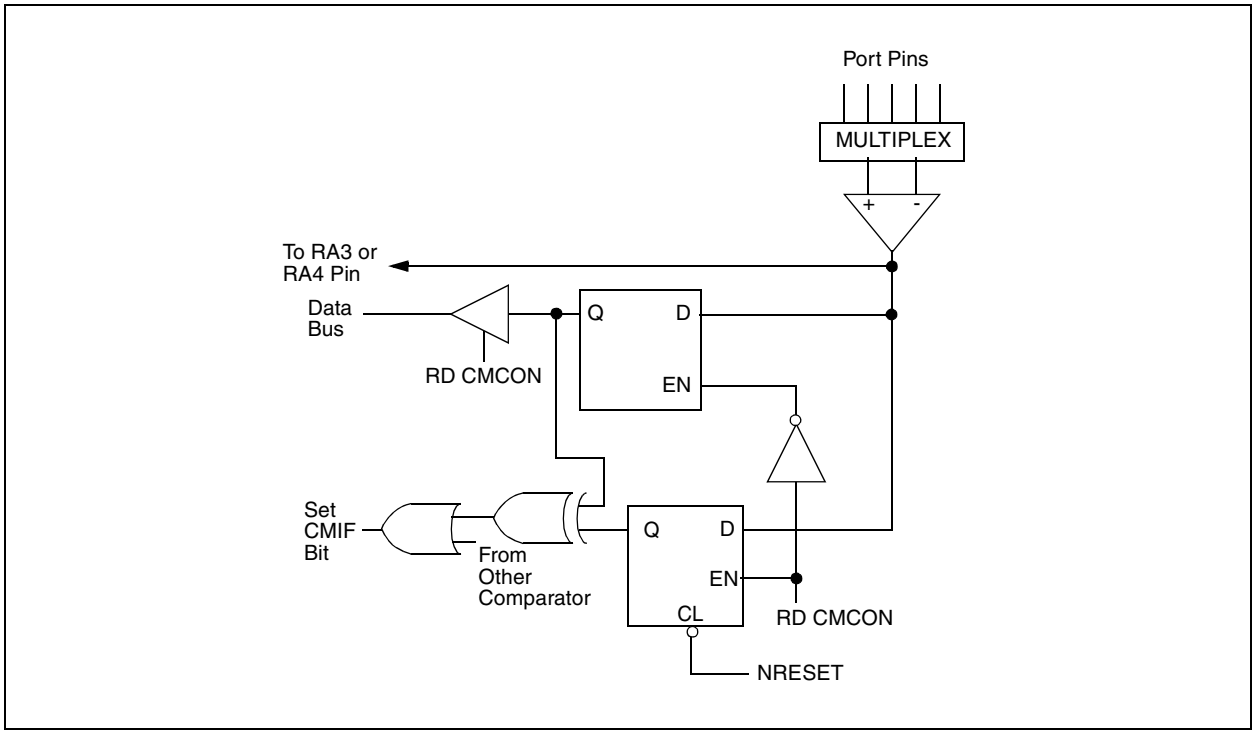
8.5 Comparator Outputs

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



10.3 Reset

The PIC16CE62X differentiates between various kinds of reset:

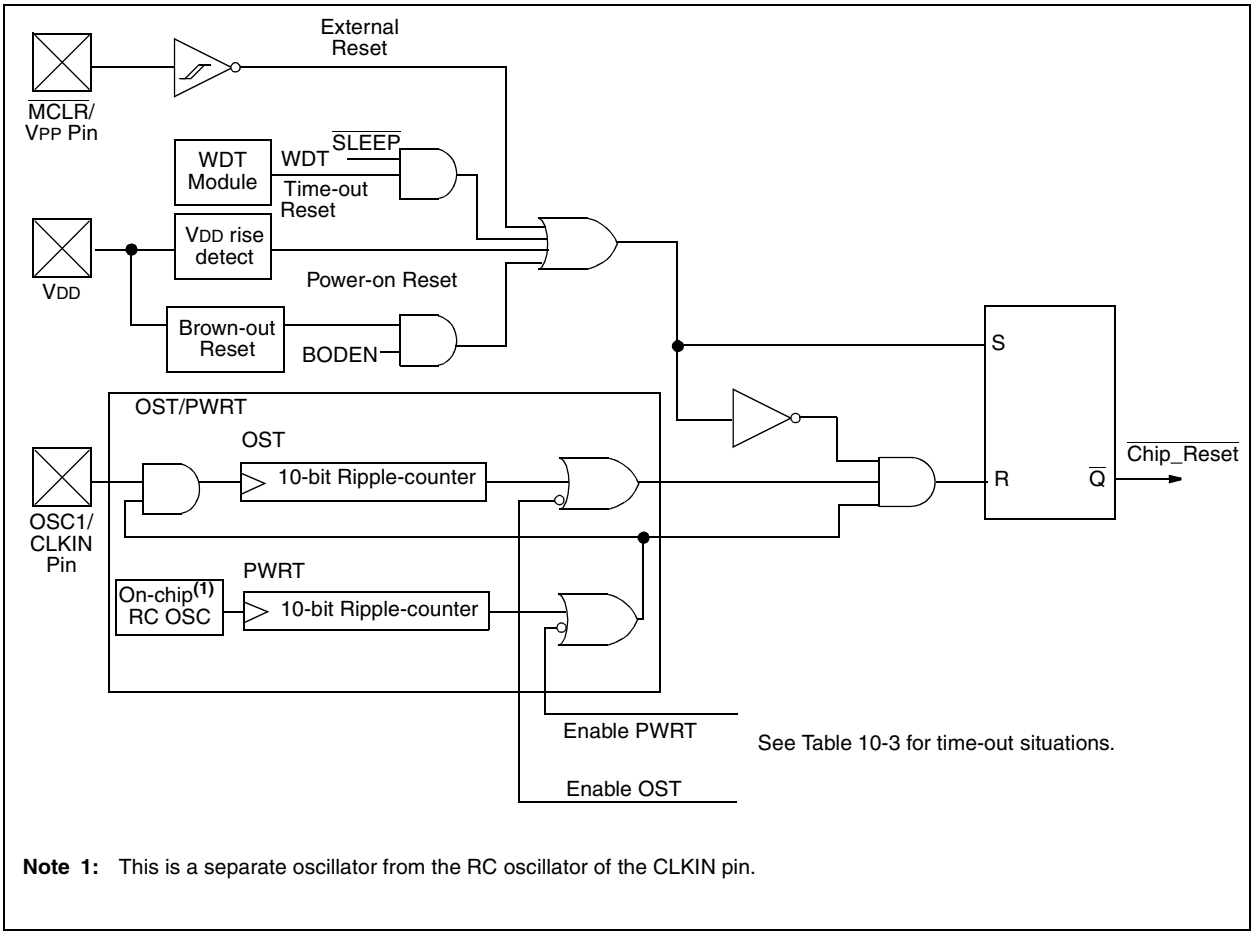
a) Power-on reset (POR)
b) $\overline{MCLR}$ reset during normal operation
c) $\overline{MCLR}$ reset during SLEEP
d) WDT reset (normal operation)
e) WDT wake-up (SLEEP)
f) Brown-out Reset (BOD)

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, $\overline{MCLR}$ reset, WDT reset and $\overline{MCLR}$ reset during SLEEP. They are not affected by a WDT wake-up, since this is viewed as the resumption of normal operation. $\overline{TO}$ and $\overline{PD}$ 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.

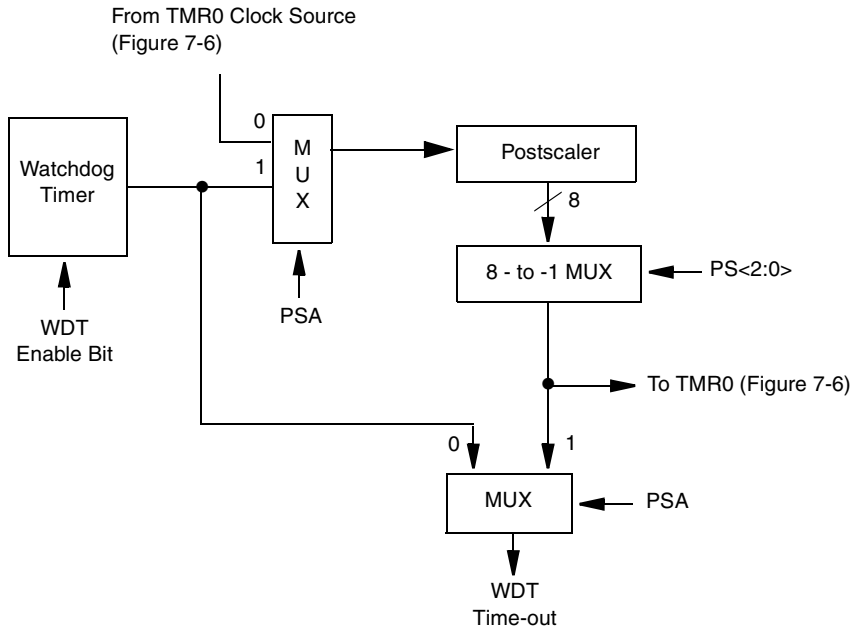
The $\overline{MCLR}$ reset path has a noise filter to detect and ignore small pulses. See Table 13-5 for pulse width specification.

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



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

FIGURE 10-17: WATCHDOG TIMER BLOCK DIAGRAM



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.

10.9 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

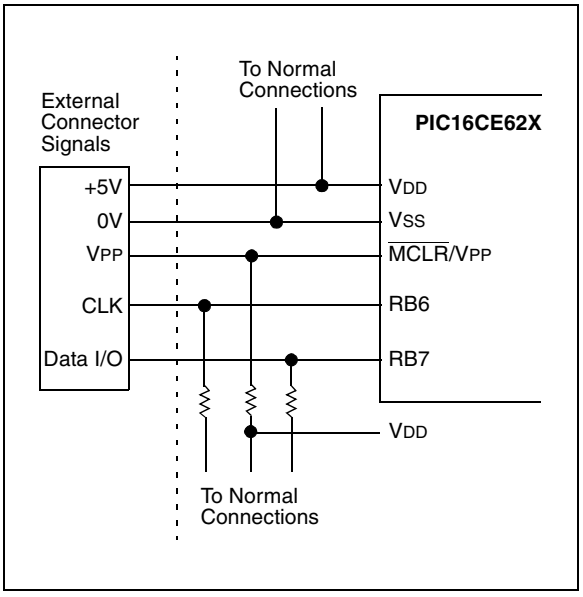
The PIC16CE62X microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. This allows customers to manufacture boards with unprogrammed devices, and then 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 $\overline{MCLR}$ (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



11.1 Instruction Descriptions

ADDLW **Add Literal and W**

Syntax: [*label*] ADDLW k

Operands: $0 \le k \le 255$

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

ANDLW **AND Literal with W**

Syntax: [*label*] ANDLW k

Operands: $0 \le k \le 255$

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

ADDWF **Add W and f**

Syntax: [*label*] ADDWF f,d

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

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

ANDWF **AND W with f**

Syntax: [*label*] ANDWF f,d

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

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

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 PIC® MCU 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
```

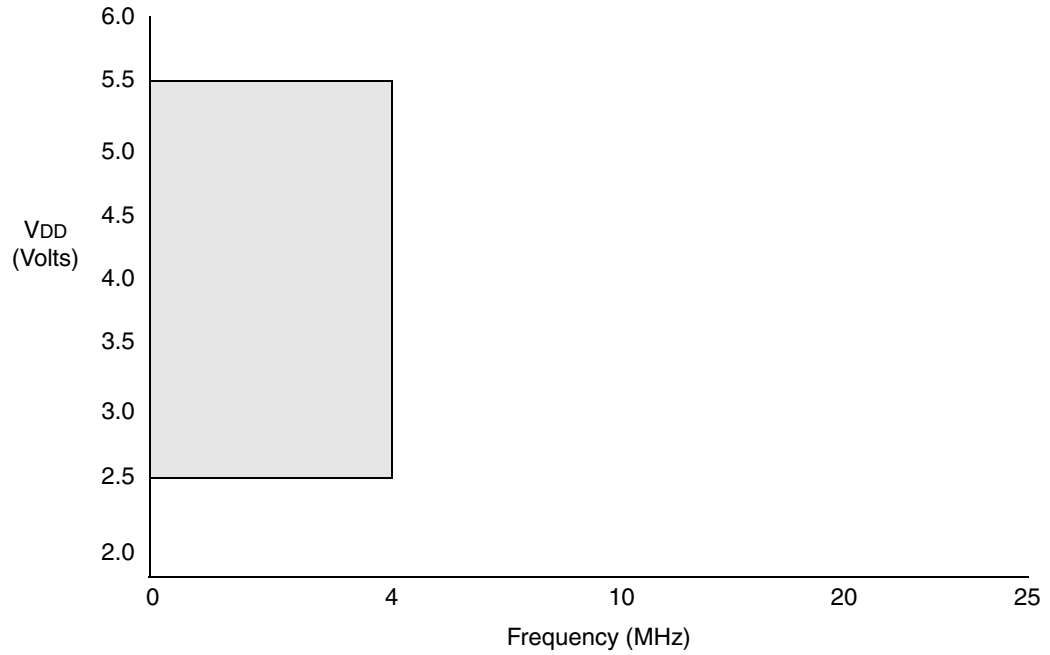
Before Instruction

REG = 0xAF
W = 0xB5

After Instruction

REG = 0x1A
W = 0xB5

FIGURE 13-3: PIC16LCE62X VOLTAGE-FREQUENCY GRAPH, -40°C ≤ TA < +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.

13.3 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		$\overline{MCLR}$, 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		$\overline{MCLR}$ 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
	IIL	**Input Leakage Current** (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, $\overline{MCLR}$	–	–	±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.

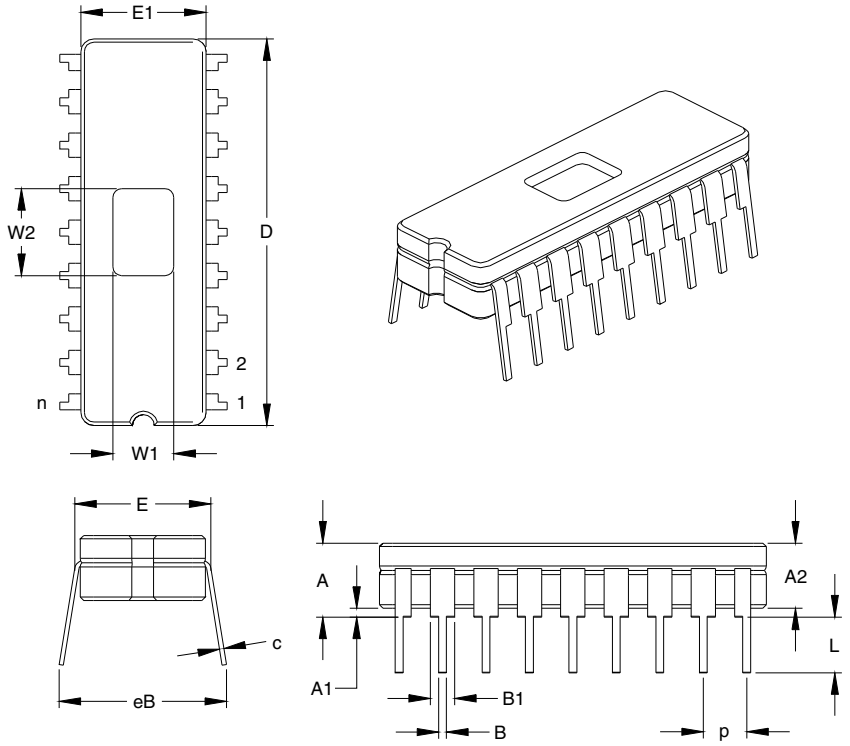
2: The leakage current on the $\overline{MCLR}$ pin is strongly dependent on applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

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

14.0 PACKAGING INFORMATION

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

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



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

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) 792-4150.

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

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: PIC16xxxxxx family Literature Number: DS40182D

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 document easy to follow? If not, why?

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

5. What deletions from the document 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?

NOTES: