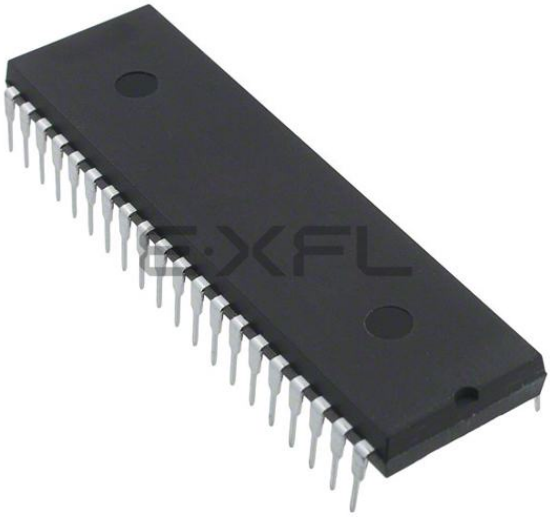


Welcome to E-XFL.COM

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.

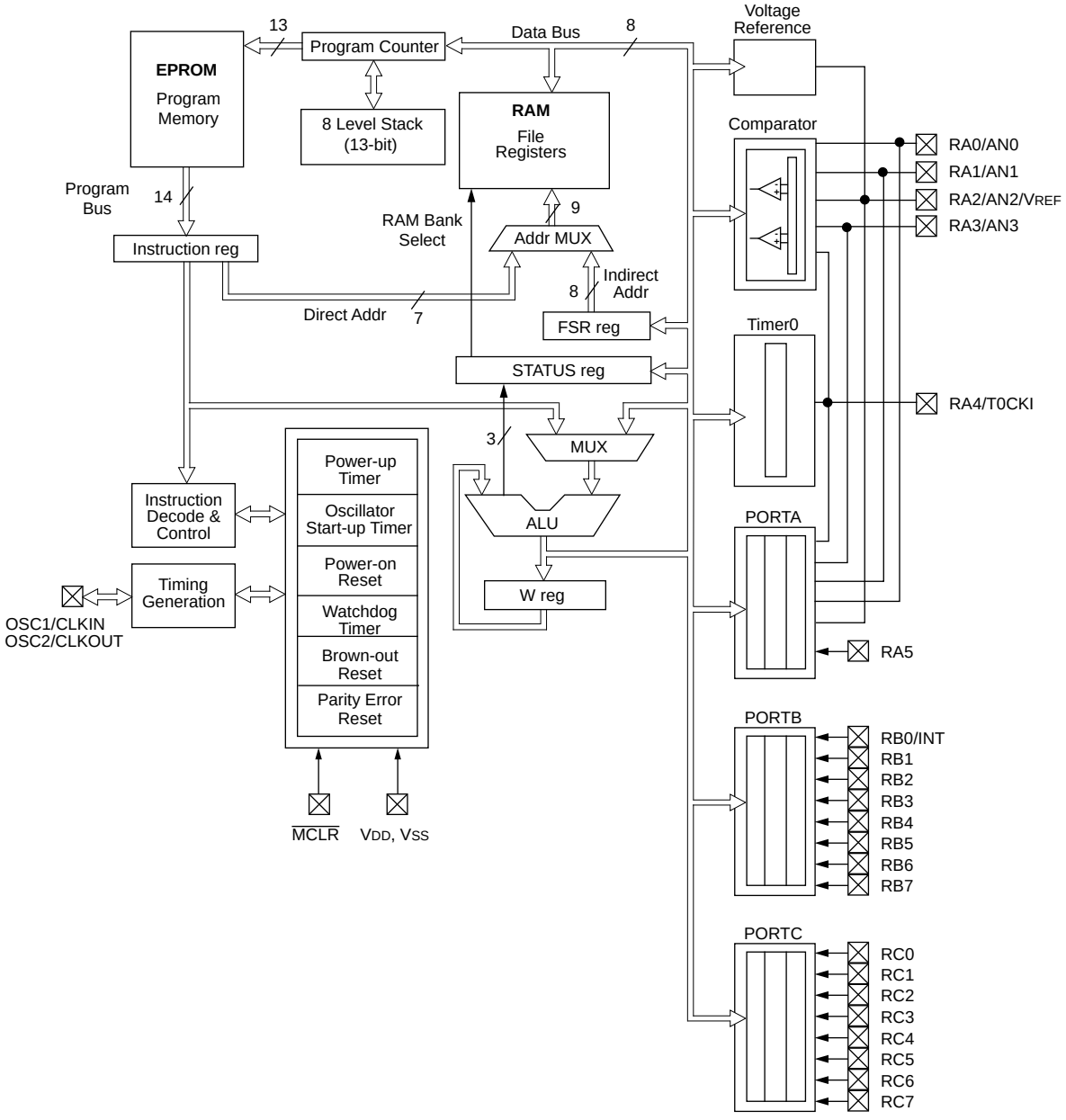
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	-
Peripherals	Brown-out Detect/Reset, LED, POR, WDT
Number of I/O	33
Program Memory Size	7KB (4K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	176 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lc662-04i-p

FIGURE 3-1: PIC16C641/642 BLOCK DIAGRAM

PIC16C641 has 2K x 14 Program Memory and 128 x 8 RAM

PIC16C642 has 4K x 14 Program Memory and 176 x 8 RAM



4.0 MEMORY ORGANIZATION

4.1 Program Memory Organization

The PIC16C64X & PIC16C66X have a 13-bit program counter capable of addressing an 8K x 14 program memory space. For the PIC16C641 and PIC16C661 only the first 2K x 14 (0000h - 07FFh) is physically implemented. For the PIC16C642 and PIC16C662 only the first 4K x 14 (0000h - 0FFh) is physically implemented. Accessing a location above the 2K or 4K boundary will cause a wrap-around. The reset vector is at 0000h and the interrupt vector is at 0004h (Figure 4-1 and Figure 4-2). See Section 4.4 for Program Memory paging.

FIGURE 4-1: PIC16C641/661 PROGRAM MEMORY MAP AND STACK

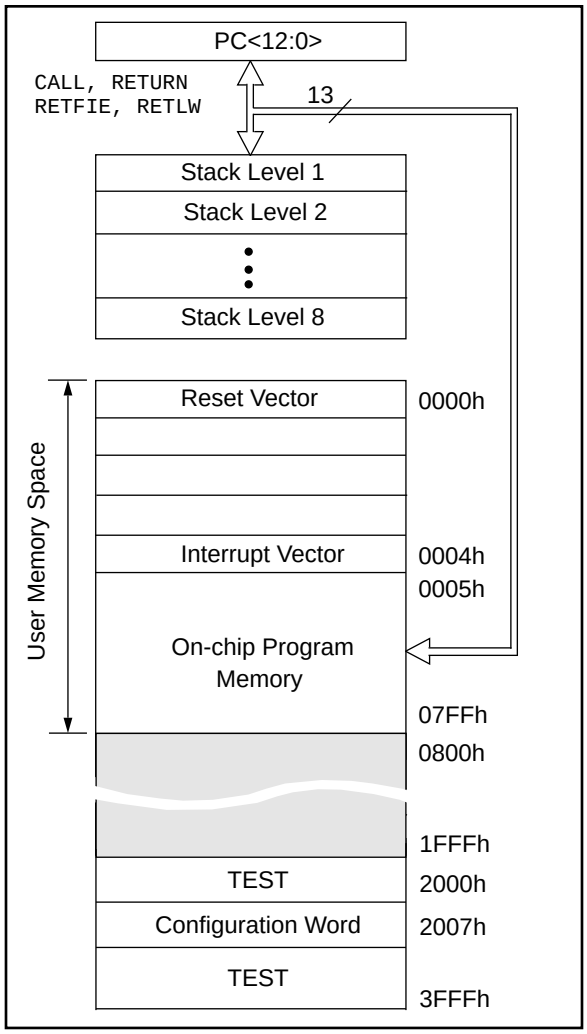
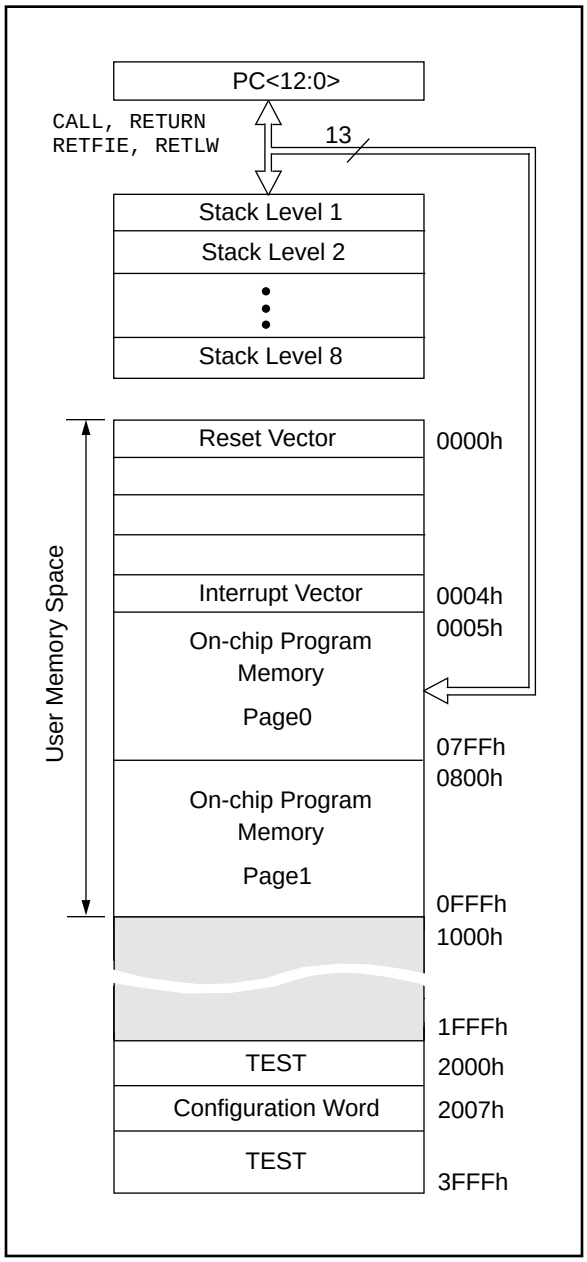


FIGURE 4-2: PIC16C642/662 PROGRAM MEMORY MAP AND STACK



4.2 Data Memory Organization

The data memory (Figure 4-4) is partitioned into two banks which contain the general purpose registers and the special function registers. Bank 0 is selected when bit RP0 (STATUS<5>) is cleared. Bank 1 is selected when the RP0 bit is set. The Special Function Registers are located in the first 32 locations of each Bank. Register locations A0h-EFh (Bank 1) are general purpose registers implemented as static RAM. Some special function registers are mapped in Bank 1.

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file is organized as 176 x 8 for the PIC16C642/662, and 128 x8 for the PIC16C641/661. Each is accessed either directly, or indirectly through the File Select Register FSR (Section 4.5).

FIGURE 4-3: PIC16C641/661 DATA MEMORY MAP

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	PORTC	TRISC	87h
08h	PORTD(2)	TRISD(2)	88h
09h	PORTE(2)	TRISE(2)	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	PIR1	PIE1	8Ch
0Dh			8Dh
0Eh		PCON	8Eh
0Fh			8Fh
10h			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
			EFh
		Mapped in Page 0	F0h
7Fh			FFh

Unimplemented data memory locations, read as '0'.

Note 1: Not a physical register.
2: Not implemented on the PIC16C641.

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 non-peripheral interrupt sources.

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>).

FIGURE 4-7: INTCON REGISTER (ADDRESS 0Bh, 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

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 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 RB7:RB4 pins changed state (See Section 5.2 to clear interrupt)
0 = None of the RB7:RB4 pins have changed state

4.2.2.6 PCON REGISTER

The PCON register contains flag bits to differentiate between a Power-on Reset (POR), an external $\overline{MCLR}$ reset, WDT reset, Brown-out Reset (BOR), and Parity Error Reset (PER). The PCON register also contains a status bit, MPEEN, which reflects the value of the MPEEN bit in Configuration Word. See Table 9-4 for status of these bits on various resets.

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

FIGURE 4-10: PCON REGISTER (ADDRESS 8Eh)

R-U	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-u
MPEEN	—	—	—	—	$\overline{PER}$	$\overline{POR}$	$\overline{BOR}$
bit7							bit0

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

bit 7: **MPEEN**: Memory Parity Error Circuitry Status bit
Reflects the value of Configuration Word bit, MPEEN

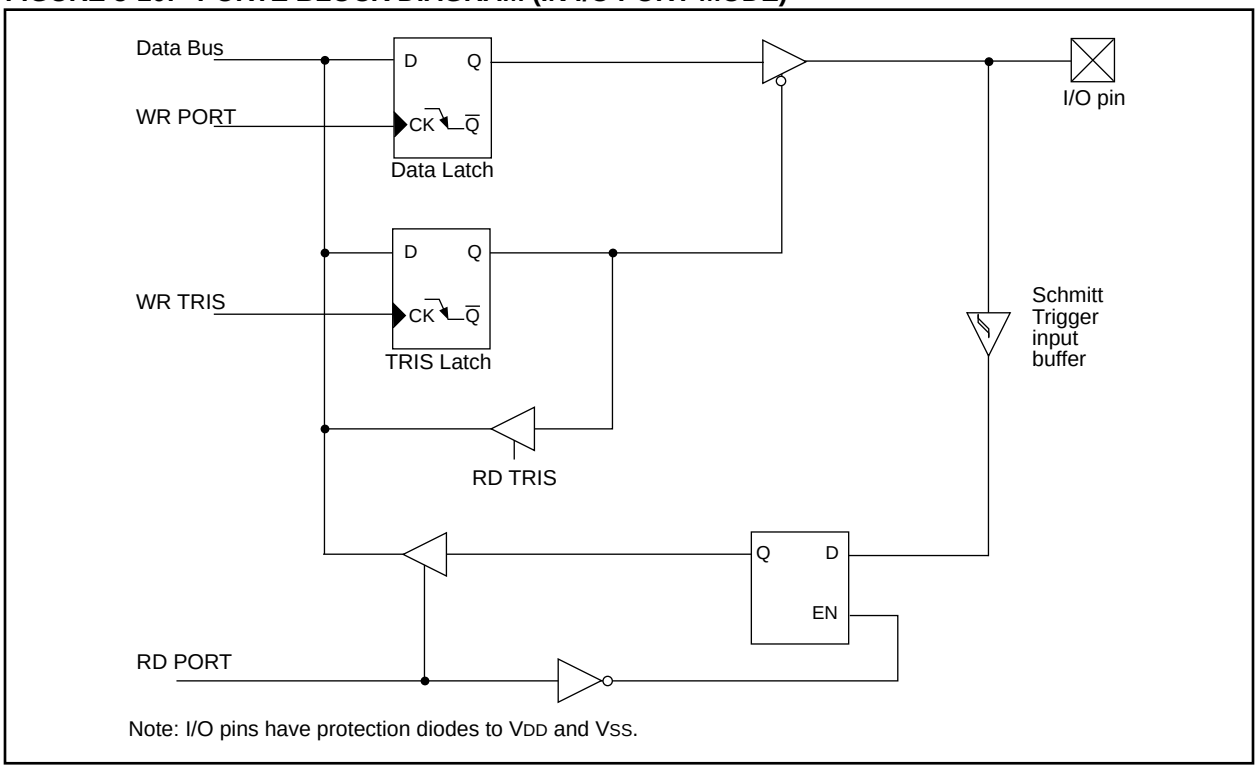
bit 6-3: **Unimplemented**: Read as '0'

bit 2: **$\overline{PER}$**: Memory Parity Error Reset Status bit
1 = No error occurred
0 = Program memory fetch parity error occurred
(must be set in software after a Parity Error Reset occurs)

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: **$\overline{BOR}$**: 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)

FIGURE 5-10: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)



Note: I/O pins have protection diodes to VDD and VSS.

TABLE 5-9: PORTE FUNCTIONS

Name	Bit#	Buffer Type	Function
RE0/$\overline{RD}$	bit0	ST/TTL(1)	Input/output port pin or read control input in parallel slave port mode: $\overline{RD}$ 1 = Not a read operation 0 = Read operation. Reads PORTD register (if chip selected)
RE1/$\overline{WR}$	bit1	ST/TTL(1)	Input/output port pin or write control input in parallel slave port mode: $\overline{WR}$ 1 = Not a write operation 0 = Write operation. Writes PORTD register (if chip selected)
RE2/$\overline{CS}$	bit2	ST/TTL(1)	Input/output port pin or chip select control input in parallel slave port mode: $\overline{CS}$ 1 = Device is not selected 0 = Device is selected

Legend: ST = Schmitt Trigger input, TTL = TTL input
Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port Mode.

TABLE 5-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
09h	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	TRISE2	TRISE1	TRISE0	0000 -111	0000 -111

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not used by PORTE.

5.6 I/O Programming Considerations

5.6.1 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 bi-directional I/O pin (e.g., 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 rewritten 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 (e.g., 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 5-4 shows the effect of two sequential read-modify-write instructions 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.

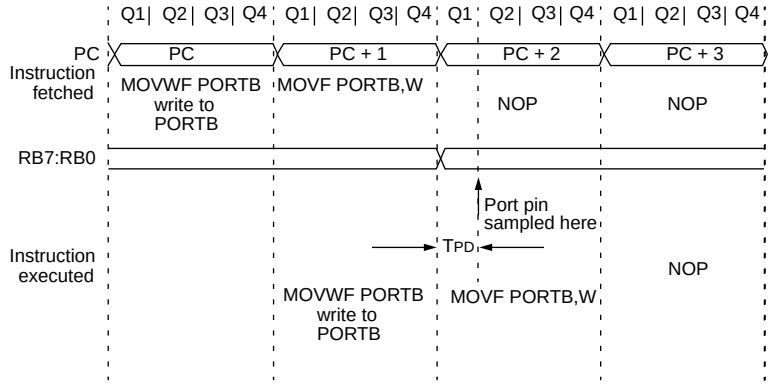
EXAMPLE 5-4: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```
;Initial PORT settings: PORTB<7:4> Inputs
;                       PORTB<3:0> Outputs
;PORTB<7:6> have external pull-ups 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
  BCF STATUS, RP1 ;
  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 ppp. The 2nd BCF
;caused RB7 to be latched as the pin value
;(high).
```

5.6.2 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 5-11). 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 be such to allow the pin voltage to stabilize (load dependent) before the next instruction which causes that file to be read into the CPU is executed. 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 5-11: SUCCESSIVE I/O OPERATION



Note:

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

Note that:

data setup time = ($0.25T_{CY} - T_{PD}$)

where T_{CY} = instruction cycle
T_{PD} = propagation delay

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

6.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control, i.e., it can be changed “on the fly” during program execution.

Note:	To avoid an unintended device RESET, the following instruction sequence (shown in Example 6-1) must be executed when changing the prescaler assignment from Timer0 to the WDT. This precaution must be followed even if the WDT is disabled.

EXAMPLE 6-1: CHANGING PRESCALER (TIMER0→WDT)

```
BCF     STATUS, RP0  ;Bank 0
CLRF    TMR0         ;Clear TMR0 & Prescaler
BSF     STATUS, RP0  ;Bank 1
CLRWDT               ;Clears WDT
MOVLW   b'xxxx1xxx'  ;Select new prescale
MOVWF   OPTION_REG   ;value & WDT
BCF     STATUS, RP0  ;Bank 0
```

To change prescaler from the WDT to the Timer0 module, use the sequence shown in Example 6-2.

EXAMPLE 6-2: CHANGING PRESCALER (WDT→TIMER0)

```
CLRWDT               ;Clear WDT and
                     ;prescaler
BSF     STATUS, RP0  ;Bank 1
MOVLW   b'xxxx0xxx'  ;Select TMR0, new
                     ;prescale value and
MOVWF   OPTION_REG   ;clock source
BCF     STATUS, RP0  ;Bank 0
```

TABLE 6-1: 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, BOR	Value on all other resets
01h	TMR0	Timer0 module’s 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	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'. Shaded cells are not used by Timer0.

7.0 COMPARATOR MODULE

The comparator module contains two analog comparators. The inputs to the comparators are multiplexed with pins RA0 through RA4. The on-chip Voltage Reference (Section 8.0) can also be an input to the comparators.

The CMCON register, shown in Figure 7-1, controls the comparator input and output multiplexers. A block diagram of the comparator is shown in Figure 7-2.

FIGURE 7-1: 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 V_{IN}+ > C2 V_{IN}–
0 = C2 V_{IN}+ < C2 V_{IN}–

bit 6: **C1OUT**: Comparator 1 output
1 = C1 V_{IN}+ > C1 V_{IN}–
0 = C1 V_{IN}+ < C1 V_{IN}–

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

bit 3: **CIS**: Comparator Input Switch

When CM2:CM0: = 001:
Then:
1 = C1 V_{IN}– connects to RA3
0 = C1 V_{IN}– connects to RA0

When CM2:CM0 = 010:
Then:
1 = C1 V_{IN}– connects to RA3
C2 V_{IN}– connects to RA2
0 = C1 V_{IN}– connects to RA0
C2 V_{IN}– connects to RA1

bit 2-0: **CM2:CM0**: Comparator mode
Figure 7-2 shows the comparator modes and CM2:CM0 bit settings.

8.0 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 module is not being used.

The VRCON register, shown in Figure 8-1, controls the operation of the Voltage Reference Module. The block diagram is given in Figure 8-2.

FIGURE 8-1: 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 up
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

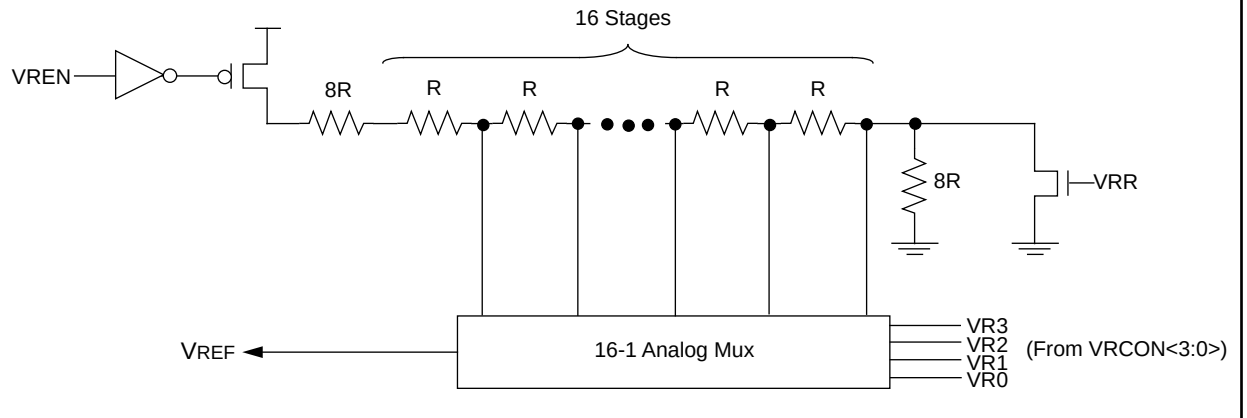
bit 4: **Unimplemented:** Read as '0'

bit 3-0: **VR3:VR0**: VREF value selection 0 ≤ VR3:VR0 ≤ 15

When: VRR = 1
Then: VREF = (VR3:VR0/ 24) • VDD

When: VRR = 0
Then: VREF = 1/4 • VDD + (VR3:VR0/ 32) • VDD

FIGURE 8-2: VOLTAGE REFERENCE BLOCK DIAGRAM



Note: R is defined in Table 12-3.

9.0 SPECIAL FEATURES OF THE CPU

What sets apart a microcontroller from other processors are special circuits to deal with the needs of real-time applications. The PIC16C64X & PIC16C66X families have 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. Oscillator selection
2. Resets
   Power-on Reset (POR)
   Power-up Timer (PWRT)
   Oscillator Start-up Timer (OST)
   Brown-out Reset (BOR)
   Parity Error Reset (PER)
3. Interrupts
4. Watchdog Timer (WDT)
5. SLEEP
6. Code protection
7. ID Locations
8. In-circuit serial programming

The PIC16C64X & PIC16C66X has a Watchdog Timer which is enabled by a configuration bit (WDTE). 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, designed to keep the part in reset while the power supply stabilizes. Circuitry has been provided for checking program memory parity with a reset when an error is indicated. 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.

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.

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 is 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

NOTES:

Standard Operating Conditions (unless otherwise stated)
Operating temperature –40°C ≤ TA ≤ +85°C for industrial,
0°C ≤ TA ≤ +70°C commercial, and
–40°C ≤ TA ≤ +125°C automotive
Operating voltage VDD range as described in DC spec Section 12.1 and 12.2

Param No.	Sym	Characteristic	Min	Typ †	Max	Unit	Conditions
	VOH	Output High Voltage (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	VDD-0.7	-	-	V	IOH = -1.3 mA, VDD=4.5V, -40° to +85°C
		(RC only)	VDD-0.7	-	-	V	IOH = -1.0 mA, VDD = 4.5V, +125°C
		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.

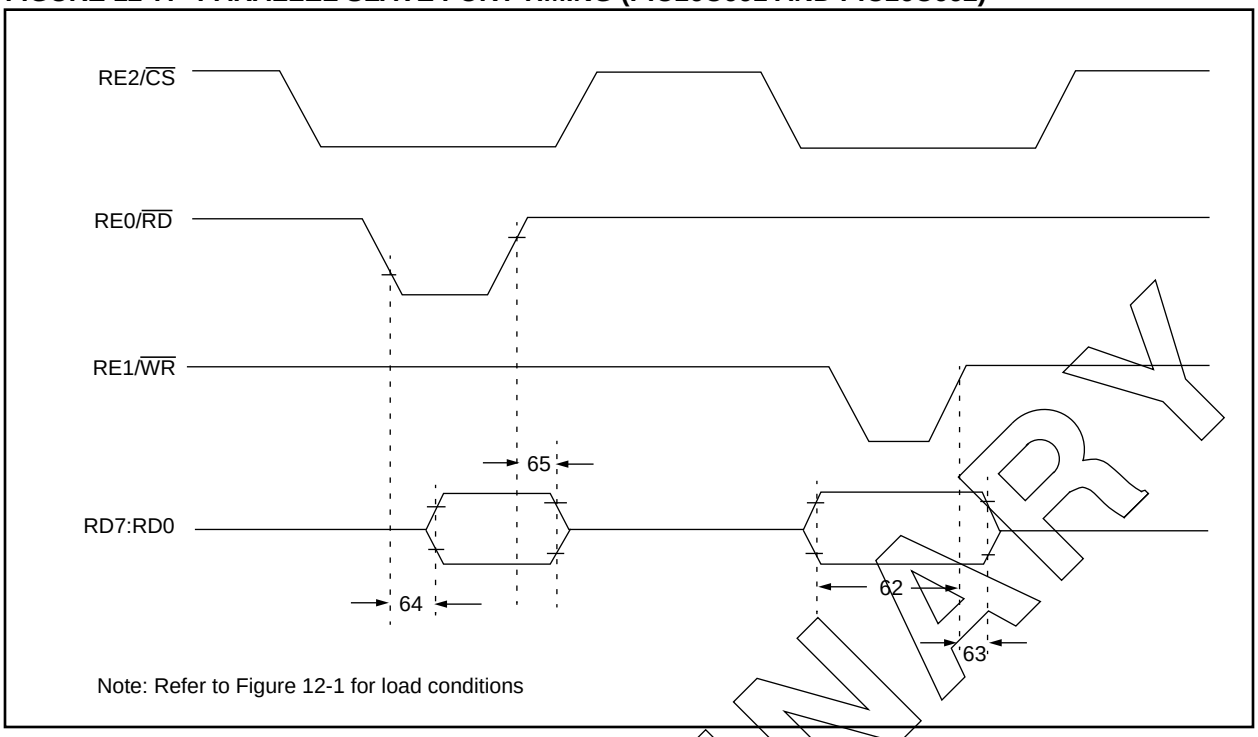
† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt Trigger input. It is not recommended that the PIC16C64X & PIC16C66X 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.

FIGURE 12-7: PARALLEL SLAVE PORT TIMING (PIC16C661 AND PIC16C662)



Note: Refer to Figure 12-1 for load conditions

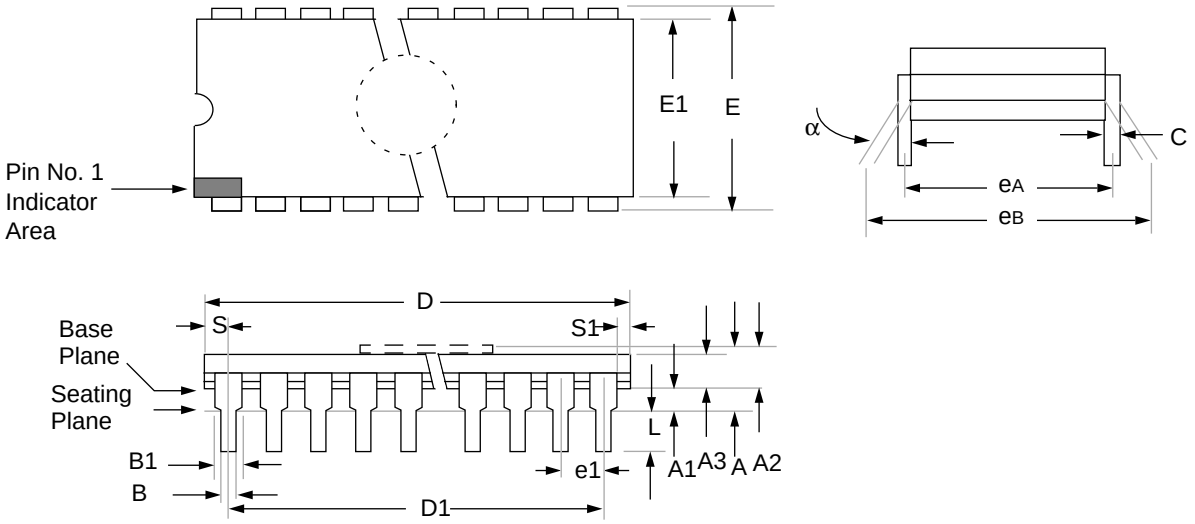
TABLE 12-8: PARALLEL SLAVE PORT REQUIREMENTS (PIC16C661 AND PIC16C662)

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
62	TdtV2wrH	Data in valid before $\overline{WR}\uparrow$ or $\overline{CS}\uparrow$ (setup time)		20	—	—	ns	
63*	TwrH2dtI	$\overline{WR}\uparrow$ or $\overline{CS}\uparrow$ to data–in invalid (hold time)	PIC16C66X	20	—	—	ns	
			PIC16LC66X	35	—	—	ns	
64	TrdL2dtV	$\overline{RD}\downarrow$ and $\overline{CS}\downarrow$ to data–out valid		—	—	80	ns	
65	TrdH2dtI	$\overline{RD}\uparrow$ or $\overline{CS}\downarrow$ to data–out invalid		10	—	30	ns	

† Data in "Typ" column is at 5V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

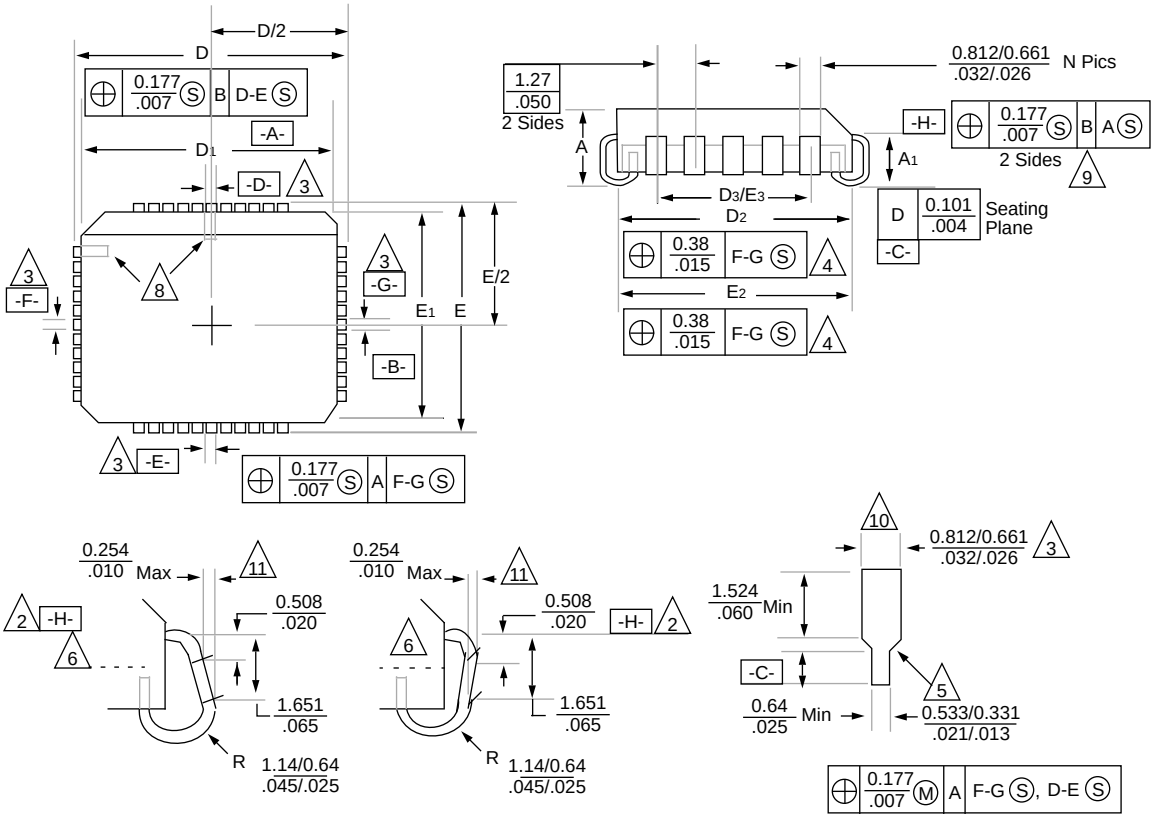
NOTES:

Package Type: 40-Lead Ceramic Dual In-Line with Window (JW) - (600 mil)



Package Group: Ceramic CERDIP Dual In-Line (CDP)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
α	0°	10°		0°	10°	
A	4.318	5.715		0.170	0.225	
A1	0.381	1.778		0.015	0.070	
A2	3.810	4.699		0.150	0.185	
A3	3.810	4.445		0.150	0.175	
B	0.355	0.585		0.014	0.023	
B1	1.270	1.651	**Typical**	0.050	0.065	**Typical**
C	0.203	0.381	**Typical**	0.008	0.015	**Typical**
D	51.435	52.705		2.025	2.075	
D1	48.260	48.260	**BSC**	1.900	1.900	**BSC**
E	15.240	15.875		0.600	0.625	
E1	12.954	15.240		0.510	0.600	
e1	2.540	2.540	**BSC**	0.100	0.100	**BSC**
eA	14.986	16.002	**Typical**	0.590	0.630	**Typical**
eB	15.240	18.034		0.600	0.710	
L	3.175	3.810		0.125	0.150	
S	1.016	2.286		0.040	0.090	
S1	0.381	1.778		0.015	0.070	

Package Type: 44-Lead Plastic Leaded Chip Carrier (L) - Square



Package Group: Plastic Leaded Chip Carrier (PLCC)						
	Millimeters			Inches		
Symbol	Min	Max	Notes	Min	Max	Notes
A	4.191	4.572		0.165	0.180	
A1	2.413	2.921		0.095	0.115	
D	17.399	17.653		0.685	0.695	
D1	16.510	16.663		0.650	0.656	
D2	15.494	16.002		0.610	0.630	
D3	12.700	12.700	**BSC**	0.500	0.500	**BSC**
E	17.399	17.653		0.685	0.695	
E1	16.510	16.663		0.650	0.656	
E2	15.494	16.002		0.610	0.630	
E3	12.700	12.700	**BSC**	0.500	0.500	**BSC**
CP	—	0.102		—	0.004	
LT	0.203	0.381		0.008	0.015	

NOTES:

PIN COMPATIBILITY

Devices that have the same package type and VDD, VSS and $\overline{MCLR}$ pin locations are said to be pin compatible. This allows these different devices to operate in the same socket. Compatible devices may only requires minor software modification to allow proper operation in the application socket (ex., PIC16C56 and PIC16C61 devices). Not all devices in the same package size are pin compatible; for example, the PIC16C62 is compatible with the PIC16C63, but not the PIC16C55.

Pin compatibility does not mean that the devices offer the same features. As an example, the PIC16C54 is pin compatible with the PIC16C71, but does not have an A/D converter, weak pull-ups on PORTB, or interrupts.

TABLE E-1: PIN COMPATIBLE DEVICES

Pin Compatible Devices	Package
PIC12C508, PIC12C509	8-pin
PIC16C54, PIC16C54A, PIC16CR54A, PIC16C56, PIC16C58A, PIC16CR58A, PIC16C61, PIC16C554, PIC16C556, PIC16C558 PIC16C620, PIC16C621, PIC16C622, PIC16C710, PIC16C71, PIC16C711, PIC16F83, PIC16CR83, PIC16C84, PIC16F84A, PIC16CR84	18-pin 20-pin
PIC16C55, PIC16C57, PIC16CR57B	28-pin
PIC16C62, PIC16CR62, PIC16C62A, PIC16C63, PIC16C72, PIC16C73, PIC16C73A	28-pin
PIC16C64, PIC16CR64, PIC16C64A, PIC16C65, PIC16C65A, PIC16C74, PIC16C74A	40-pin
PIC17C42, PIC17CR42, PIC17C42A, PIC17C43, PIC17CR43, PIC17C44	40-pin
PIC16C923, PIC16C924	64/68-pin