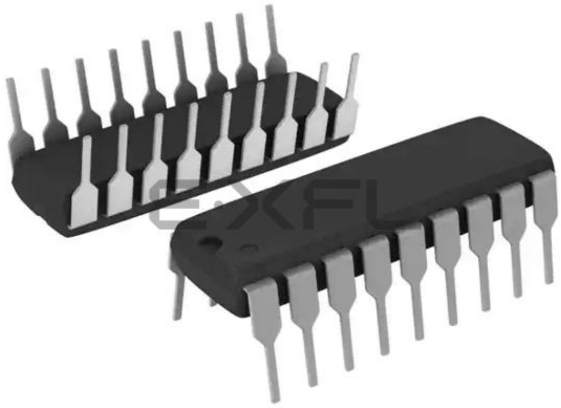


Welcome to E-XFL.COM

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	POR, WDT
Number of I/O	13
Program Memory Size	3.5KB (2K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°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/pic16lc558-04-p

3.0 ARCHITECTURAL OVERVIEW

The high performance of the PIC16C55X family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16C55X uses a Harvard architecture in which program and data are accessed from separate memories using separate busses. 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 from 8-bit wide data words. Instruction opcodes are 14-bit 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 the memory (EPROM and RAM).

Device	Program Memory (EPROM)	Data Memor (RAM)
PIC16C554	512	80
PIC16C557	2 K	128
PIC16C558	2 K	128

The PIC16C554 addresses 512 x 14 on-chip program memory. The PIC16C557 and PIC16C558 addresses 2 K x 14 program memory. All program memory is internal.

The PIC16C55X can directly or indirectly address its register files or data memory. All special function registers, including the program counter, are mapped into the data memory. The PIC16C55X 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 PIC16C55X simple yet efficient. In addition, the learning curve is reduced significantly.

The PIC16C55X 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 $\overline{\text{Digit Borrow}}$ out bit, respectively, 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.

TABLE 3-1: PIC16C55X PINOUT DESCRIPTION

Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	SOIC	SSOP			
OSC1/CLKIN	16	16	18	I	ST/CMOS	Oscillator crystal input/external clock source output.
OSC2/CLKOUT	15	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.
$\overline{MCLR}$/VPP	4	4	4	I/P	ST	Master clear (Reset) input/programming voltage input. This pin is an active low RESET to the device.
RA0	17	17	19	I/O	ST	Bi-directional I/O port
RA1	18	18	20	I/O	ST	Bi-directional I/O port
RA2	1	1	1	I/O	ST	Bi-directional I/O port
RA3	2	2	2	I/O	ST	Bi-directional I/O port
RA4/T0CKI	3	3	3	I/O	ST	Bi-directional I/O port or external clock input for TMR0. Output is open drain type.
RB0/INT	6	6	7	I/O	TTL/ST(1)	Bi-directional I/O port can be software programmed for internal weak pull-up. RB0/INT can also be selected as an external interrupt pin.
RB1	7	7	8	I/O	TTL	Bi-directional I/O port can be software programmed for internal weak pull-up.
RB2	8	8	9	I/O	TTL	Bi-directional I/O port can be software programmed for internal weak pull-up.
RB3	9	9	10	I/O	TTL	Bi-directional I/O port can be software programmed for internal weak pull-up.
RB4	10	10	11	I/O	TTL	Bi-directional I/O port can be software programmed for internal weak pull-up. Interrupt-on-change pin.
RB5	11	11	12	I/O	TTL	Bi-directional I/O port can be software programmed for internal weak pull-up. Interrupt-on-change pin.
RB6	12	12	13	I/O	TTL/ST(2)	Bi-directional I/O port can be software programmed for internal weak pull-up. Interrupt-on-change pin. Serial programming clock.
RB7	13	13	14	I/O	TTL/ST(2)	Bi-directional I/O port can be software programmed for internal weak pull-up. Interrupt-on-change pin. Serial programming data.
RC0(3)	18	18	18	I/O	TTL	Bi-directional I/O port input buffer.
RC1(3)	19	19	19	I/O	TTL	Bi-directional I/O port input buffer.
RC2(3)	20	20	20	I/O	TTL	Bi-directional I/O port input buffer.
RC3(3)	21	21	21	I/O	TTL	Bi-directional I/O port input buffer.
RC4(3)	22	22	22	I/O	TTL	Bi-directional I/O port input buffer.
RC5(3)	23	23	23	I/O	TTL	Bi-directional I/O port input buffer.
RC6(3)	24	24	24	I/O	TTL	Bi-directional I/O port input buffer.
RC7(3)	25	25	25	I/O	TTL	Bi-directional I/O port input buffer.
VSS	5	5	5,6	P	—	Ground reference for logic and I/O pins.
VDD	14	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.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: PIC16C557 only.

4.2.2.1 STATUS Register

The STATUS register, shown in Figure 4-2, 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 $\overline{TO}$ and $\overline{PD}$ bits are not writable. Therefore, the result of an instruction with the STATUS register as the 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 be used to alter the STATUS register because these instructions do not affect any status bits. 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 PIC16C55X 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.
>
> **2:** The C and DC bits operate as a $\overline{Borrow}$ and $\overline{Digit\ Borrow}$ 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	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit7							bit0

bit 7 **IRP**: Register Bank Select bit (used for Indirect addressing)
`1` = Bank 2, 3 (100h - 1FFh)
`0` = Bank 0, 1 (00h - FFh)
The IRP bit is reserved on the PIC16C55X, always maintain this bit clear

bit 6-5 **RP1:RP0**: 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 on the PIC16C55X, always maintain this bit clear.

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

bit 3 **$\overline{PD}$**: 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

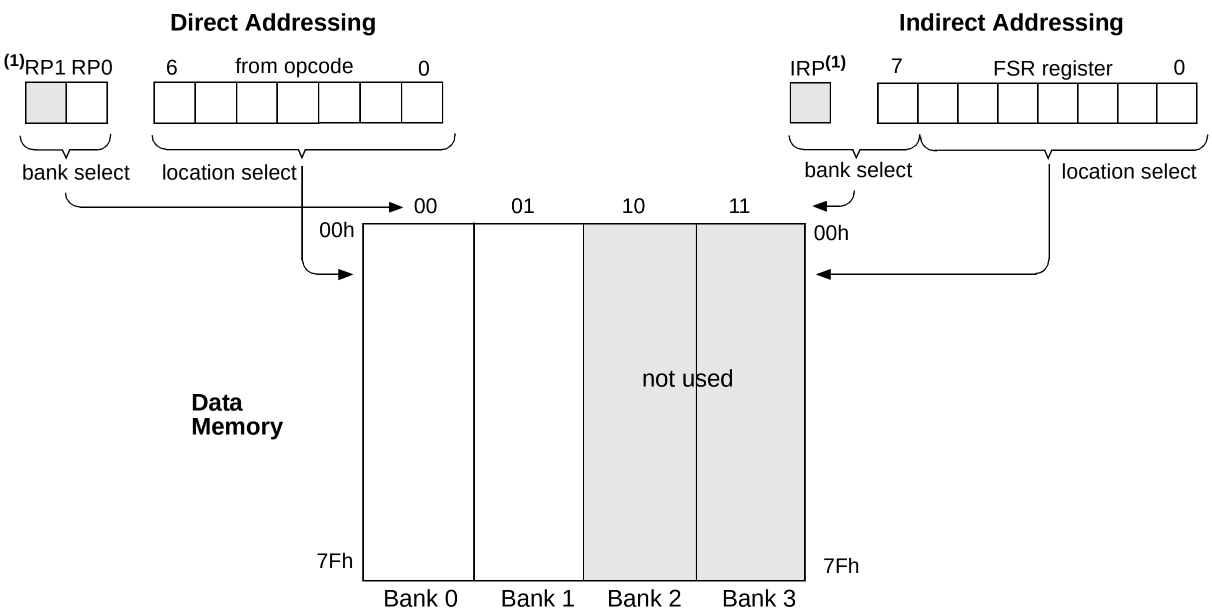
bit 1 **DC**: Digit carry/$\overline{borrow}$ bit (`ADDWF`, `ADDLW`, `SUBLW`, `SUBWF` instructions) (for $\overline{borrow}$ the polarity is reversed)
`1` = A carry-out from the 4th low order bit of the result occurred
`0` = No carry-out from the 4th low order bit of the result

bit 0 **C**: Carry/$\overline{borrow}$ 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 1: For $\overline{borrow}$ 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.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
- n = Value at POR reset	’1’ = Bit is set	’0’ = Bit is cleared	x = Bit is unknown

FIGURE 4-7: DIRECT/INDIRECT ADDRESSING PIC16C55X



For memory map detail see Figure 4-3 and Figure 4-5.

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

5.0 I/O PORTS

The PIC16C554 and PIC16C558 have two ports, PORTA and PORTB. The PIC16C557 has three ports, PORTA, PORTB and PORTC.

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.

Note 1: On RESET, the TRISA register is set to all inputs.

FIGURE 5-1: BLOCK DIAGRAM OF PORT PINS RA<3:0>

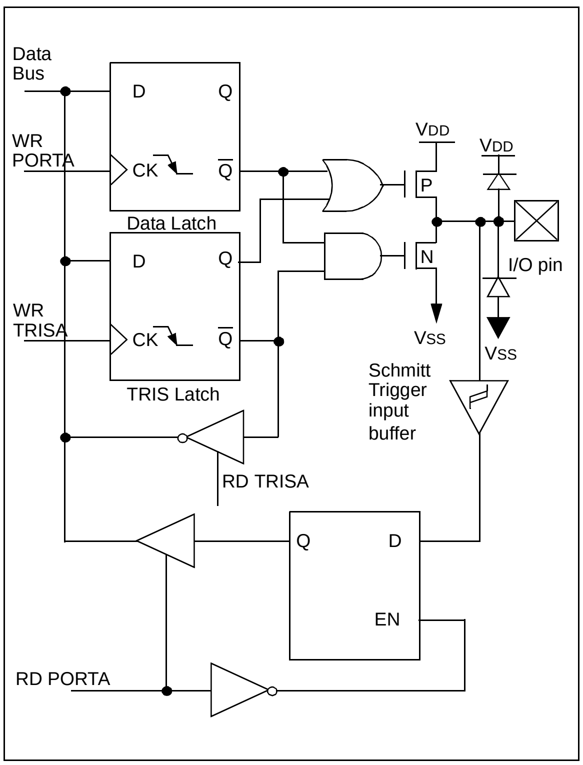
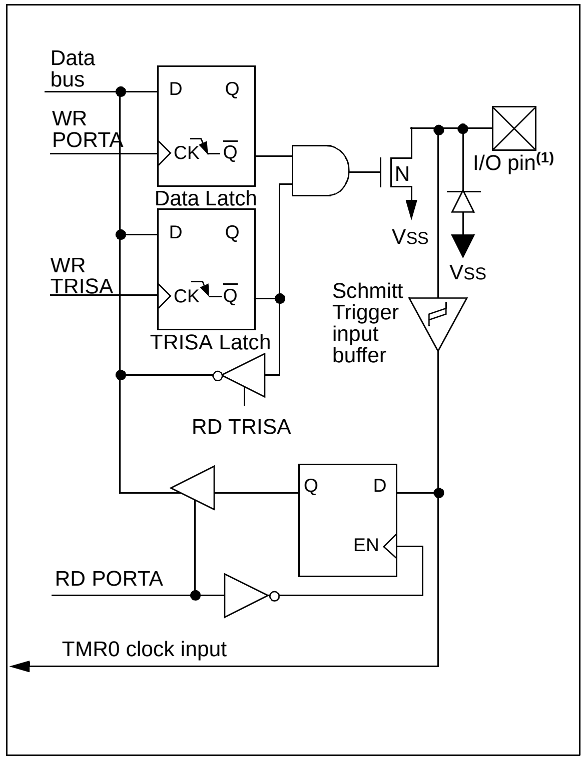


FIGURE 5-2: BLOCK DIAGRAM OF RA4 PIN



5.3 PORTC and TRISC Registers[1]

PORTC is a 8-bit wide latch. All pins have data direction bits (TRIS registers) which can configure these pins as input or output.

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

Reading the PORTC 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

FIGURE 5-5: BLOCK DIAGRAM OF PORT PINS RC<7:0>

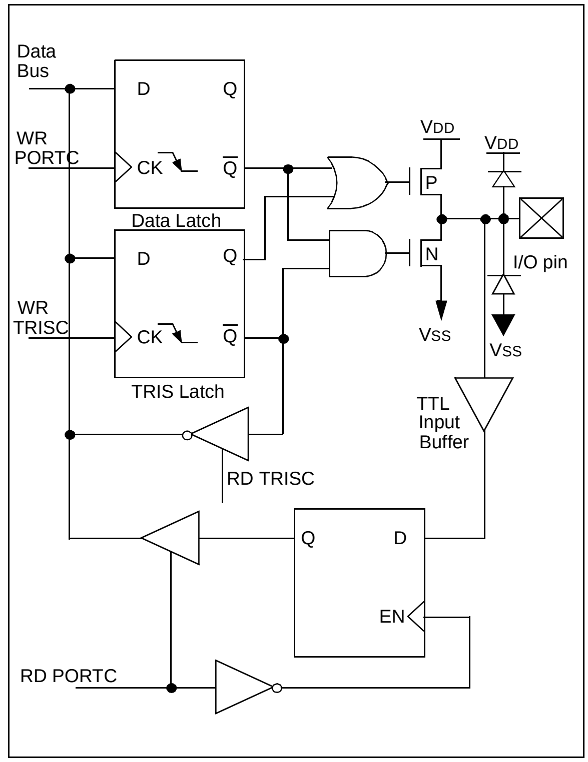


TABLE 5-5: PORTC FUNCTIONS

Name	Bit #	Buffer Type	Function
RC0	Bit 0	TTL	Bi-directional I/O port.
RC1	Bit 1	TTL	Bi-directional I/O port.
RC2	Bit 2	TTL	Bi-directional I/O port.
RC3	Bit 3	TTL	Bi-directional I/O port.
RC4	Bit 4	TTL	Bi-directional I/O port.
RC5	Bit 5	TTL	Bi-directional I/O port.
RC6	Bit 6	TTL	Bi-directional I/O port.
RC7	Bit 7	TTL	Bi-directional I/O port.

Legend: ST = Schmitt Trigger, TTL = TTL input

TABLE 5-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC AND TRISC

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
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
87h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged
Note 1: PIC16C557 ONLY.

EXAMPLE 5-1: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

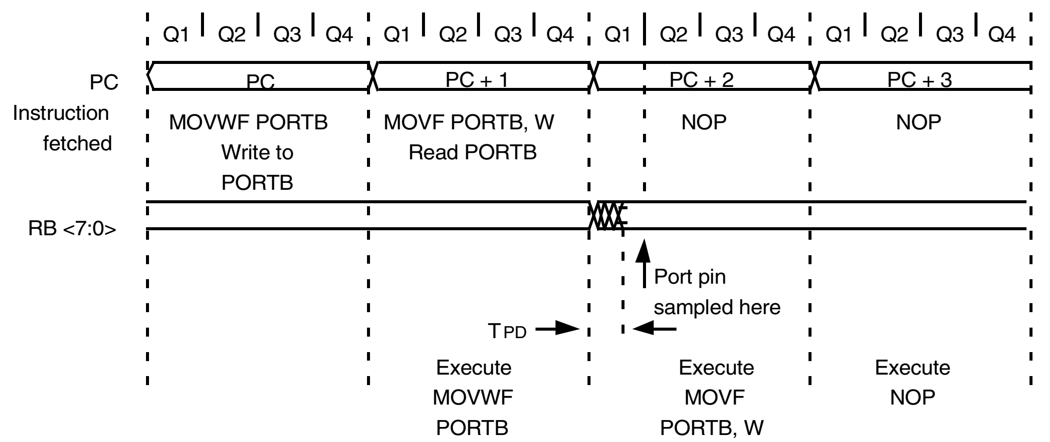
```
; Initial PORT settings: PORTB<7:4> Inputs
;
;                        PORTB<3:0> Outputs
; PORTB<7:6> have external pull-up and are
; not connected to other circuitry
;
;                      PORT latch PORT pins
;                      ---------- ---------
;

     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
```

5.4.2 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, as shown in Figure 5-6. 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 5-6: SUCCESSIVE I/O OPERATION



Note 1: This example shows write to PORTB followed by a read from PORTB.

2: 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.

6.2 Oscillator Configurations

6.2.1 OSCILLATOR TYPES

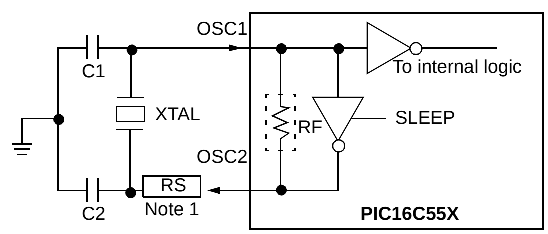
The PIC16C55X can be operated in four different oscillator options. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

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

6.2.2 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 6-1). The PIC16C55X 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 6-2).

FIGURE 6-1: CRYSTAL OPERATION (OR CERAMIC RESONATOR) (HS, XT OR LP OSC CONFIGURATION)



Note 1: A series resistor may be required for AT strip cut crystals.
2: See Table 6-1 and Table 6-2 for recommended values of C1 and C2.

FIGURE 6-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

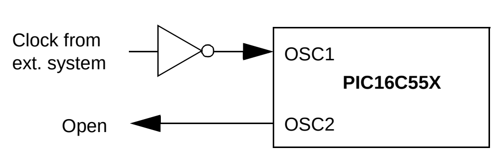


TABLE 6-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS (PRELIMINARY)

Ranges Characterized:			
Mode	**Freq**	**OSC1(C1)**	**OSC2(C2)**
XT	455 kHz 2.0 MHz 4.0 MHz	22 - 100 pF 15 - 68 pF 15 - 68 pF	22 - 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

Note 1: Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Since each resonator has its own characteristics, the user should consult with the resonator manufacturer for appropriate values of external components.

TABLE 6-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR (PRELIMINARY)

Mode	Freq	OSC1(C1)	OSC2(C2)
LP	32 kHz 200 kHz	68 - 100 pF 15 - 30 pF	68 - 100 pF 15 - 30 pF
XT	100 kHz 2 MHz 4 MHz	68 - 150 pF 15 - 30 pF 15 - 30 pF	150 - 200 pF 15 - 30 pF 15 - 30 pF
HS	8 MHz 10 MHz 20 MHz	15 - 30 pF 15 - 30 pF 15 - 30 pF	15 - 30 pF 15 - 30 pF 15 - 30 pF

Note 1: Higher capacitance increases the stability of the oscillator but also increases the start-up time. These values are for design guidance only. Rs may be required in HS mode as well as XT mode to avoid overdriving crystals with low-drive level specification. Since each crystal has its own characteristics, the user should consult with the crystal manufacturer for appropriate values of external components.

6.2.3 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

Either a pre-packaged 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 6-3 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 6-3: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT

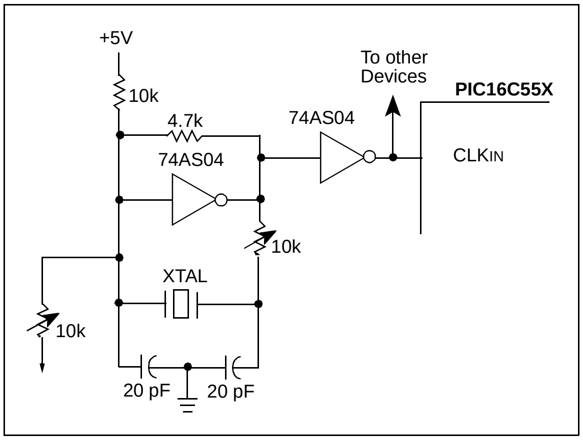
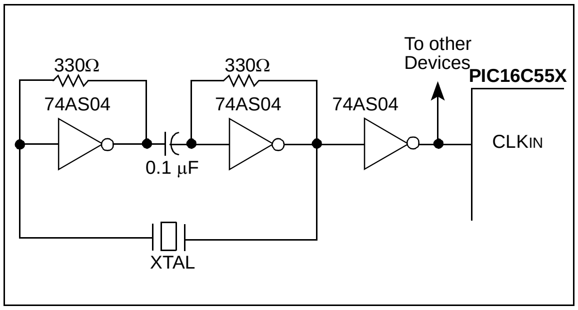


Figure 6-4 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180° phase shift in a series resonant oscillator circuit. The 330Ω resistors provide the negative feedback to bias the inverters in their linear region.

FIGURE 6-4: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT



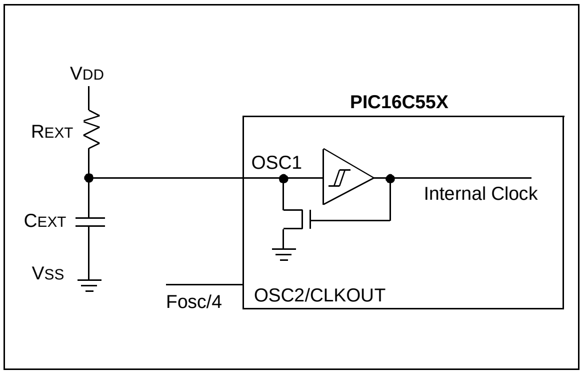
6.2.4 RC OSCILLATOR

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 6-5 shows how the R/C combination is connected to the PIC16C55X. For REXT values below 2.2 kΩ, the oscillator operation may become unstable, or stop completely. For very high REXT values (e.g., 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.

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 6-5: RC OSCILLATOR MODE



6.6 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 (e.g., W register and STATUS register). This will have to be implemented in software.

Example 6-1 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 0x20 in Bank 0 and it must also be defined at 0xA0 in Bank 1). The user register, STATUS_TEMP, must be defined in Bank 0. The Example 6-1:

- 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 6-1: 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 bank0
                     ;regardless of
                     ;current bank
MOVWF  STATUS_TEMP   ;save STATUS to bank0
                     ;register
:
:
:
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
```

6.7 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 has been stopped, for example, by execution of a SLEEP instruction. During normal operation, a WDT timeout generates a device RESET. If the device is in SLEEP mode, a WDT timeout 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 6.1).

6.7.1 WDT PERIOD

The WDT has a nominal timeout period of 18 ms, (with no prescaler). The timeout periods vary with temperature, VDD and process variations from part-to-part (see DC specs). If longer timeout 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, timeout periods up to 2.3 seconds can be realized.

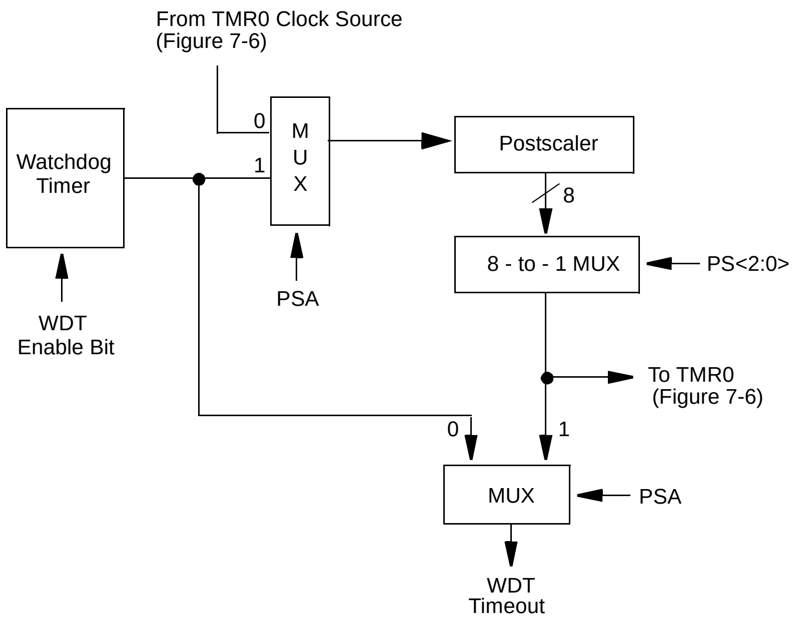
The CLRWDT and SLEEP instructions clear the WDT and the postscaler, if assigned to the WDT, and prevent it from timing out and generating a device RESET.

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

6.7.2 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 timeout occurs.

FIGURE 6-13: WATCHDOG TIMER BLOCK DIAGRAM



Note 1: T0SE, T0CS, PSA, PS0-PS2 are bits in the OPTION register.

TABLE 6-7: SUMMARY OF WATCHDOG TIMER REGISTERS

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
2007h	Config. bits	—	Reserved	CP1	CP0	PWRTE	WDTE	FOSC1	FOSC0		
81h	OPTION	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged, q = value depends on condition, — = unimplemented, read as ‘0’. Shaded cells are not used by the Watchdog Timer.

 Preliminary

MOVF	Move f			
Syntax:	[*label*] MOVF f,d			
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]			
Operation:	(f) → (dest)			
Status Affected:	Z			
Encoding:	00	1000	dfff	ffff
Description:	The contents of register f is 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 ≤ f ≤ 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 PIC MCU 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

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

RETLW — Return with Literal in W

Syntax: [*label*] RETLW k

Operands: 0 ≤ k ≤ 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
      •
      •
      ADDWF PC   ;W = offset
TABLE RETLW k1   ;Begin table
      RETLW k2   ;
      •
      •
      •
      RETLW kn   ; End of table
```

Before Instruction
W = 0x07

After Instruction
W = value of k8

RLF — Rotate Left f through Carry

Syntax: [*label*] RLF f,d

Operands: 0 ≤ f ≤ 127
d ∈ [0,1]

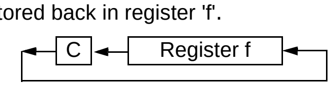
Operation: See description below

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

9.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK object linker combines relocatable objects created by the MPASM assembler and the MPLAB C17 and MPLAB C18 C compilers. It can also link relocatable objects from pre-compiled libraries, using directives from a linker script.

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

The MPLINK object linker features include:

- Integration with MPASM assembler and MPLAB C17 and MPLAB C18 C compilers.
- Allows all memory areas to be defined as sections to provide link-time flexibility.

The MPLIB object librarian features include:

- Easier linking because single libraries can be included instead of many smaller files.
- Helps keep code maintainable by grouping related modules together.
- Allows libraries to be created and modules to be added, listed, replaced, deleted or extracted.

9.5 MPLAB SIM Software Simulator

The MPLAB SIM software simulator allows code development in a PC-hosted environment by simulating the PIC 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.

The MPLAB SIM simulator fully supports symbolic debugging using the MPLAB C17 and the MPLAB C18 C compilers and the MPASM assembler. 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.

9.6 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 PIC microcontrollers (MCUs). Software control of the MPLAB ICE in-circuit emulator is provided by the MPLAB Integrated Development Environment (IDE), which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily re configured for emulation of different processors. The universal architecture of the MPLAB ICE in-circuit emulator allows expansion to support new PIC microcontrollers.

The MPLAB ICE in-circuit 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 environment were chosen to best make these features available to you, the end user.

9.7 ICEPIC In-Circuit Emulator

The ICEPIC low cost, in-circuit emulator is a 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.

9.13 PICDEM 3 Low Cost PIC16CXXX Demonstration Board

The PICDEM 3 demonstration board 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 an 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 demonstration board on a PRO MATE II device programmer, or a PICSTART Plus development programmer with an adapter socket, and easily test firmware. The MPLAB ICE in-circuit emulator may also be used with the PICDEM 3 demonstration board to test firmware. A prototype area has been provided to the user for adding 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 thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM 3 demonstration board is a LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM 3 demonstration board provides an additional RS-232 interface and Windows 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.

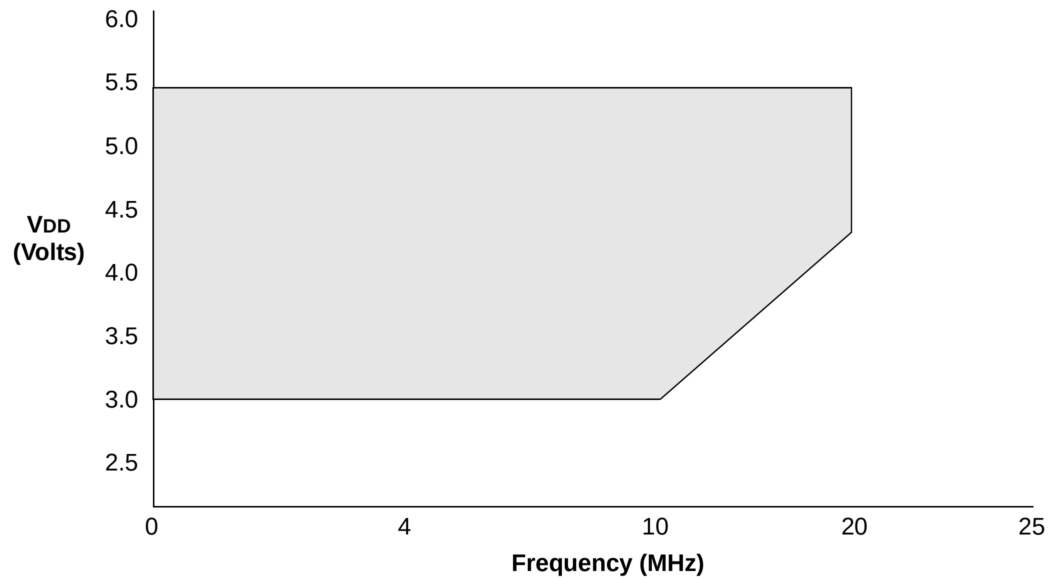
9.14 PICDEM 17 Demonstration Board

The PICDEM 17 demonstration board is an evaluation board that demonstrates the capabilities of several Microchip microcontrollers, including PIC17C752, PIC17C756A, 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 device programmer, or the PICSTART Plus development programmer, and easily debug and test the sample code. In addition, the PICDEM 17 demonstration board supports downloading of programs to and executing out of external FLASH memory on board. The PICDEM 17 demonstration board is also usable with the MPLAB ICE in-circuit emulator, or the 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.

9.15 KEELOQ Evaluation and Programming Tools

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

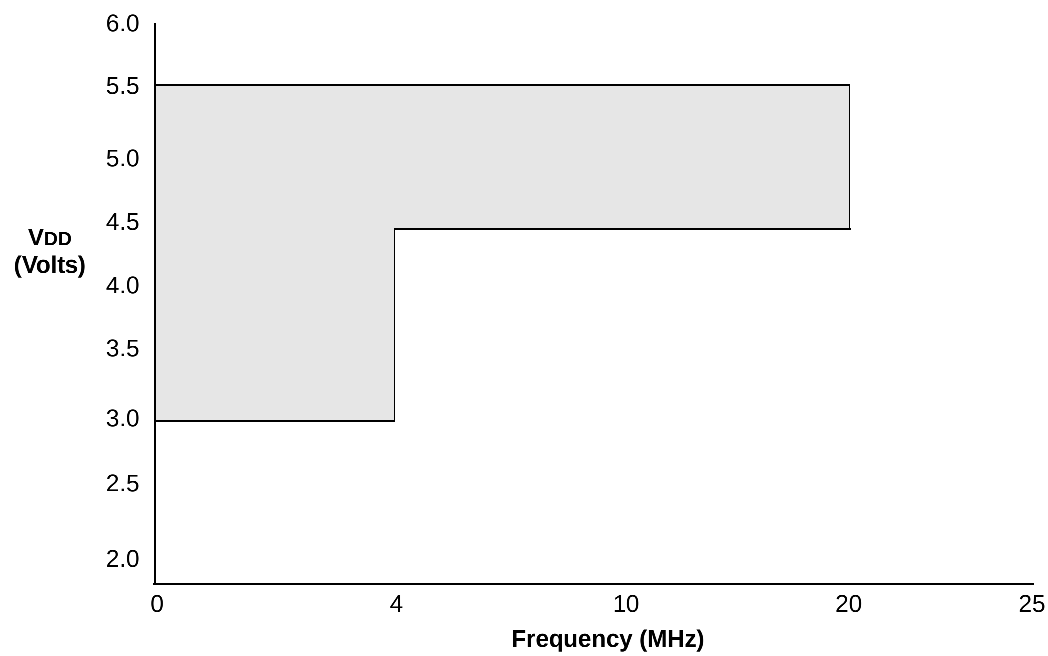
FIGURE 10-1: VOLTAGE-FREQUENCY GRAPH, 0°C ≤ T_A ≤ +70°C (COMMERCIAL TEMPS)



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 10-2: VOLTAGE-FREQUENCY GRAPH, -40°C ≤ T_A < 0°C, +70°C < T_A ≤ +125°C (OUTSIDE OF COMMERCIAL TEMPS)



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.

10.2 DC Characteristics: PIC16C55X (Commercial, Industrial, Extended) PIC16LC55X(Commercial, Industrial, Extended) (Continued)

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 automotive Operating voltage VDD range as described in DC spec Table 10-1				
Param. No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Unit**	**Conditions**
			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
*	VOD	**Open-Drain High Voltage**			10*	V	RA4 pin
		Capacitive Loading Specs on Output Pins					
D100	COSC 2	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.

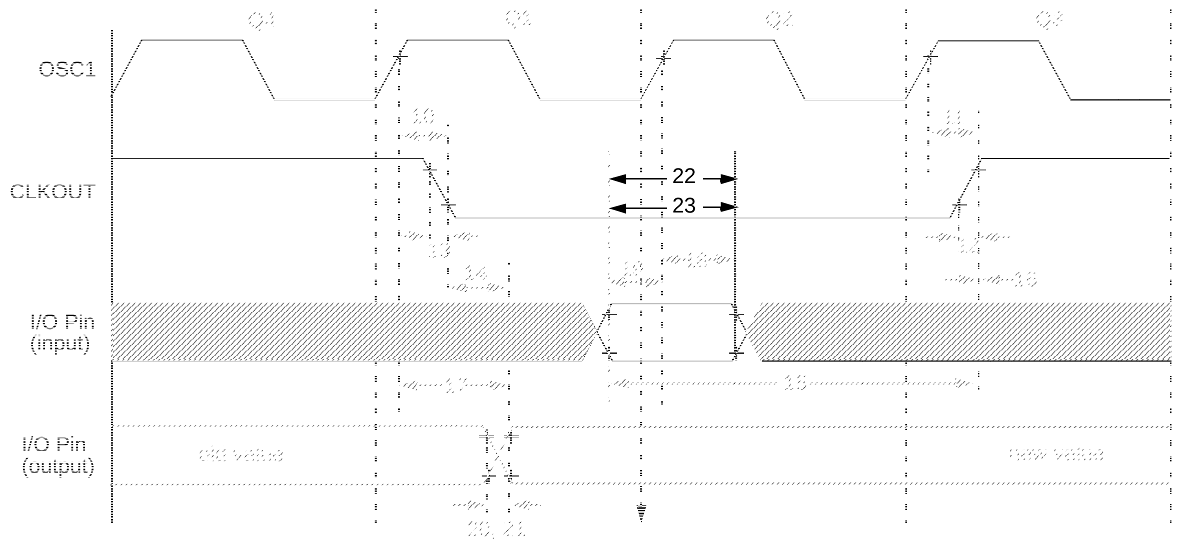
† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt Trigger input. It is not recommended that the PIC16C55X 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.

FIGURE 10-7: CLKOUT AND I/O TIMING

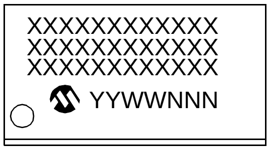


Note 1: All tests must be done with specified capacitance loads (Figure 10-5) 50 pF on I/O pins and CLKOUT.

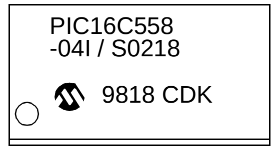
 Preliminary

Package Marking Information (Cont'd)

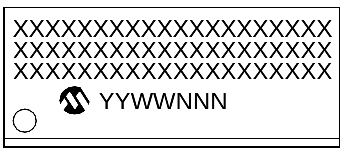
18-Lead SOIC (.300")



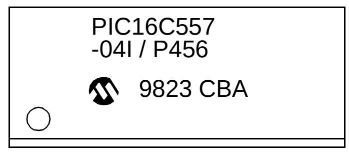
Example



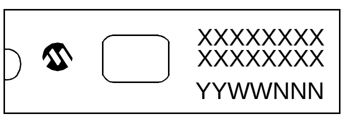
28-Lead SOIC (.300")



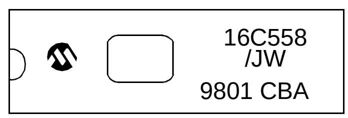
Example



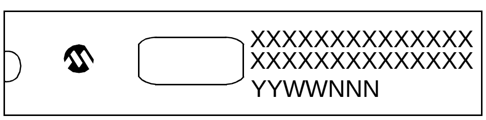
18-Lead CERDIP Windowed



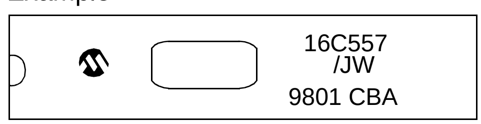
Example



28-Lead CERDIP Windowed

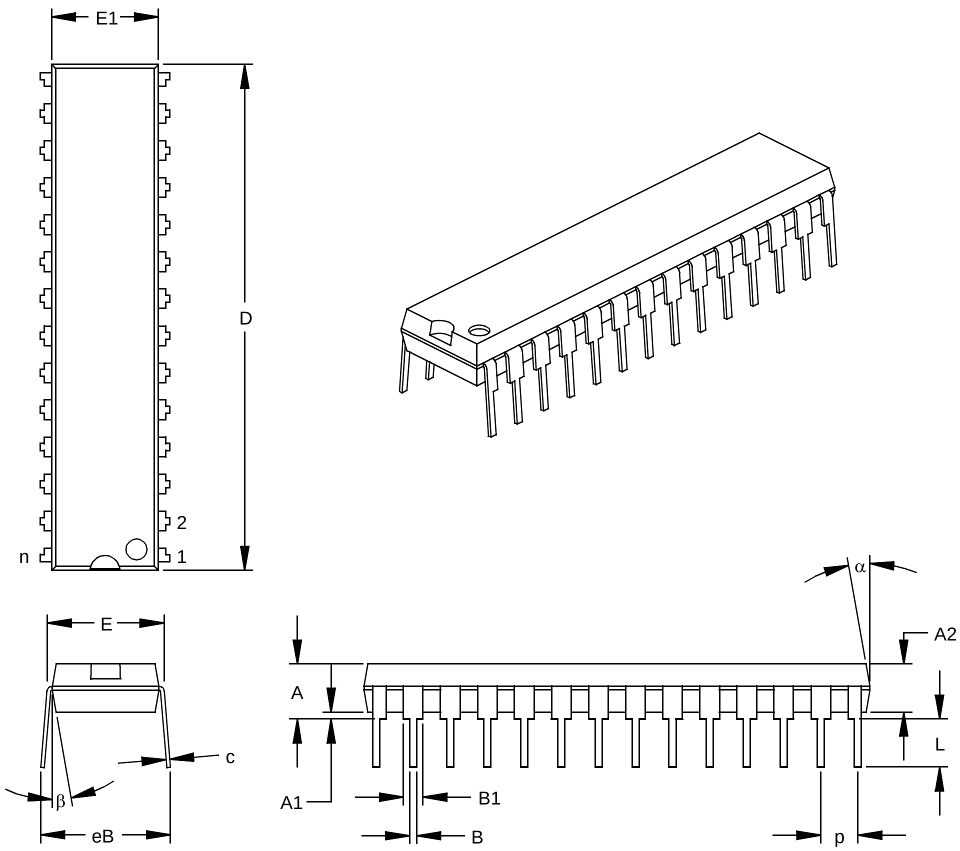


Example



28-Lead Skinny Plastic Dual In-line (SP) – 300 mil (PDIP)

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		28			28	
Pitch	p		.100			2.54	
Top to Seating Plane	A	.140	.150	.160	3.56	3.81	4.06
Molded Package Thickness	A2	.125	.130	.135	3.18	3.30	3.43
Base to Seating Plane	A1	.015			0.38		
Shoulder to Shoulder Width	E	.300	.310	.325	7.62	7.87	8.26
Molded Package Width	E1	.275	.285	.295	6.99	7.24	7.49
Overall Length	D	1.345	1.365	1.385	34.16	34.67	35.18
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	.040	.053	.065	1.02	1.33	1.65
Lower Lead Width	B	.016	.019	.022	0.41	0.48	0.56
Overall Row Spacing §	eB	.320	.350	.430	8.13	8.89	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
§ Significant Characteristic
Notes:
Dimension 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-095
Drawing No. C04-070

Preliminary