



Welcome to [E-XFL.COM](https://www.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	16MHz
Connectivity	UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	33
Program Memory Size	16KB (8K x 16)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	454 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 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	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic17c44-16i-p">https://www.e-xfl.com/product-detail/microchip-technology/pic17c44-16i-p</a>

## 2.0 PIC17C4X DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. Depending on application and production requirements, the proper device option can be selected using the information in the PIC17C4X Product Selection System section at the end of this data sheet. When placing orders, please use the "PIC17C4X Product Identification System" at the back of this data sheet to specify the correct part number.

For the PIC17C4X family of devices, there are four device "types" as indicated in the device number:

1. **C**, as in PIC17**C**42. These devices have EPROM type memory and operate over the standard voltage range.
2. **LC**, as in PIC17**LC**42. These devices have EPROM type memory, operate over an extended voltage range, and reduced frequency range.
3. **CR**, as in PIC17**CR**42. These devices have ROM type memory and operate over the standard voltage range.
4. **LCR**, as in PIC17**LCR**42. These devices have ROM type memory, operate over an extended voltage range, and reduced frequency range.

### 2.1 UV Erasable Devices

The UV erasable version, offered in CERDIP package, is optimal for prototype development and pilot programs.

The UV erasable version can be erased and reprogrammed to any of the configuration modes. Microchip's PRO MATE™ programmer supports programming of the PIC17C4X. Third party programmers also are available; refer to the *Third Party Guide* for a list of sources.

### 2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers expecting frequent code changes and updates.

The OTP devices, packaged in plastic packages, permit the user to program them once. In addition to the program memory, the configuration bits must also be programmed.

### 2.3 Quick-Turnaround-Production (QTP) Devices

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

### 2.4 Serialized Quick-Turnaround Production (SQTP<sup>SM</sup>) Devices

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

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

ROM devices do not allow serialization information in the program memory space.

For information on submitting ROM code, please contact your regional sales office.

### 2.5 Read Only Memory (ROM) Devices

Microchip offers masked ROM versions of several of the highest volume parts, thus giving customers a low cost option for high volume, mature products.

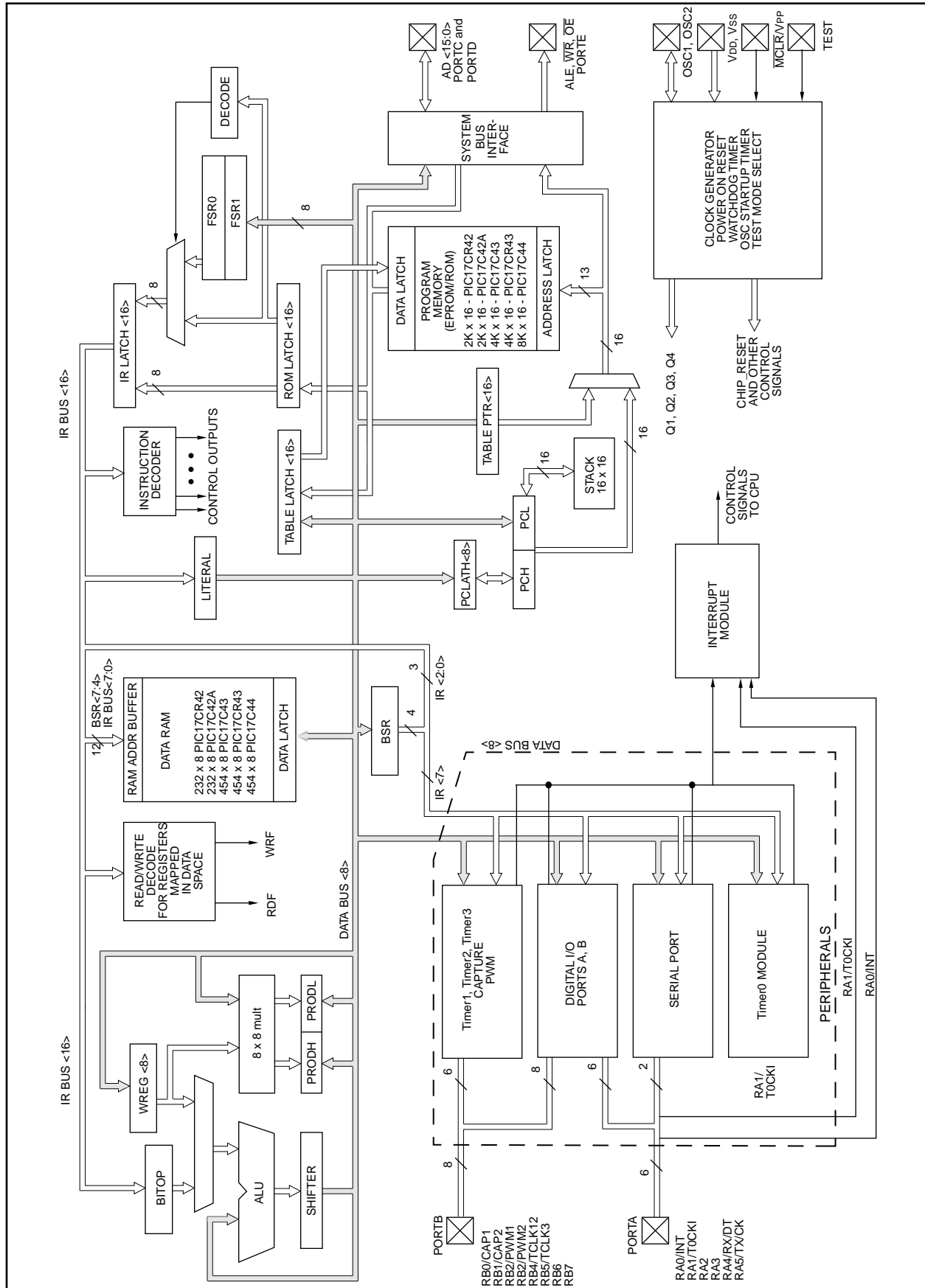
For information on submitting ROM code, please contact your regional sales office.

# PIC17C4X

---

NOTES:

FIGURE 3-2: PIC17CR42/42A/43/R43/44 BLOCK DIAGRAM



## 3.1 Clocking Scheme/Instruction Cycle

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

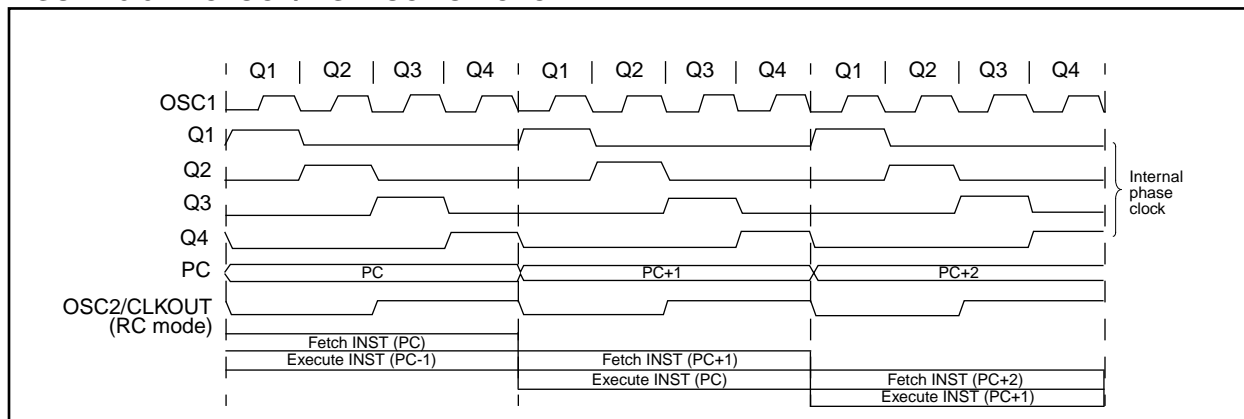
## 3.2 Instruction Flow/Pipelining

An "Instruction Cycle" consists of four Q cycles (Q1, Q2, Q3, and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle while decode and execute takes another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g. GOTO) then two cycles are required to complete the instruction (Example 3-2).

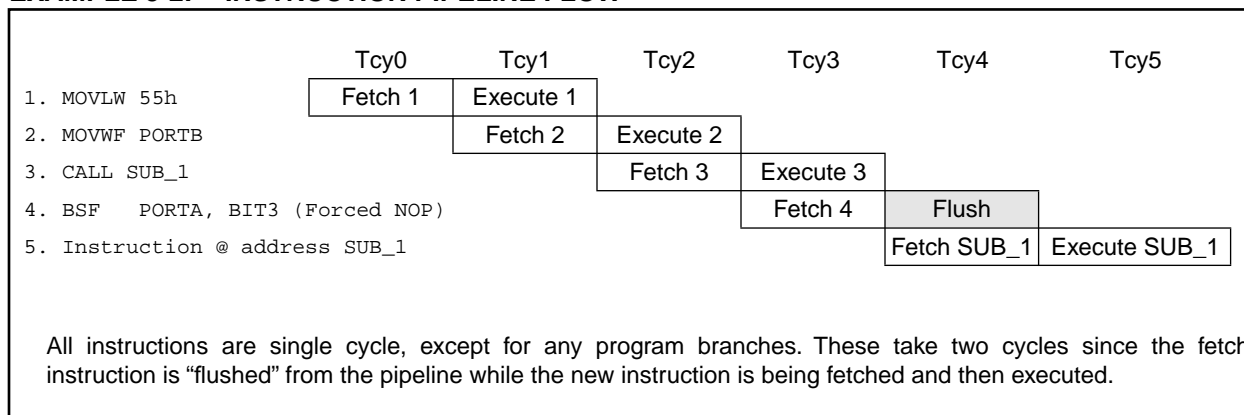
A fetch cycle begins with the program counter incrementing in Q1.

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

**FIGURE 3-3: CLOCK/INSTRUCTION CYCLE**



**EXAMPLE 3-2: INSTRUCTION PIPELINE FLOW**



## 4.1.3 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides a 1024 oscillator cycle (1024Tosc) delay after  $\overline{\text{MCLR}}$  is detected high or a wake-up from SLEEP event occurs.

The OST time-out is invoked only for XT and LF oscillator modes on a Power-on Reset or a Wake-up from SLEEP.

The OST counts the oscillator pulses on the OSC1/CLKIN pin. The counter only starts incrementing after the amplitude of the signal reaches the oscillator input thresholds. This delay allows the crystal oscillator or resonator to stabilize before the device exits reset. The length of time-out is a function of the crystal/resonator frequency.

## 4.1.4 TIME-OUT SEQUENCE

On power-up the time-out sequence is as follows: First the internal POR signal goes high when the POR trip point is reached. If  $\overline{\text{MCLR}}$  is high, then both the OST and PWRT timers start. In general the PWRT time-out is longer, except with low frequency crystals/resonators. The total time-out also varies based on oscillator configuration. Table 4-1 shows the times that are associated with the oscillator configuration. Figure 4-2 and Figure 4-3 display these time-out sequences.

If the device voltage is not within electrical specification at the end of a time-out, the  $\overline{\text{MCLR}}/\text{VPP}$  pin must be held low until the voltage is within the device specification. The use of an external RC delay is sufficient for many of these applications.

**TABLE 4-1: TIME-OUT IN VARIOUS SITUATIONS**

Oscillator Configuration	Power-up	Wake up from SLEEP	MCLR Reset
XT, LF	Greater of: 96 ms or 1024Tosc	1024Tosc	—
EC, RC	Greater of: 96 ms or 1024Tosc	—	—

The time-out sequence begins from the first rising edge of  $\overline{\text{MCLR}}$ .

Table 4-3 shows the reset conditions for some special registers, while Table 4-4 shows the initialization conditions for all the registers. The shaded registers (in Table 4-4) are for all devices except the PIC17C42. In the PIC17C42, the PRODH and PRODL registers are general purpose RAM.

**TABLE 4-2: STATUS BITS AND THEIR SIGNIFICANCE**

$\overline{\text{TO}}$	$\overline{\text{PD}}$	Event
1	1	Power-on Reset, $\overline{\text{MCLR}}$ Reset during normal operation, or CLRWDT instruction executed
1	0	$\overline{\text{MCLR}}$ Reset during SLEEP or interrupt wake-up from SLEEP
0	1	WDT Reset during normal operation
0	0	WDT Reset during SLEEP

In Figure 4-2, Figure 4-3 and Figure 4-4,  $\text{TPWRT} > \text{TOST}$ , as would be the case in higher frequency crystals. For lower frequency crystals, (i.e., 32 kHz)  $\text{TOST}$  would be greater.

**TABLE 4-3: RESET CONDITION FOR THE PROGRAM COUNTER AND THE CPUSTA REGISTER**

Event		PCH:PCL	CPUSTA	OST Active
Power-on Reset		0000h	--11 11--	Yes
$\overline{\text{MCLR}}$ Reset during normal operation		0000h	--11 11--	No
$\overline{\text{MCLR}}$ Reset during SLEEP		0000h	--11 10--	Yes <sup>(2)</sup>
WDT Reset during normal operation		0000h	--11 01--	No
WDT Reset during SLEEP <sup>(3)</sup>		0000h	--11 00--	Yes <sup>(2)</sup>
Interrupt wake-up from SLEEP	GLINTD is set	PC + 1	--11 10--	Yes <sup>(2)</sup>
	GLINTD is clear	PC + 1 <sup>(1)</sup>	--10 10--	Yes <sup>(2)</sup>

Legend: u = unchanged, x = unknown, - = unimplemented read as '0'.

Note 1: On wake-up, this instruction is executed. The instruction at the appropriate interrupt vector is fetched and then executed.

2: The OST is only active when the Oscillator is configured for XT or LF modes.

3: The Program Counter = 0, that is the device branches to the reset vector. This is different from the mid-range devices.

## 6.4.1 INDIRECT ADDRESSING REGISTERS

The PIC17C4X has four registers for indirect addressing. These registers are:

- INDF0 and FSR0
- INDF1 and FSR1

Registers INDF0 and INDF1 are not physically implemented. Reading or writing to these registers activates indirect addressing, with the value in the corresponding FSR register being the address of the data. The FSR is an 8-bit register and allows addressing anywhere in the 256-byte data memory address range. For banked memory, the bank of memory accessed is specified by the value in the BSR.

If file INDF0 (or INDF1) itself is read indirectly via an FSR, all '0's are read (Zero bit is set). Similarly, if INDF0 (or INDF1) is written to indirectly, the operation will be equivalent to a NOP, and the status bits are not affected.

## 6.4.2 INDIRECT ADDRESSING OPERATION

The indirect addressing capability has been enhanced over that of the PIC16CXX family. There are two control bits associated with each FSR register. These two bits configure the FSR register to:

- Auto-decrement the value (address) in the FSR after an indirect access
- Auto-increment the value (address) in the FSR after an indirect access
- No change to the value (address) in the FSR after an indirect access

These control bits are located in the ALUSTA register. The FSR1 register is controlled by the FS3:FS2 bits and FSR0 is controlled by the FS1:FS0 bits.

When using the auto-increment or auto-decrement features, the effect on the FSR is not reflected in the ALUSTA register. For example, if the indirect address causes the FSR to equal '0', the Z bit will not be set.

If the FSR register contains a value of 0h, an indirect read will read 0h (Zero bit is set) while an indirect write will be equivalent to a NOP (status bits are not affected).

Indirect addressing allows single cycle data transfers within the entire data space. This is possible with the use of the MOVPPF and MOVFP instructions, where either 'p' or 'f' is specified as INDF0 (or INDF1).

If the source or destination of the indirect address is in banked memory, the location accessed will be determined by the value in the BSR.

A simple program to clear RAM from 20h - FFh is shown in Example 6-1.

### EXAMPLE 6-1: INDIRECT ADDRESSING

```
MOVLW    0x20      ;
MOVWF    FSR0      ; FSR0 = 20h
BCF      ALUSTA, FS1 ; Increment FSR
BSF      ALUSTA, FS0 ; after access
BCF      ALUSTA, C   ; C = 0
MOVLW    END_RAM + 1 ;
LP CLRf    INDF0      ; Addr(FSR) = 0
CPFSEQ   FSR0        ; FSR0 = END_RAM+1?
GOTO     LP          ; NO, clear next
:        :            ; YES, All RAM is
:        :            ; cleared
```

## 6.5 Table Pointer (TBLPTRL and TBLPTRH)

File registers TBLPTRL and TBLPTRH form a 16-bit pointer to address the 64K program memory space. The table pointer is used by instructions TABLWT and TABLRD.

The TABLRD and the TABLWT instructions allow transfer of data between program and data space. The table pointer serves as the 16-bit address of the data word within the program memory. For a more complete description of these registers and the operation of Table Reads and Table Writes, see Section 7.0.

## 6.6 Table Latch (TBLATH, TBLATL)

The table latch (TBLAT) is a 16-bit register, with TBLATH and TBLATL referring to the high and low bytes of the register. It is not mapped into data or program memory. The table latch is used as a temporary holding latch during data transfer between program and data memory (see descriptions of instructions TABLRD, TABLWT, TLRD and TLWT). For a more complete description of these registers and the operation of Table Reads and Table Writes, see Section 7.0.

Example 9-1 shows the instruction sequence to initialize PORTB. The Bank Select Register (BSR) must be selected to Bank 0 for the port to be initialized.

## EXAMPLE 9-1: INITIALIZING PORTB

```
MOVLB 0      ; Select Bank 0
CLRF  PORTB  ; Initialize PORTB by clearing
              ; output data latches
MOVLW 0xCF   ; Value used to initialize
              ; data direction
MOVWF DDRB   ; Set RB<3:0> as inputs
              ; RB<5:4> as outputs
              ; RB<7:6> as inputs
```

**TABLE 9-3: PORTB FUNCTIONS**

Name	Bit	Buffer Type	Function
RB0/CAP1	bit0	ST	Input/Output or the RB0/CAP1 input pin. Software programmable weak pull-up and interrupt on change features.
RB1/CAP2	bit1	ST	Input/Output or the RB1/CAP2 input pin. Software programmable weak pull-up and interrupt on change features.
RB2/PWM1	bit2	ST	Input/Output or the RB2/PWM1 output pin. Software programmable weak pull-up and interrupt on change features.
RB3/PWM2	bit3	ST	Input/Output or the RB3/PWM2 output pin. Software programmable weak pull-up and interrupt on change features.
RB4/TCLK12	bit4	ST	Input/Output or the external clock input to Timer1 and Timer2. Software programmable weak pull-up and interrupt on change features.
RB5/TCLK3	bit5	ST	Input/Output or the external clock input to Timer3. Software programmable weak pull-up and interrupt on change features.
RB6	bit6	ST	Input/Output pin. Software programmable weak pull-up and interrupt on change features.
RB7	bit7	ST	Input/Output pin. Software programmable weak pull-up and interrupt on change features.

Legend: ST = Schmitt Trigger input.

**TABLE 9-4: REGISTERS/BITS ASSOCIATED WITH PORTB**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note1)
12h, Bank 0	PORTB	PORTB data latch								xxxx xxxx	uuuu uuuu
11h, Bank 0	DDRB	Data direction register for PORTB								1111 1111	1111 1111
10h, Bank 0	PORTA	RBPU	—	RA5	RA4	RA3	RA2	RA1/T0CKI	RA0/INT	0-xx xxxx	0-uu uuuu
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	T0	PD	—	—	--11 11--	--11 qq--
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
16h, Bank 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/PR3	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000

Legend: x = unknown, u = unchanged, - = unimplemented read as '0', q = Value depends on condition.

Shaded cells are not used by PORTB.

Note 1: Other (non power-up) resets include: external reset through  $\overline{\text{MCLR}}$  and the Watchdog Timer Reset.



## 9.4.1 PORTE AND DDRE REGISTER

PORTE is a 3-bit bi-directional port. The corresponding data direction register is DDRE. A '1' in DDRE configures the corresponding port pin as an input. A '0' in the DDRE register configures the corresponding port pin as an output. Reading PORTE reads the status of the pins, whereas writing to it will write to the port latch. PORTE is multiplexed with the system bus. When operating as the system bus, PORTE contains the control signals for the address/data bus (AD15:AD0). These control signals are Address Latch Enable (ALE), Output Enable ( $\overline{OE}$ ), and Write ( $\overline{WR}$ ). The control signals  $\overline{OE}$  and  $\overline{WR}$  are active low signals. The timing for the system bus is shown in the Electrical Characteristics section.

**Note:** This port is configured as the system bus when the device's configuration bits are selected to Microprocessor or Extended Microcontroller modes. In the two other microcontroller modes, this port is a general purpose I/O.

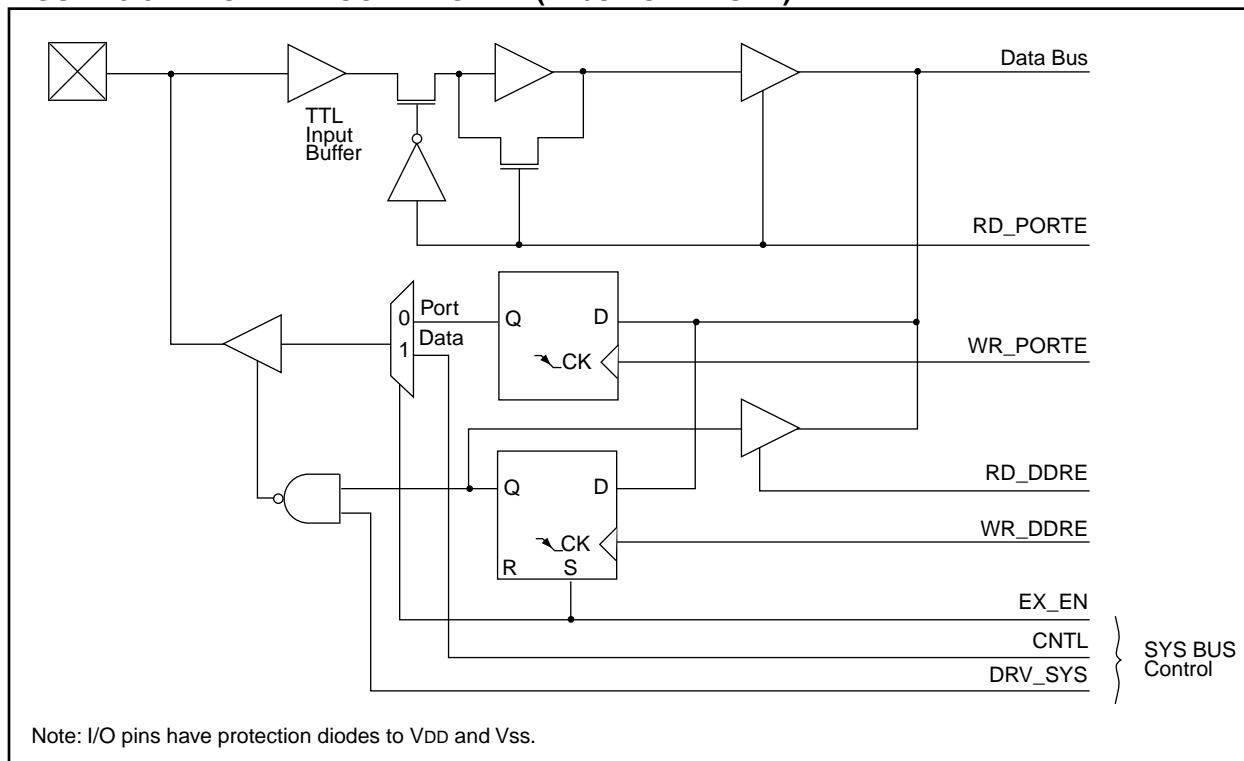
Example 9-4 shows the instruction sequence to initialize PORTE. The Bank Select Register (BSR) must be selected to Bank 1 for the port to be initialized.

### EXAMPLE 9-4: INITIALIZING PORTE

```

MOVLB 1           ; Select Bank 1
CLRF  PORTE       ; Initialize PORTE data
                  ; latches before setting
                  ; the data direction
                  ; register
MOVLW 0x03        ; Value used to initialize
                  ; data direction
MOVWF DDRE        ; Set RE<1:0> as inputs
                  ; RE<2> as outputs
                  ; RE<7:3> are always
                  ; read as '0'
    
```

**FIGURE 9-8: PORTE BLOCK DIAGRAM (IN I/O PORT MODE)**



## 12.2.2 DUAL CAPTURE REGISTER MODE

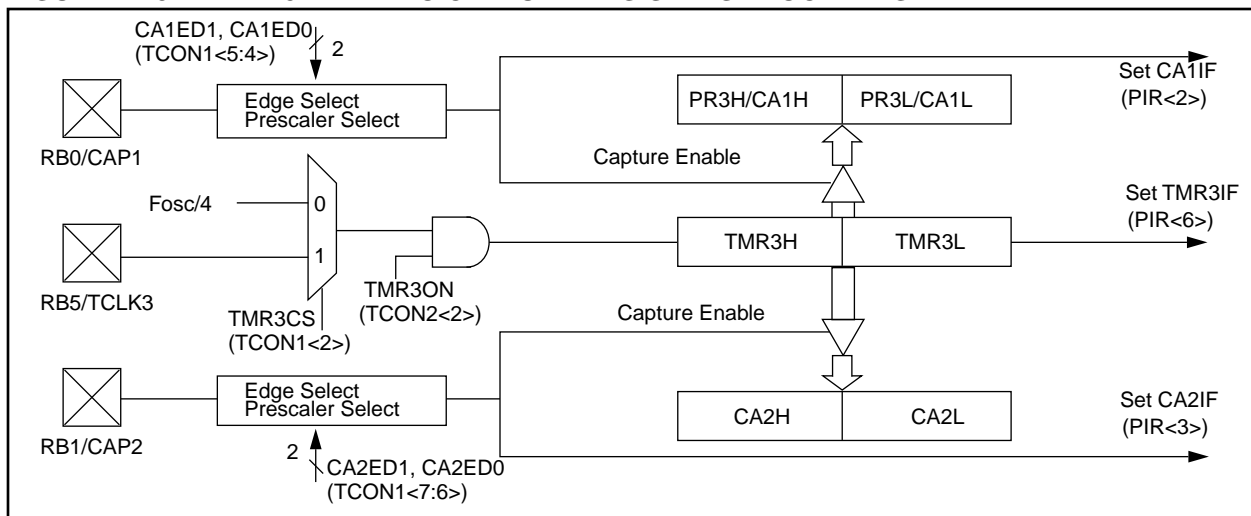
This mode is selected by setting CA1/PR3. A block diagram is shown in Figure 12-8. In this mode, TMR3 runs without a period register and increments from 0000h to FFFFh and rolls over to 0000h. The TMR3 interrupt Flag (TMR3IF) is set on this roll over. The TMR3IF bit must be cleared in software.

Registers PR3H/CA1H and PR3L/CA1L make a 16-bit capture register (Capture1). It captures events on pin RB0/CAP1. Capture mode is configured by the CA1ED1 and CA1ED0 bits. Capture1 Interrupt Flag bit (CA1IF) is set on the capture event. The corresponding interrupt mask bit is CA1IE. The Capture1 Overflow Status bit is CA1OVF.

The Capture2 overflow status flag bit is double buffered. The master bit is set if one captured word is already residing in the Capture2 register and another "event" has occurred on the RB1/CA2 pin. The new event will not transfer the TMR3 value to the capture register which protects the previous unread capture value. When the user reads both the high and the low bytes (in any order) of the Capture2 register, the master overflow bit is transferred to the slave overflow bit (CA2OVF) and then the master bit is reset. The user can then read TCON2 to determine the value of CA2OVF.

The operation of the Capture1 feature is identical to Capture2 (as described in Section 12.2.1).

**FIGURE 12-8: TIMER3 WITH TWO CAPTURE REGISTERS BLOCK DIAGRAM**



**TABLE 12-5: REGISTERS ASSOCIATED WITH CAPTURE**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note1)
16h, Bank 3	TCON1	CA2ED1	CA2ED0	CA1ED1	CA1ED0	T16	TMR3CS	TMR2CS	TMR1CS	0000 0000	0000 0000
17h, Bank 3	TCON2	CA2OVF	CA1OVF	PWM2ON	PWM1ON	CA1/PR3	TMR3ON	TMR2ON	TMR1ON	0000 0000	0000 0000
12h, Bank 2	TMR3L	TMR3 register; low byte								xxxx xxxx	uuuu uuuu
13h, Bank 2	TMR3H	TMR3 register; high byte								xxxx xxxx	uuuu uuuu
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
07h, Unbanked	INTSTA	PEIF	T0CKIF	T0IF	INTF	PEIE	T0CKIE	T0IE	INTE	0000 0000	0000 0000
06h, Unbanked	CPUSTA	—	—	STKAV	GLINTD	T0	PD	—	—	--11 11--	--11 qq--
16h, Bank 2	PR3L/CA1L	Timer3 period register, low byte/capture1 register, low byte								xxxx xxxx	uuuu uuuu
17h, Bank 2	PR3H/CA1H	Timer3 period register, high byte/capture1 register, high byte								xxxx xxxx	uuuu uuuu
14h, Bank 3	CA2L	Capture2 low byte								xxxx xxxx	uuuu uuuu
15h, Bank 3	CA2H	Capture2 high byte								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, — = unimplemented read as '0', q - value depends on condition, shaded cells are not used by Capture.

Note 1: Other (non power-up) resets include: external reset through MCLR and WDT Timer Reset.

# PIC17C4X

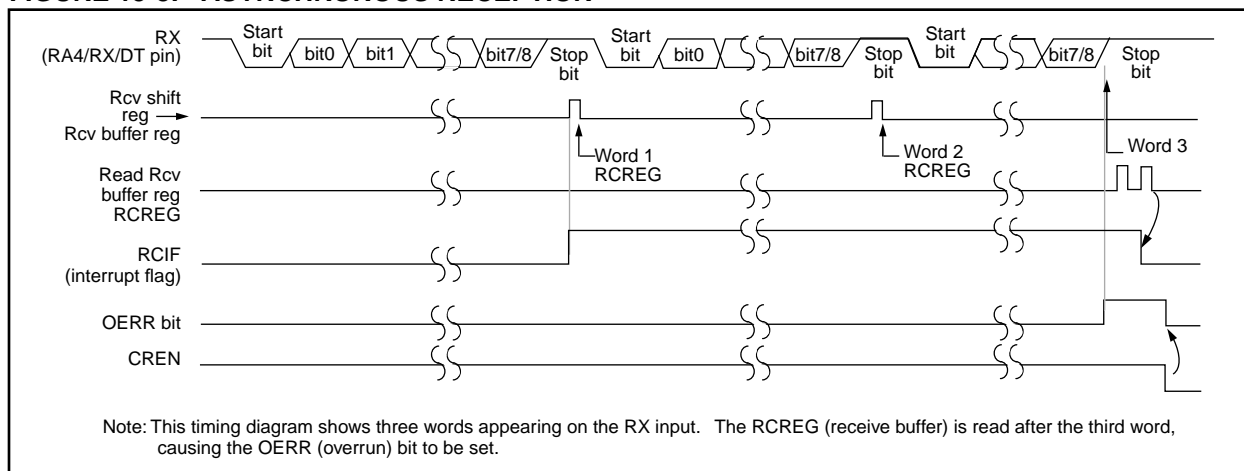
Steps to follow when setting up an Asynchronous Reception:

1. Initialize the SPBRG register for the appropriate baud rate.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are desired, then set the RCIE bit.
4. If 9-bit reception is desired, then set the RX9 bit.
5. Enable the reception by setting the CREN bit.
6. The RCIF bit will be set when reception completes and an interrupt will be generated if the RCIE bit was set.

7. Read RCSTA to get the ninth bit (if enabled) and FERR bit to determine if any error occurred during reception.
8. Read RCREG for the 8-bit received data.
9. If an overrun error occurred, clear the error by clearing the OERR bit.

**Note:** To terminate a reception, either clear the SREN and CREN bits, or the SPEN bit. This will reset the receive logic, so that it will be in the proper state when receive is re-enabled.

**FIGURE 13-8: ASYNCHRONOUS RECEPTION**



**TABLE 13-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note1)
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
13h, Bank 0	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00u
14h, Bank 0	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	xxxx xxxx	uuuu uuuu
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
15h, Bank 0	TXSTA	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D	0000 --1x	0000 --1u
17h, Bank 0	SPBRG	Baud rate generator register								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', shaded cells are not used for asynchronous reception.

Note 1: Other (non power-up) resets include: external reset through MCLR and Watchdog Timer Reset.

**TABLE 13-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note1)
16h, Bank 1	PIR	RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF	0000 0010	0000 0010
13h, Bank 0	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00u
14h, Bank 0	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	xxxx xxxx	uuuu uuuu
17h, Bank 1	PIE	RBIE	TMR3IE	TMR2IE	TMR1IE	CA2IE	CA1IE	TXIE	RCIE	0000 0000	0000 0000
15h, Bank 0	TXSTA	CSRC	TX9	TXEN	SYNC	—	—	TRMT	TX9D	0000 --1x	0000 --1u
17h, Bank 0	SPBRG	Baud rate generator register								xxxx xxxx	uuuu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented read as a '0', shaded cells are not used for synchronous master reception.

Note 1: Other (non power-up) resets include: external reset through  $\overline{\text{MCLR}}$  and Watchdog Timer Reset.

## 14.2.4 EXTERNAL CRYSTAL OSCILLATOR CIRCUIT

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

Figure 14-5 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-degree phase shift that a parallel oscillator requires. The 4.7 k $\Omega$  resistor provides the negative feedback for stability. The 10 k $\Omega$  potentiometer biases the 74AS04 in the linear region. This could be used for external oscillator designs.

**FIGURE 14-5: EXTERNAL PARALLEL RESONANT CRYSTAL OSCILLATOR CIRCUIT**

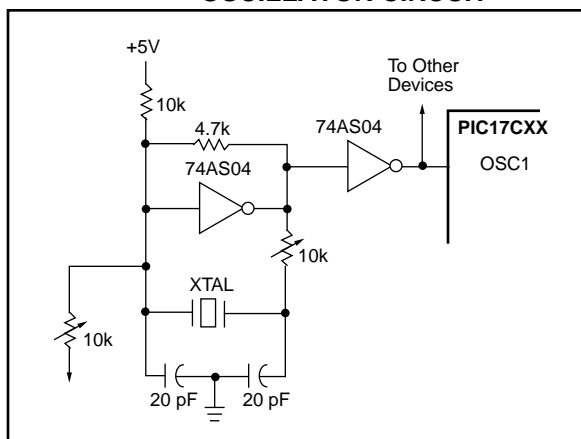
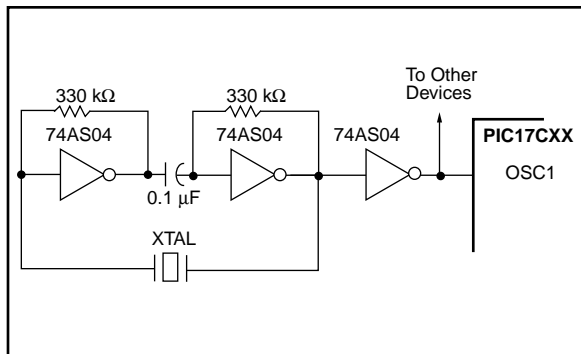


Figure 14-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 k $\Omega$  resistors provide the negative feedback to bias the inverters in their linear region.

**FIGURE 14-6: EXTERNAL SERIES RESONANT CRYSTAL OSCILLATOR CIRCUIT**



## 14.2.5 RC OSCILLATOR

For timing insensitive applications, the RC device option offers additional cost savings. RC oscillator frequency is a function of the supply voltage, the resistor ( $R_{ext}$ ) and capacitor ( $C_{ext}$ ) values, and the operating temperature. In addition to this, 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 oscillation frequency, especially for low  $C_{ext}$  values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 14-6 shows how the R/C combination is connected to the PIC17CXX. For  $R_{ext}$  values below 2.2 k $\Omega$ , the oscillator operation may become unstable, or stop completely. For very high  $R_{ext}$  values (e.g. 1 M $\Omega$ ), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend to keep  $R_{ext}$  between 3 k $\Omega$  and 100 k $\Omega$ .

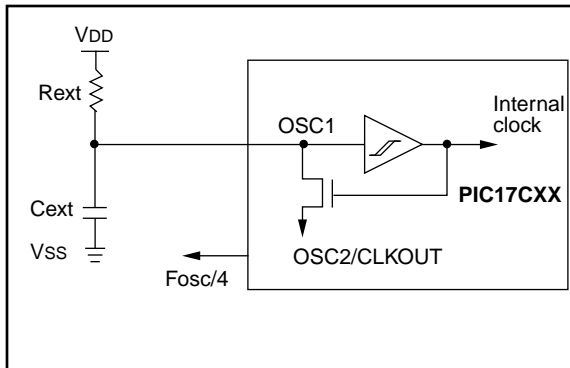
Although the oscillator will operate with no external capacitor ( $C_{ext} = 0$  pF), we recommend using values above 20 pF for noise and stability reasons. With little or no external capacitance, oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance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

See Section 18.0 for RC frequency variation from part to part due to normal process variation. The variation is larger for larger R (since leakage current variation will affect RC frequency more for large R) and for smaller C (since variation of input capacitance will affect RC frequency more).

See Section 18.0 for variation of oscillator frequency due to  $V_{DD}$  for given  $R_{ext}/C_{ext}$  values as well as frequency variation due to operating temperature for given R, C, and  $V_{DD}$  values.

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

**FIGURE 14-7: RC OSCILLATOR MODE**



BSF		Bit Set f			
Syntax:	[ <i>label</i> ] BSF f,b				
Operands:	$0 \leq f \leq 255$ $0 \leq b \leq 7$				
Operation:	$1 \rightarrow (f < b)$				
Status Affected:	None				
Encoding:	1000	0bbb	ffff	ffff	
Description:	Bit 'b' in register 'f' is set.				
Words:	1				
Cycles:	1				
Q Cycle Activity:					

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Execute	Write register 'f'

**Example:** BSF FLAG\_REG, 7

Before Instruction  
FLAG\_REG= 0x0A

After Instruction  
FLAG\_REG= 0x8A

BTFSC		Bit Test, skip if Clear						
Syntax:	[ <i>label</i> ] BTFSC f,b							
Operands:	$0 \leq f \leq 255$ $0 \leq b \leq 7$							
Operation:	skip if (f<b>) = 0							
Status Affected:	None							
Encoding:	<table border="1"><tr><td>1001</td><td>1bbb</td><td>ffff</td><td>ffff</td></tr></table>				1001	1bbb	ffff	ffff
1001	1bbb	ffff	ffff					
Description:	If bit 'b' in register 'f' is 0 then the next instruction is skipped.  If bit 'b' is 0 then the next instruction fetched during the current instruction execution is discarded, and a NOP is executed instead, making this a two-cycle instruction.							

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Execute	NOP

If skip:

Q1	Q2	Q3	Q4
Forced NOP	NOP	Execute	NOP

**Example:** HERE BTFSC FLAG, 1  
FALSE :  
TRUE :

Before Instruction  
PC = address (HERE)

After Instruction  
If FLAG<1> = 0;  
PC = address (TRUE)  
If FLAG<1> = 1;  
PC = address (FALSE)

## CPFSLT Compare f with WREG, skip if f < WREG

**Syntax:** [label] CPFSLT f

**Operands:**  $0 \leq f \leq 255$

**Operation:** (f) – (WREG), skip if (f) < (WREG) (unsigned comparison)

**Status Affected:** None

**Encoding:**

0011	0000	ffff	ffff
------	------	------	------

**Description:** Compares the contents of data memory location 'f' to the contents of WREG by performing an unsigned subtraction. If the contents of 'f' < the contents of WREG, then the fetched instruction is discarded and an NOP is executed instead making this a two-cycle instruction.

**Words:** 1

**Cycles:** 1 (2)

**Q Cycle Activity:**

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Execute	NOP

**If skip:**

Q1	Q2	Q3	Q4
Forced NOP	NOP	Execute	NOP

**Example:**

```

HERE    CPFSLT REG
NLESS   :
LESS    :
```

**Before Instruction**

```

PC      = Address (HERE)
W       = ?
```

**After Instruction**

```

If REG < WREG;
PC      = Address (LESS)
If REG ≥ WREG;
PC      = Address (NLESS)
```

## DAW Decimal Adjust WREG Register

**Syntax:** [label] DAW f,s

**Operands:**  $0 \leq f \leq 255$   
 $s \in [0,1]$

**Operation:** If [WREG<3:0> >9] .OR. [DC = 1] then  
WREG<3:0> + 6 → f<3:0>, s<3:0>;  
else  
WREG<3:0> → f<3:0>, s<3:0>;  
If [WREG<7:4> >9] .OR. [C = 1] then  
WREG<7:4> + 6 → f<7:4>, s<7:4>;  
else  
WREG<7:4> → f<7:4>, s<7:4>

**Status Affected:** C

**Encoding:**

0010	111s	ffff	ffff
------	------	------	------

**Description:** DAW adjusts the eight bit value in WREG resulting from the earlier addition of two variables (each in packed BCD format) and produces a correct packed BCD result.

s = 0: Result is placed in Data memory location 'f' and WREG.

s = 1: Result is placed in Data memory location 'f'.

**Words:** 1

**Cycles:** 1

**Q Cycle Activity:**

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Execute	Write register 'f' and other specified register

**Example1:** DAW REG1, 0

**Before Instruction**

```

WREG = 0xA5
REG1 = ??
C    = 0
DC   = 0
```

**After Instruction**

```

WREG = 0x05
REG1 = 0x05
C    = 1
DC   = 0
```

**Example 2:**

**Before Instruction**

```

WREG = 0xCE
REG1 = ??
C    = 0
DC   = 0
```

**After Instruction**

```

WREG = 0x24
REG1 = 0x24
C    = 1
DC   = 0
```

# PIC17C4X

## RRNCF Rotate Right f (no carry)

Syntax: [label] RRNCF f,d  
 Operands:  $0 \leq f \leq 255$   
 $d \in [0,1]$

Operation:  $f \langle n \rangle \rightarrow d \langle n-1 \rangle$ ;  
 $f \langle 0 \rangle \rightarrow d \langle 7 \rangle$

Status Affected: None

Encoding: 

0010	000d	ffff	ffff
------	------	------	------

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



Words: 1  
 Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Execute	Write to destination

**Example 1:** RRNCF REG, 1

Before Instruction

WREG = ?  
 REG = 1101 0111

After Instruction

WREG = 0  
 REG = 1110 1011

**Example 2:** RRNCF REG, 0

Before Instruction

WREG = ?  
 REG = 1101 0111

After Instruction

WREG = 1110 1011  
 REG = 1101 0111

## SETF Set f

Syntax: [label] SETF f,s  
 Operands:  $0 \leq f \leq 255$   
 $s \in [0,1]$

Operation:  $FFh \rightarrow f$ ;  
 $FFh \rightarrow d$

Status Affected: None

Encoding: 

0010	101s	ffff	ffff
------	------	------	------

Description: If 's' is 0, both the data memory location 'f' and WREG are set to FFh. If 's' is 1 only the data memory location 'f' is set to FFh.

Words: 1  
 Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Execute	Write register 'f' and other specified register

**Example1:** SETF REG, 0

Before Instruction

REG = 0xDA  
 WREG = 0x05

After Instruction

REG = 0xFF  
 WREG = 0xFF

**Example2:** SETF REG, 1

Before Instruction

REG = 0xDA  
 WREG = 0x05

After Instruction

REG = 0xFF  
 WREG = 0x05



Applicable Devices	42	R42	42A	43	R43	44
--------------------	----	-----	-----	----	-----	----

Standard Operating Conditions (unless otherwise stated)							
DC CHARACTERISTICS							
Operating temperature -40°C ≤ TA ≤ +85°C for industrial and 0°C ≤ TA ≤ +70°C for commercial							
Operating voltage VDD range as described in Section 17.1							
Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
D080 D081	VOL	<b>Output Low Voltage</b> I/O ports (except RA2 and RA3) with TTL buffer	–	–	0.1VDD 0.4	V V	IOL = 4 mA IOL = 6 mA, VDD = 4.5V Note 6
D082 D083		RA2 and RA3 OSC2/CLKOUT (RC and EC osc modes)	–	–	3.0 0.4	V V	IOL = 60.0 mA, VDD = 5.5V IOL = 2 mA, VDD = 4.5V
D090 D091		<b>Output High Voltage</b> (Note 3) I/O ports (except RA2 and RA3) with TTL buffer	0.9VDD 2.4	– –	– –	V V	IOH = -2 mA IOH = -6.0 mA, VDD = 4.5V Note 6
D092 D093		RA2 and RA3 OSC2/CLKOUT (RC and EC osc modes)	– 2.4	– –	12 –	V V	Pulled-up to externally applied voltage IOH = -5 mA, VDD = 4.5V
D100	Cosc2	<b>Capacitive Loading Specs on Output Pins</b> OSC2 pin	–	–	25 ††	pF	In EC or RC osc modes when OSC2 pin is outputting CLKOUT. External clock is used to drive OSC1.
D101	CIO	All I/O pins and OSC2 (in RC mode)	–	–	50 ††	pF	
D102	CAD	System Interface Bus (PORTC, PORTD and PORTE)	–	–	100 ††	pF	

\* These parameters are characterized but not tested.

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

‡ These parameters are for design guidance only and are not tested, nor characterized.

†† Design guidance to attain the AC timing specifications. These loads are not tested.

Note 1: In RC oscillator configuration, the OSC1 pin is a Schmitt Trigger input. It is not recommended that the PIC17CXX devices be driven with external clock in RC mode.

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

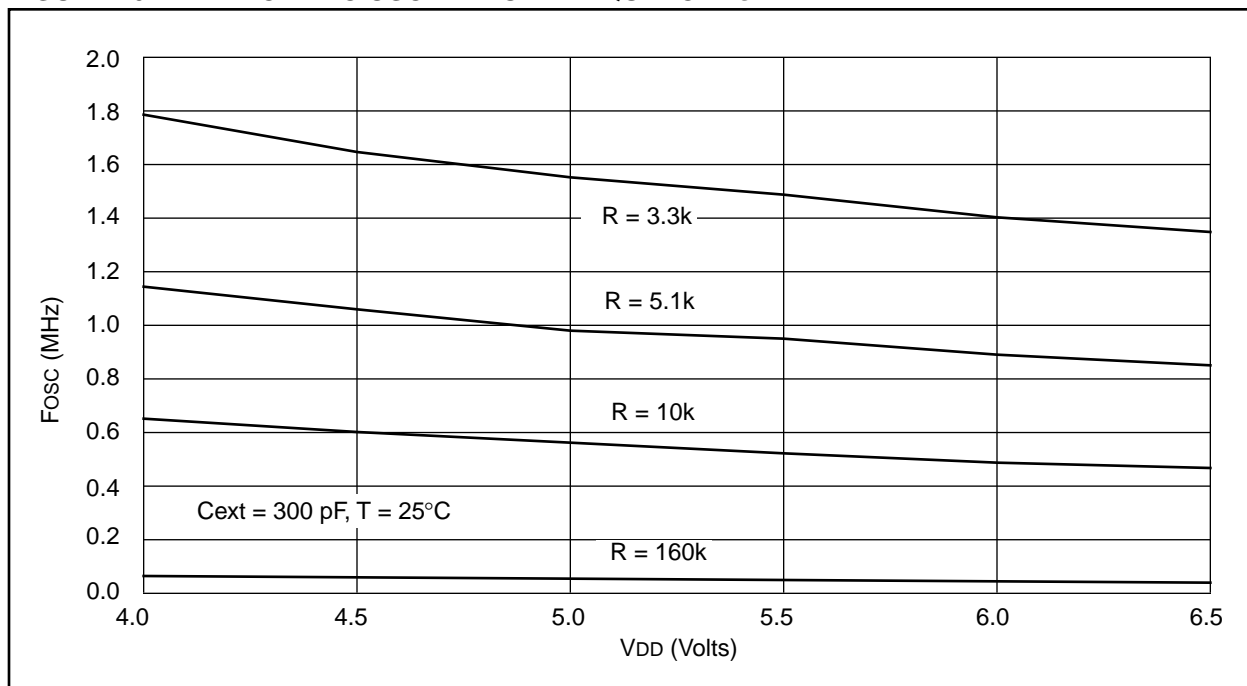
3: Negative current is defined as coming out of the pin.

4: These specifications are for the programming of the on-chip program memory EPROM through the use of the table write instructions. The complete programming specifications can be found in: PIC17CXX Programming Specifications (Literature number DS30139).

5: The MCLR/Vpp pin may be kept in this range at times other than programming, but this is not recommended.

6: For TTL buffers, the better of the two specifications may be used.

**FIGURE 20-4: TYPICAL RC OSCILLATOR FREQUENCY vs. V<sub>DD</sub>**

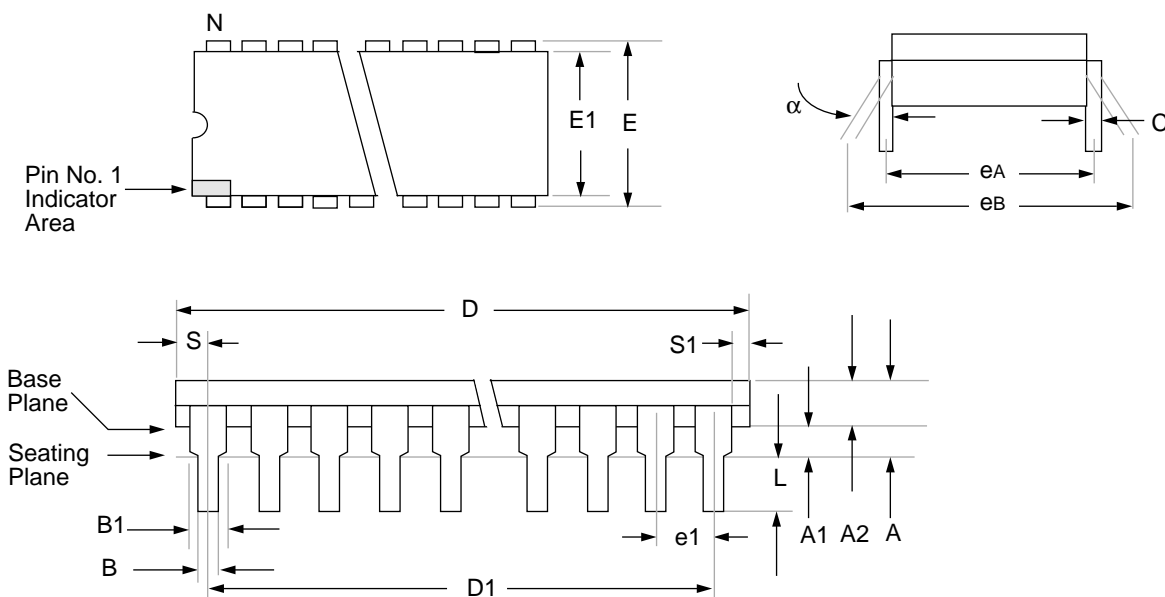


**TABLE 20-2: RC OSCILLATOR FREQUENCIES**

Cext	Rext	Average	
		Fosc @ 5V, 25°C	
22 pF	10k	3.33 MHz	± 12%
	100k	353 kHz	± 13%
100 pF	3.3k	3.54 MHz	± 10%
	5.1k	2.43 MHz	± 14%
	10k	1.30 MHz	± 17%
	100k	129 kHz	± 10%
300 pF	3.3k	1.54 MHz	± 14%
	5.1k	980 kHz	± 12%
	10k	564 kHz	± 16%
	160k	35 kHz	± 18%

# PIC17C4X

## 21.2 40-Lead Plastic Dual In-line (600 mil)



Package Group: Plastic Dual In-Line (PLA)						
Symbol	Millimeters			Inches		
	Min	Max	Notes	Min	Max	Notes
$\alpha$	0°	10°		0°	10°	
A	—	5.080		—	0.200	
A1	0.381	—		0.015	—	
A2	3.175	4.064		0.125	0.160	
B	0.355	0.559		0.014	0.022	
B1	1.270	1.778	Typical	0.050	0.070	Typical
C	0.203	0.381	Typical	0.008	0.015	Typical
D	51.181	52.197		2.015	2.055	
D1	48.260	48.260	Reference	1.900	1.900	Reference
E	15.240	15.875		0.600	0.625	
E1	13.462	13.970		0.530	0.550	
e1	2.489	2.591	Typical	0.098	0.102	Typical
eA	15.240	15.240	Reference	0.600	0.600	Reference
eB	15.240	17.272		0.600	0.680	
L	2.921	3.683		0.115	0.145	
N	40	40		40	40	
S	1.270	—		0.050	—	
S1	0.508	—		0.020	—	

## APPENDIX C: WHAT'S NEW

The structure of the document has been made consistent with other data sheets. This ensures that important topics are covered across all PIC16/17 families. Here is an overview of new features.

Added the following devices:

PIC17CR42

PIC17C42A

PIC17CR43

A 33 MHz option is now available.

## APPENDIX D: WHAT'S CHANGED

To make software more portable across the different PIC16/17 families, the name of several registers and control bits have been changed. This allows control bits that have the same function, to have the same name (regardless of processor family). Care must still be taken, since they may not be at the same special function register address. The following shows the register and bit names that have been changed:

Old Name	New Name
TX8/9	TX9
RC8/9	RX9
RCD8	RX9D
TXD8	TX9D

Instruction DECFSNZ corrected to DCFSNZ

Instruction INCFSNZ corrected to INFSNZ

Enhanced discussion on PWM to include equation for determining bits of PWM resolution.

Section 13.2.2 and 13.3.2 have had the description of updating the FERR and RX9 bits enhanced.

The location of configuration bit PM2 was changed (Figure 6-1 and Figure 14-1).

Enhanced description of the operation of the INTSTA register.

Added note to discussion of interrupt operation.

Tightened electrical spec D110.

Corrected steps for setting up USART Asynchronous Reception.

# PIC17C4X

Delay From External Clock Edge .....	68
Development Support .....	143
Development Tools .....	143
Device Drawings	
44-Lead Plastic Surface Mount (MQFP	
10x10 mm Body 1.6/0.15 mm Lead Form) .....	209
DIGIT BORROW .....	9
Digit Carry (DC) .....	9
Duty Cycle .....	75

## E

### Electrical Characteristics

PIC17C42	
Absolute Maximum Ratings .....	147
Capture Timing .....	159
CLKOUT and I/O Timing .....	156
DC Characteristics .....	149
External Clock Timing .....	155
Memory Interface Read Timing .....	162
Memory Interface Write Timing .....	161
PWM Timing .....	159
RESET, Watchdog Timer, Oscillator Start-up	
Timer and Power-up Timer .....	157
Timer0 Clock Timings .....	158
Timer1, Timer2 and Timer3 Clock Timing .....	158
USART Module, Synchronous Receive .....	160
USART Module, Synchronous Transmission .....	160
PIC17C43/44	
Absolute Maximum Ratings .....	175
Capture Timing .....	188
CLKOUT and I/O Timing .....	185
DC Characteristics .....	177
External Clock Timing .....	184
Memory Interface Read Timing .....	191
Memory Interface Write Timing .....	190
Parameter Measurement Information .....	183
RESET, Watchdog Timer, Oscillator Start-up	
Timer and Power-up Timer Timing .....	186
Timer0 Clock Timing .....	187
Timer1, Timer2 and Timer3 Clock Timing .....	187
Timing Parameter Symbolology .....	182
USART Module Synchronous Receive	
Timing .....	189
USART Module Synchronous Transmission	
Timing .....	189
EPROM Memory Access Time Order Suffix .....	31
Extended Microcontroller .....	29
Extended Microcontroller Mode .....	31
External Memory Interface .....	31
External Program Memory Waveforms .....	31

## F

Family of Devices .....	6
PIC14000 .....	213
PIC16C5X .....	214
PIC16CXXX .....	215
PIC16C6X .....	216
PIC16C7X .....	217
PIC16C8X .....	218
PIC16C9XX .....	219
PIC17CXX .....	220
FERR .....	84, 91
FOSC0 .....	99

FOSC1 .....	99
FS0 .....	36
FS1 .....	36
FS2 .....	36
FS3 .....	36
FSR0 .....	34, 40
FSR1 .....	34, 40
Fuzzy Logic Dev. System ( <i>fuzzyTECH</i> <sup>®</sup> -MP) .....	143, 145

## G

General Format for Instructions .....	108
General Purpose RAM .....	29
General Purpose RAM Bank .....	42
General Purpose Register (GPR) .....	32
GLINTD .....	25, 37, 78, 105
GOTO .....	122
GPR (General Purpose Register) .....	32
Graphs	
IOH vs. VOH, VDD = 3V .....	170, 200
IOH vs. VOH, VDD = 5V .....	171, 201
IOL vs. VOL, VDD = 3V .....	171, 201
IOL vs. VOL, VDD = 5V .....	172, 202
Maximum IDD vs. Frequency	
(External Clock 125°C to -40°C) .....	167, 197
Maximum IPD vs. VDD Watchdog Disabled .....	168, 198
Maximum IPD vs. VDD Watchdog Enabled .....	169, 199
RC Oscillator Frequency vs.	
VDD (Cext = 100 pF) .....	164, 194
RC Oscillator Frequency vs.	
VDD (Cext = 22 pF) .....	164, 194
RC Oscillator Frequency vs.	
VDD (Cext = 300 pF) .....	165, 195
Transconductance of LF Oscillator vs. VDD .....	166, 196
Transconductance of XT Oscillator vs. VDD .....	166, 196
Typical IDD vs. Frequency	
(External Clock 25°C) .....	167, 197
Typical IPD vs. VDD Watchdog Disabled 25°C .....	168, 198
Typical IPD vs. VDD Watchdog Enabled 25°C .....	169, 199
Typical RC Oscillator vs. Temperature .....	163, 193
VTH (Input Threshold Voltage) of I/O Pins vs.	
VDD .....	172, 202
VTH (Input Threshold Voltage) of OSC1 Input	
(In XT, HS, and LP Modes) vs. VDD .....	173, 203
VTH, VIL of $\overline{\text{MCLR}}$ , T0CKI and OSC1	
(In RC Mode) vs. VDD .....	173, 203
WDT Timer Time-Out Period vs. VDD .....	170, 200

## H

Hardware Multiplier .....	49
---------------------------	----

## I

I/O Ports	
Bi-directional .....	64
I/O Ports .....	53
Programming Considerations .....	64
Read-Modify-Write Instructions .....	64
Successive Operations .....	64
INCF .....	123
INCFSNZ .....	124
INCFSZ .....	123
INDF0 .....	34, 40
INDF1 .....	34, 40