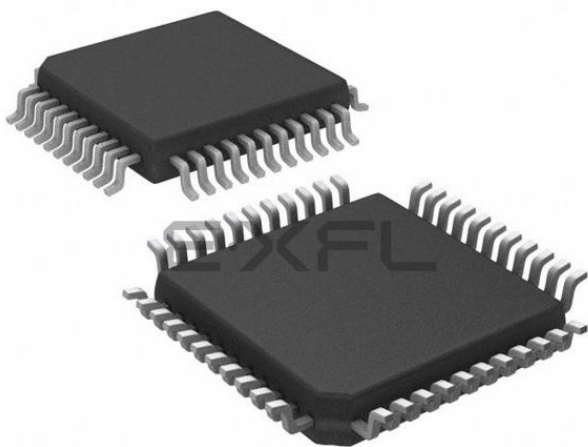




Welcome to [E-XFL.COM](https://www.e-xfl.com)

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	25MHz
Connectivity	UART/USART
Peripherals	POR, PWM, WDT
Number of I/O	33
Program Memory Size	8KB (4K 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	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-MQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic17c43-25-pq

PIC17C4X

NOTES:

4.0 RESET

The PIC17CXX differentiates between various kinds of reset:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$ reset during normal operation
- WDT Reset (normal operation)

Some registers are not affected in any reset condition; their status is unknown on POR and unchanged in any other reset. Most other registers are forced to a "reset state" on Power-on Reset (POR), on $\overline{\text{MCLR}}$ or WDT Reset and on $\overline{\text{MCLR}}$ reset during SLEEP. They are not affected by a WDT Reset during SLEEP, since this reset is viewed as the resumption of normal operation. The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits are set or cleared differently in different reset situations as indicated in Table 4-3. These bits are used in software to determine the nature of reset. See Table 4-4 for a full description of reset states of all registers.

Note: While the device is in a reset state, the internal phase clock is held in the Q1 state. Any processor mode that allows external execution will force the RE0/ALE pin as a low output and the RE1/ $\overline{\text{OE}}$ and RE2/ $\overline{\text{WR}}$ pins as high outputs.

A simplified block diagram of the on-chip reset circuit is shown in Figure 4-1.

4.1 Power-on Reset (POR), Power-up Timer (PWRT), and Oscillator Start-up Timer (OST)

4.1.1 POWER-ON RESET (POR)

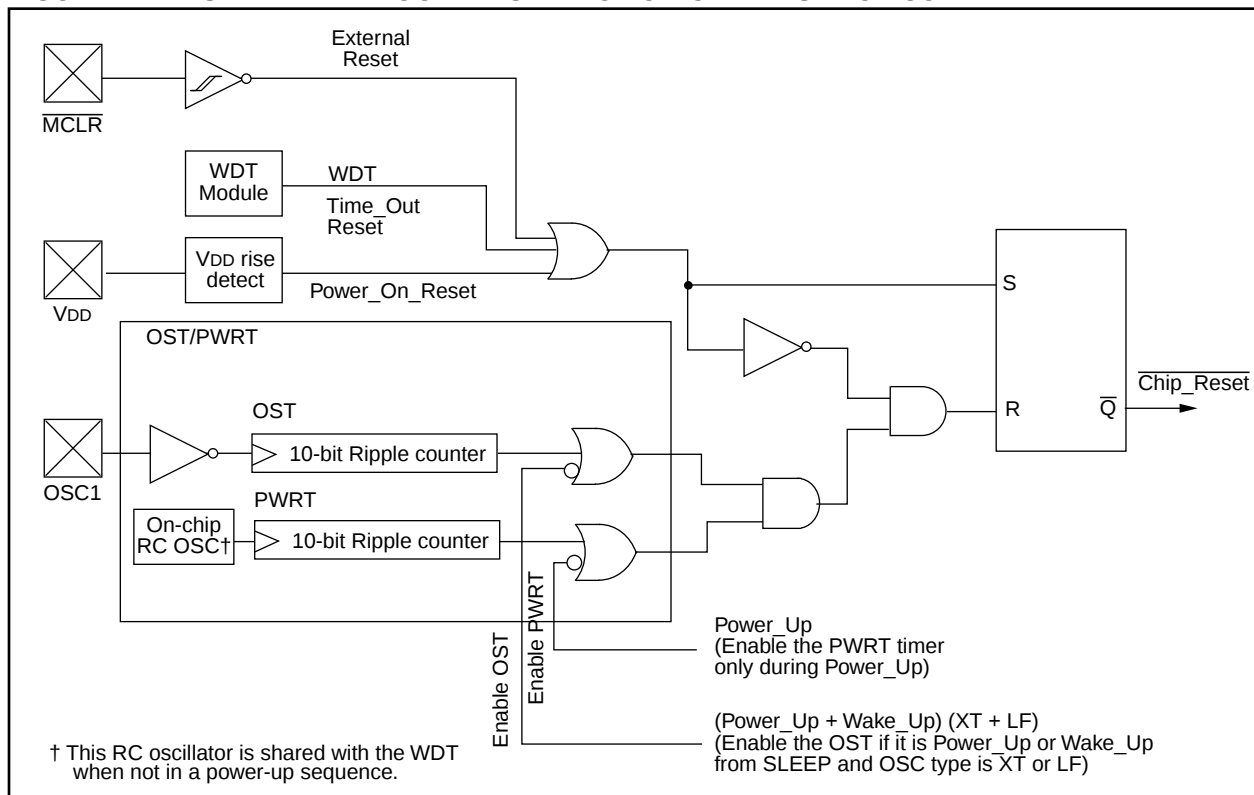
The Power-on Reset circuit holds the device in reset until V_{DD} is above the trip point (in the range of 1.4V - 2.3V). The PIC17C42 does not produce an internal reset when V_{DD} declines. All other devices will produce an internal reset for both rising and falling V_{DD} . To take advantage of the POR, just tie the $\overline{\text{MCLR/VPP}}$ pin directly (or through a resistor) to V_{DD} . This will eliminate external RC components usually needed to create Power-on Reset. A minimum rise time for V_{DD} is required. See Electrical Specifications for details.

4.1.2 POWER-UP TIMER (PWRT)

The Power-up Timer provides a fixed 96 ms time-out (nominal) on power-up. This occurs from rising edge of the POR signal and after the first rising edge of $\overline{\text{MCLR}}$ (detected high). The Power-up Timer operates on an internal RC oscillator. The chip is kept in RESET as long as the PWRT is active. In most cases the PWRT delay allows the V_{DD} to rise to an acceptable level.

The power-up time delay will vary from chip to chip and to V_{DD} and temperature. See DC parameters for details.

FIGURE 4-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



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	$\overline{\text{MCLR}}$ 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 CLRWDI 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) 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 ⁽²⁾
WDT Reset during normal operation		0000h	--11 01--	No
WDT Reset during SLEEP ⁽³⁾		0000h	--11 00--	Yes ⁽²⁾
Interrupt wake-up from SLEEP	GLINTD is set	PC + 1	--11 10--	Yes ⁽²⁾
	GLINTD is clear	PC + 1 ⁽¹⁾	--10 10--	Yes ⁽²⁾

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.

5.3 Peripheral Interrupt Request Register (PIR)

This register contains the individual flag bits for the peripheral interrupts.

Note: These bits will be set by the specified condition, even if the corresponding interrupt enable bit is cleared (interrupt disabled), or the GLINTD bit is set (all interrupts disabled). Before enabling an interrupt, the user may wish to clear the interrupt flag to ensure that the program does not immediately branch to the peripheral interrupt service routine.

FIGURE 5-4: PIR REGISTER (ADDRESS: 16h, BANK 1)

R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R/W - 0	R - 1	R - 0
RBIF	TMR3IF	TMR2IF	TMR1IF	CA2IF	CA1IF	TXIF	RCIF
bit7							bit0
bit 7: RBIF: PORTB Interrupt on Change Flag bit 1 = One of the PORTB inputs changed (Software must end the mismatch condition) 0 = None of the PORTB inputs have changed bit 6: TMR3IF: Timer3 Interrupt Flag bit If Capture1 is enabled ($CA1/\overline{PR3} = 1$) 1 = Timer3 overflowed 0 = Timer3 did not overflow If Capture1 is disabled ($CA1/\overline{PR3} = 0$) 1 = Timer3 value has rolled over to 0000h from equalling the period register (PR3H:PR3L) value 0 = Timer3 value has not rolled over to 0000h from equalling the period register (PR3H:PR3L) value bit 5: TMR2IF: Timer2 Interrupt Flag bit 1 = Timer2 value has rolled over to 0000h from equalling the period register (PR2) value 0 = Timer2 value has not rolled over to 0000h from equalling the period register (PR2) value bit 4: TMR1IF: Timer1 Interrupt Flag bit If Timer1 is in 8-bit mode ($T16 = 0$) 1 = Timer1 value has rolled over to 0000h from equalling the period register (PR) value 0 = Timer1 value has not rolled over to 0000h from equalling the period register (PR2) value If Timer1 is in 16-bit mode ($T16 = 1$) 1 = TMR1:TMR2 value has rolled over to 0000h from equalling the period register (PR1:PR2) value 0 = TMR1:TMR2 value has not rolled over to 0000h from equalling the period register (PR1:PR2) value bit 3: CA2IF: Capture2 Interrupt Flag bit 1 = Capture event occurred on RB1/CAP2 pin 0 = Capture event did not occur on RB1/CAP2 pin bit 2: CA1IF: Capture1 Interrupt Flag bit 1 = Capture event occurred on RB0/CAP1 pin 0 = Capture event did not occur on RB0/CAP1 pin bit 1: TXIF: USART Transmit Interrupt Flag bit 1 = Transmit buffer is empty 0 = Transmit buffer is full bit 0: RCIF: USART Receive Interrupt Flag bit 1 = Receive buffer is full 0 = Receive buffer is empty							

R = Readable bit
 W = Writable bit
 -n = Value at POR reset

PIC17C4X

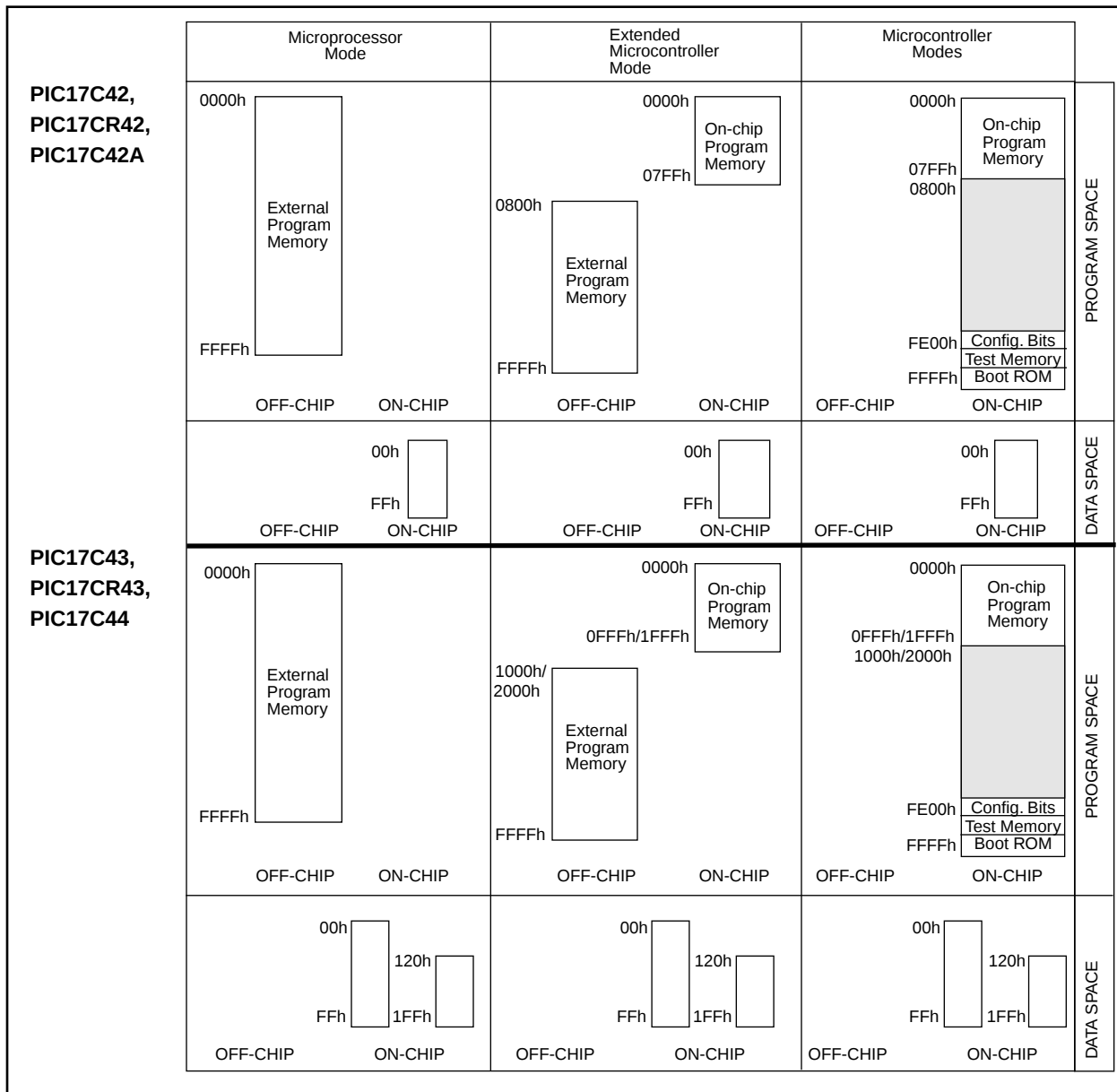
TABLE 6-1: MODE MEMORY ACCESS

Operating Mode	Internal Program Memory	Configuration Bits, Test Memory, Boot ROM
Microprocessor	No Access	No Access
Microcontroller	Access	Access
Extended Microcontroller	Access	No Access
Protected Microcontroller	Access	Access

The PIC17C4X can operate in modes where the program memory is off-chip. They are the microprocessor and extended microcontroller modes. The microprocessor mode is the default for an unprogrammed device.

Regardless of the processor mode, data memory is always on-chip.

FIGURE 6-2: MEMORY MAP IN DIFFERENT MODES



9.4 PORTD and DDRD Registers

PORTD is an 8-bit bi-directional port. The corresponding data direction register is DDRD. A '1' in DDRD configures the corresponding port pin as an input. A '0' in the DDRC register configures the corresponding port pin as an output. Reading PORTD reads the status of the pins, whereas writing to it will write to the port latch. PORTD is multiplexed with the system bus. When operating as the system bus, PORTD is the high order byte of the address/data bus (AD15:AD8). 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.

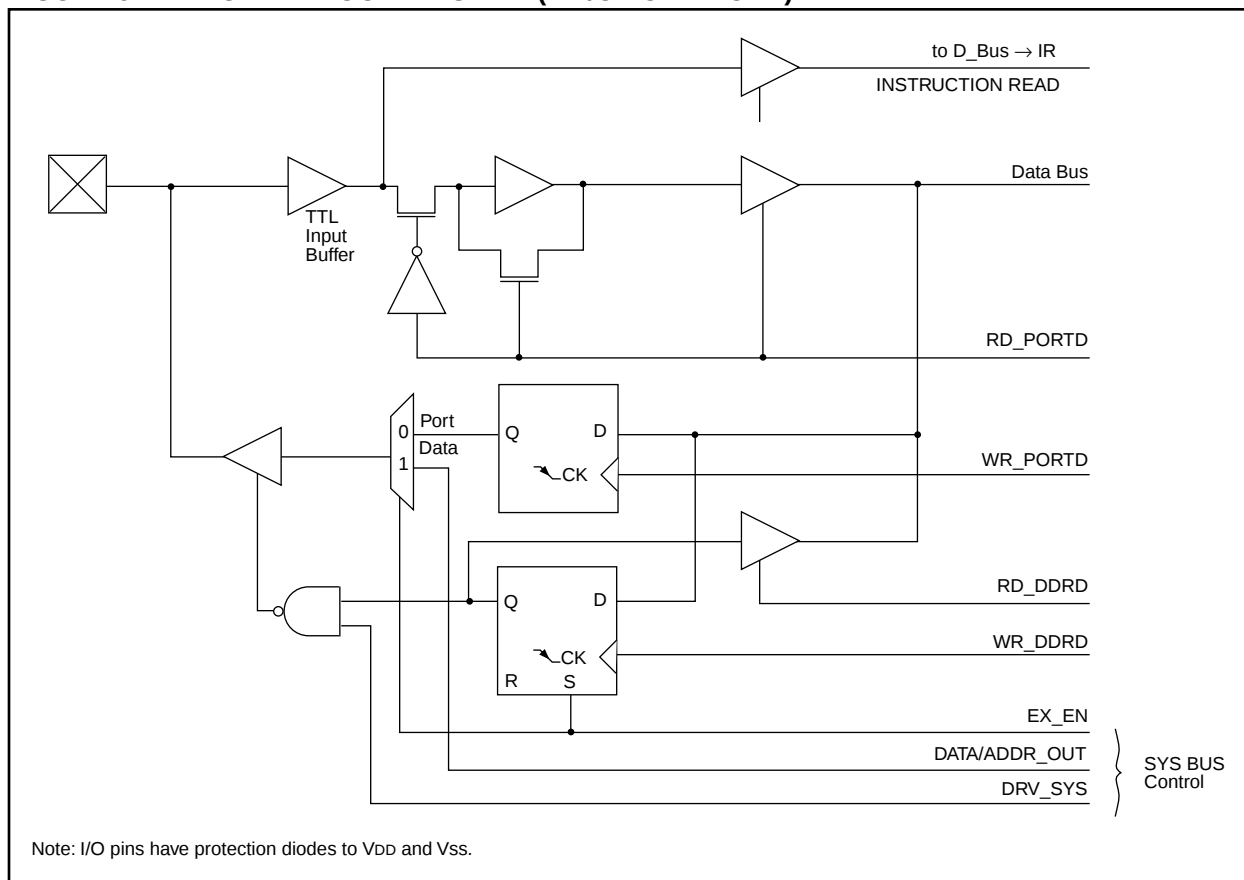
Example 9-3 shows the instruction sequence to initialize PORTD. The Bank Select Register (BSR) must be selected to Bank 1 for the port to be initialized.

EXAMPLE 9-3: INITIALIZING PORTD

```

MOVLB 1      ; Select Bank 1
CLRF  PORTD  ; Initialize PORTD data
              ; latches before setting
              ; the data direction
              ; register
MOVLW 0xCF   ; Value used to initialize
              ; data direction
MOVWF DDRD   ; Set RD<3:0> as inputs
              ; RD<5:4> as outputs
              ; RD<7:6> as inputs
    
```

FIGURE 9-7: PORTD BLOCK DIAGRAM (IN I/O PORT MODE)



13.4 USART Synchronous Slave Mode

The synchronous slave mode differs from the master mode in the fact that the shift clock is supplied externally at the RA5/TX/CK pin (instead of being supplied internally in the master mode). This allows the device to transfer or receive data in the SLEEP mode. The slave mode is entered by clearing the CSRC (TXSTA<7>) bit.

13.4.1 USART SYNCHRONOUS SLAVE TRANSMIT

The operation of the sync master and slave modes are identical except in the case of the SLEEP mode.

If two words are written to TXREG and then the SLEEP instruction executes, the following will occur. The first word will immediately transfer to the TSR and will transmit as the shift clock is supplied. The second word will remain in TXREG. TXIF will not be set. When the first word has been shifted out of TSR, TXREG will transfer the second word to the TSR and the TXIF flag will now be set. If TXIE is enabled, the interrupt will wake the chip from SLEEP and if the global interrupt is enabled, then the program will branch to interrupt vector (0020h).

Steps to follow when setting up a Synchronous Slave Transmission:

1. Enable the synchronous slave serial port by setting the SYNC and SPEN bits and clearing the CSRC bit.
2. Clear the CREN bit.
3. If interrupts are desired, then set the TXIE bit.
4. If 9-bit transmission is desired, then set the TX9 bit.
5. Start transmission by loading data to TXREG.
6. If 9-bit transmission is selected, the ninth bit should be loaded in TX9D.
7. Enable the transmission by setting TXEN.

Writing the transmit data to the TXREG, then enabling the transmit (setting TXEN) allows transmission to start sooner than doing these two events in the reverse order.

Note: To terminate a transmission, either clear the SPEN bit, or the TXEN bit. This will reset the transmit logic, so that it will be in the proper state when transmit is re-enabled.

13.4.2 USART SYNCHRONOUS SLAVE RECEPTION

Operation of the synchronous master and slave modes are identical except in the case of the SLEEP mode. Also, SREN is a don't care in slave mode.

If receive is enabled (CREN) prior to the SLEEP instruction, then a word may be received during SLEEP. On completely receiving the word, the RSR will transfer the data to RCREG (setting RCIF) and if the RCIE bit is set, the interrupt generated will wake the chip from SLEEP. If the global interrupt is enabled, the program will branch to the interrupt vector (0020h).

Steps to follow when setting up a Synchronous Slave Reception:

1. Enable the synchronous master serial port by setting the SYNC and SPEN bits and clearing the CSRC bit.
2. If interrupts are desired, then set the RCIE bit.
3. If 9-bit reception is desired, then set the RX9 bit.
4. To enable reception, set the CREN bit.
5. The RCIF bit will be set when reception is complete and an interrupt will be generated if the RCIE bit was set.
6. Read RCSTA to get the ninth bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading RCREG.
8. If any error occurred, clear the error by clearing the CREN bit.

Note: To abort reception, either clear the SPEN bit, the SREN bit (when in single receive mode), or the CREN bit (when in continuous receive mode). This will reset the receive logic, so that it will be in the proper state when receive is re-enabled.

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 Ω resistor provides the negative feedback for stability. The 10 k Ω 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

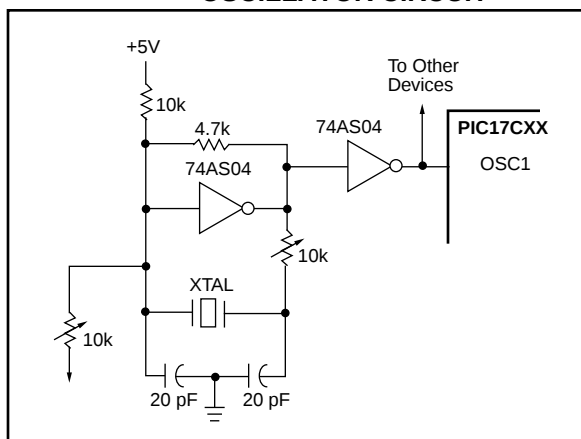
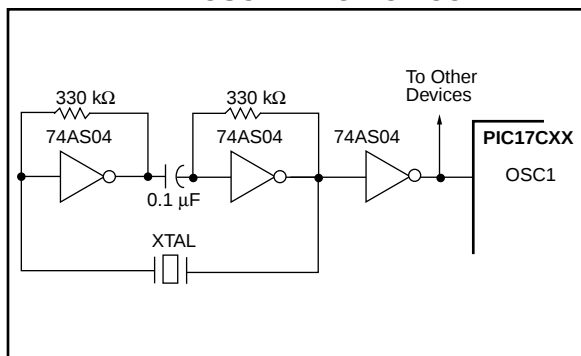


Figure 14-6 shows a series resonant oscillator circuit. This circuit is also designed to use the fundamental frequency of the crystal. The inverter performs a 180-degree phase shift in a series resonant oscillator circuit. The 330 k Ω resistors provide the negative feedback to bias the inverters in their 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 Ω , the oscillator operation may become unstable, or stop completely. For very high R_{ext} values (e.g. 1 M Ω), the oscillator becomes sensitive to noise, humidity and leakage. Thus, we recommend to keep R_{ext} between 3 k Ω and 100 k Ω .

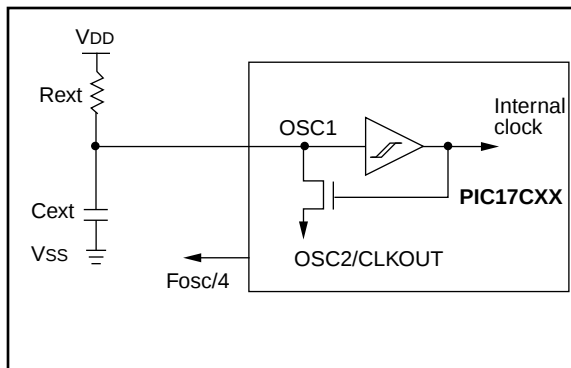
Although the oscillator will operate with no external capacitor (C_{ext} = 0 pF), we recommend using values above 20 pF for noise and stability reasons. With little or no external capacitance, oscillation frequency can vary dramatically due to changes in external capacitances, such as PCB trace capacitance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



15.0 INSTRUCTION SET SUMMARY

The PIC17CXX instruction set consists of 58 instructions. Each instruction is a 16-bit word divided into an OPCODE and one or more operands. The opcode specifies the instruction type, while the operand(s) further specify the operation of the instruction. The PIC17CXX instruction set can be grouped into three types:

- byte-oriented
- bit-oriented
- literal and control operations.

These formats are shown in Figure 15-1.

Table 15-1 shows the field descriptions for the opcodes. These descriptions are useful for understanding the opcodes in Table 15-2 and in each specific instruction descriptions.

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

The destination designator specifies where the result of the operation is to be placed. If 'd' = '0', the result is placed in the WREG register. If 'd' = '1', the result is placed in the file register specified by the instruction.

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

literal and control operations, 'k' represents an 8- or 11-bit constant or literal value.

The instruction set is highly orthogonal and is grouped into:

- byte-oriented operations
- bit-oriented operations
- literal and control operations

All instructions are executed within one single instruction cycle, unless:

- a conditional test is true
- the program counter is changed as a result of an instruction
- a table read or a table write instruction is executed (in this case, the execution takes two instruction cycles with the second cycle executed as a NOP)

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 25 MHz, the normal instruction execution time is 160 ns. If a conditional test is true or the program counter is changed as a result of an instruction, the instruction execution time is 320 ns.

TABLE 15-1: OPCODE FIELD DESCRIPTIONS

Field	Description
f	Register file address (00h to FFh)
p	Peripheral register file address (00h to 1Fh)
i	Table pointer control i = '0' (do not change) i = '1' (increment after instruction execution)
t	Table byte select t = '0' (perform operation on lower byte) t = '1' (perform operation on upper byte literal field, constant data)
WREG	Working register (accumulator)
b	Bit address within an 8-bit file register
k	Literal field, constant data or label
x	Don't care location (= '0' or '1') The assembler will generate code with x = '0'. It is the recommended form of use for compatibility with all Microchip software tools.
d	Destination select 0 = store result in WREG 1 = store result in file register f Default is d = '1'
u	Unused, encoded as '0'
s	Destination select 0 = store result in file register f and in the WREG 1 = store result in file register f Default is s = '1'
label	Label name
C, DC, Z, OV	ALU status bits Carry, Digit Carry, Zero, Overflow
GLINTD	Global Interrupt Disable bit (CPUSTA<4>)
TBLPTR	Table Pointer (16-bit)
TBLAT	Table Latch (16-bit) consists of high byte (TBLATH) and low byte (TBLATL)
TBLATL	Table Latch low byte
TBLATH	Table Latch high byte
TOS	Top of Stack
PC	Program Counter
BSR	Bank Select Register
WDT	Watchdog Timer Counter
T0	Time-out bit
PD	Power-down bit
dest	Destination either the WREG register or the specified register file location
[]	Options
()	Contents
→	Assigned to
< >	Register bit field
∈	In the set of
<i>italics</i>	User defined term (font is courier)

CLRWDT Clear Watchdog Timer

Syntax: [*label*] CLRWDT

Operands: None

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

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

Encoding:

0000	0000	0000	0100
------	------	------	------

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

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ALUSTA	Execute	NOP

Example: CLRWDT

Before Instruction

 WDT counter = ?

After Instruction

 WDT counter = 0x00

 WDT Postscaler = 0

$\overline{TO}$ = 1

$\overline{PD}$ = 1

COMF Complement f

Syntax: [*label*] COMF f,d

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

Operation: $(\overline{f}) \rightarrow (\text{dest})$

Status Affected: Z

Encoding:

0001	001d	ffff	ffff
------	------	------	------

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

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: COMF REG1, 0

Before Instruction

 REG1 = 0x13

After Instruction

 REG1 = 0x13

 WREG = 0xEC

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
 $WREG<3:0> + 6 \rightarrow f<3:0>, s<3:0>;$
else
 $WREG<3:0> \rightarrow f<3:0>, s<3:0>;$
If $[WREG<7:4> > 9]$.OR. $[C = 1]$ then
 $WREG<7:4> + 6 \rightarrow f<7:4>, s<7:4>;$
else
 $WREG<7:4> \rightarrow f<7:4>, s<7:4>;$

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
```

DECF Decrement f

Syntax: [label] DECF f,d

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

Operation: $(f) - 1 \rightarrow (\text{dest})$

Status Affected: OV, C, DC, Z

Encoding:

0000	011d	ffff	ffff
------	------	------	------

Description: Decrement register 'f'. If 'd' is 0 the result is stored in WREG. If 'd' is 1 the result is stored back in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: DECF CNT, 1

Before Instruction

CNT = 0x01
Z = 0

After Instruction

CNT = 0x00
Z = 1

DECFSZ Decrement f, skip if 0

Syntax: [label] DECFSZ f,d

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

Operation: $(f) - 1 \rightarrow (\text{dest})$;
skip if result = 0

Status Affected: None

Encoding:

0001	011d	ffff	ffff
------	------	------	------

Description: The contents of register 'f' are decremented. If 'd' is 0 the result is placed in WREG. 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, and an NOP is executed instead making it a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

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

Example: HERE DECFSZ CNT, 1
GOTO LOOP

CONTINUE

Before Instruction

PC = Address (HERE)

After Instruction

CNT = CNT - 1
If CNT = 0;
PC = Address (CONTINUE)
If CNT $\neq$ 0;
PC = Address (HERE+1)

PIC17C4X

DCFSNZ Decrement f, skip if not 0

Syntax: `[label] DCFSNZ f,d`

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

Operation: $(f) - 1 \rightarrow (\text{dest})$;
skip if not 0

Status Affected: None

Encoding:

0010	011d	ffff	ffff
------	------	------	------

Description: The contents of register 'f' are decremented. If 'd' is 0 the result is placed in WREG. If 'd' is 1 the result is placed back in register 'f'.
If the result is not 0, the next instruction, which is already fetched, is discarded, and an NOP is executed instead making it a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

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

If skip:

Q1	Q2	Q3	Q4
Forced NOP	NOP	Execute	NOP

Example: HERE DCFSNZ TEMP, 1
 ZERO :
 NZERO :

Before Instruction

TEMP_VALUE = ?

After Instruction

TEMP_VALUE = TEMP_VALUE - 1,

If TEMP_VALUE = 0;

PC = Address (ZERO)

If TEMP_VALUE $\neq$ 0;

PC = Address (NZERO)

GOTO Unconditional Branch

Syntax: `[label] GOTO k`

Operands: $0 \leq k \leq 8191$

Operation: $k \rightarrow PC<12:0>$;
 $k<12:8> \rightarrow PCLATH<4:0>$;
 $PC<15:13> \rightarrow PCLATH<7:5>$

Status Affected: None

Encoding:

110k	kkkk	kkkk	kkkk
------	------	------	------

Description: GOTO allows an unconditional branch anywhere within an 8K page boundary. The thirteen bit immediate value is loaded into PC bits <12:0>. Then the upper eight bits of PC are loaded into PCLATH. GOTO is always a two-cycle instruction.

Words: 1

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'<7:0>	Execute	NOP
Forced NOP	NOP	Execute	NOP

Example: GOTO THERE

After Instruction

PC = Address (THERE)

PIC17C4X

NOTES:

16.6 PICDEM-1 Low-Cost PIC16/17 Demonstration Board

The PICDEM-1 is a simple board which demonstrates the capabilities of several of Microchip's microcontrollers. The microcontrollers supported are: PIC16C5X (PIC16C54 to PIC16C58A), PIC16C61, PIC16C62X, PIC16C71, PIC16C8X, PIC17C42, PIC17C43 and PIC17C44. All necessary hardware and software is included to run basic demo programs. The users can program the sample microcontrollers provided with the PICDEM-1 board, on a PRO MATE II or PICSTART-16B programmer, and easily test firmware. The user can also connect the PICDEM-1 board to the PICMASTER emulator and download the firmware to the emulator for testing. Additional prototype area is available for the user to build some additional hardware and connect it to the microcontroller socket(s). Some of the features include an RS-232 interface, a potentiometer for simulated analog input, push-button switches and eight LEDs connected to PORTB.

16.7 PICDEM-2 Low-Cost PIC16CXX Demonstration Board

The PICDEM-2 is a simple demonstration board that supports the PIC16C62, PIC16C64, PIC16C65, PIC16C73 and PIC16C74 microcontrollers. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-2 board, on a PRO MATE II programmer or PICSTART-16C, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-2 board to test firmware. Additional prototype area has been provided to the user for adding additional hardware and connecting it to the microcontroller socket(s). Some of the features include a RS-232 interface, push-button switches, a potentiometer for simulated analog input, a Serial EEPROM to demonstrate usage of the I²C bus and separate headers for connection to an LCD module and a keypad.

16.8 PICDEM-3 Low-Cost PIC16CXXX Demonstration Board

The PICDEM-3 is a simple demonstration board that supports the PIC16C923 and PIC16C924 in the PLCC package. It will also support future 44-pin PLCC microcontrollers with a LCD Module. All the necessary hardware and software is included to run the basic demonstration programs. The user can program the sample microcontrollers provided with the PICDEM-3 board, on a PRO MATE II programmer or PICSTART Plus with an adapter socket, and easily test firmware. The PICMASTER emulator may also be used with the PICDEM-3 board to test firmware. Additional prototype area has been provided to the user for adding hardware and connecting it to the microcontroller socket(s). Some of the features

include an RS-232 interface, push-button switches, a potentiometer for simulated analog input, a thermistor and separate headers for connection to an external LCD module and a keypad. Also provided on the PICDEM-3 board is an LCD panel, with 4 commons and 12 segments, that is capable of displaying time, temperature and day of the week. The PICDEM-3 provides an additional RS-232 interface and Windows 3.1 software for showing the demultiplexed LCD signals on a PC. A simple serial interface allows the user to construct a hardware demultiplexer for the LCD signals. PICDEM-3 will be available in the 3rd quarter of 1996.

16.9 MPLAB Integrated Development Environment Software

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

- A full featured editor
- Three operating modes
 - editor
 - emulator
 - simulator
- A project manager
- Customizable tool bar and key mapping
- A status bar with project information
- Extensive on-line help

MPLAB allows you to:

- Edit your source files (either assembly or 'C')
- One touch assemble (or compile) and download to PIC16/17 tools (automatically updates all project information)
- Debug using:
 - source files
 - absolute listing file
- Transfer data dynamically via DDE (soon to be replaced by OLE)
- Run up to four emulators on the same PC

The ability to use MPLAB with Microchip's simulator allows a consistent platform and the ability to easily switch from the low cost simulator to the full featured emulator with minimal retraining due to development tools.

16.10 Assembler (MPASM)

The MPASM Universal Macro Assembler is a PC-hosted symbolic assembler. It supports all microcontroller series including the PIC12C5XX, PIC14000, PIC16C5X, PIC16CXXX, and PIC17CXX families.

MPASM offers full featured Macro capabilities, conditional assembly, and several source and listing formats. It generates various object code formats to support Microchip's development tools as well as third party programmers.

17.4 Timing Diagrams and Specifications

FIGURE 17-2: EXTERNAL CLOCK TIMING

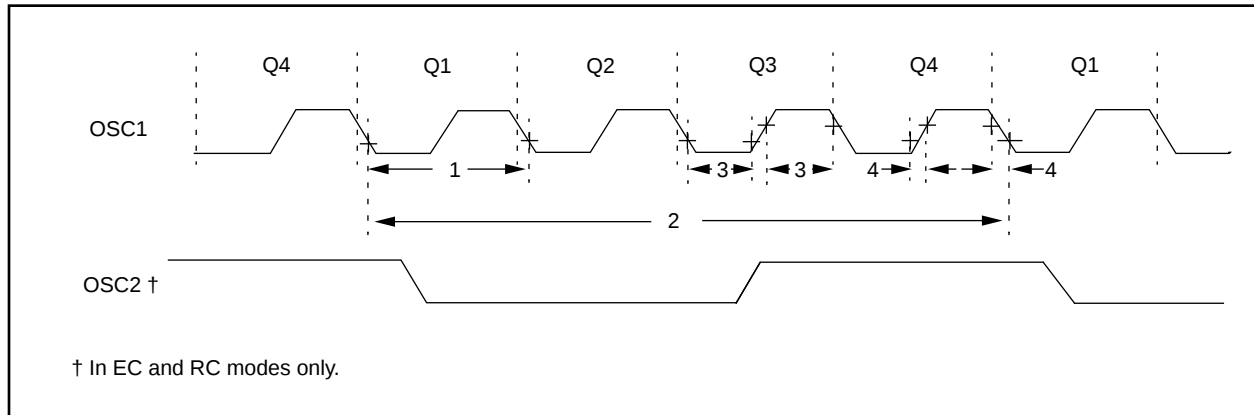


TABLE 17-2: EXTERNAL CLOCK TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
	Fosc	External CLKIN Frequency (Note 1)	DC	—	16	MHz	EC osc mode - PIC17C42-16
			DC	—	25	MHz	- PIC17C42-25
		Oscillator Frequency (Note 1)	DC	—	4	MHz	RC osc mode
			1	—	16	MHz	XT osc mode - PIC17C42-16
1	Tosc	External CLKIN Period (Note 1)	1	—	25	MHz	- PIC17C42-25
			DC	—	2	MHz	LF osc mode
		Oscillator Period (Note 1)	250	—	—	ns	RC osc mode
			62.5	—	1,000	ns	XT osc mode - PIC17C42-16
2	Tcy	Instruction Cycle Time (Note 1)	40	—	—	ns	- PIC17C42-25
			—	—	—	ns	LF osc mode
			—	—	—	ns	
			—	—	—	ns	
3	TosL, TosH	Clock in (OSC1) High or Low Time	10 ‡	—	—	ns	EC oscillator
4	TosR, TosF	Clock in (OSC1) Rise or Fall Time	—	—	5 ‡	ns	EC oscillator

† 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.

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

19.0 PIC17CR42/42A/43/R43/44 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings †

Ambient temperature under bias	-55 to +125°C
Storage temperature	-65°C to +150°C
Voltage on VDD with respect to VSS	0 to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to VSS (Note 2)	-0.6V to +14V
Voltage on RA2 and RA3 with respect to VSS.....	-0.6V to +14V
Voltage on all other pins with respect to VSS	-0.6V to VDD + 0.6V
Total power dissipation (Note 1).....	1.0W
Maximum current out of VSS pin(s) - total	250 mA
Maximum current into VDD pin(s) - total	200 mA
Input clamp current, I _{IK} (V _I < 0 or V _I > VDD)	±20 mA
Output clamp current, I _{OK} (V _O < 0 or V _O > VDD).....	±20 mA
Maximum output current sunk by any I/O pin (except RA2 and RA3).....	35 mA
Maximum output current sunk by RA2 or RA3 pins	60 mA
Maximum output current sourced by any I/O pin	20 mA
Maximum current sunk by PORTA and PORTB (combined).....	150 mA
Maximum current sourced by PORTA and PORTB (combined).....	100 mA
Maximum current sunk by PORTC, PORTD and PORTE (combined).....	150 mA
Maximum current sourced by PORTC, PORTD and PORTE (combined).....	100 mA

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

Note 2: Voltage spikes below VSS at the $\overline{\text{MCLR}}$ pin, inducing currents greater than 80 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the $\overline{\text{MCLR}}$ pin rather than pulling this pin directly to VSS.

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

PIC17C4X

Applicable Devices 42 R42 42A 43 R43 44

FIGURE 20-13: WDT TIMER TIME-OUT PERIOD vs. VDD

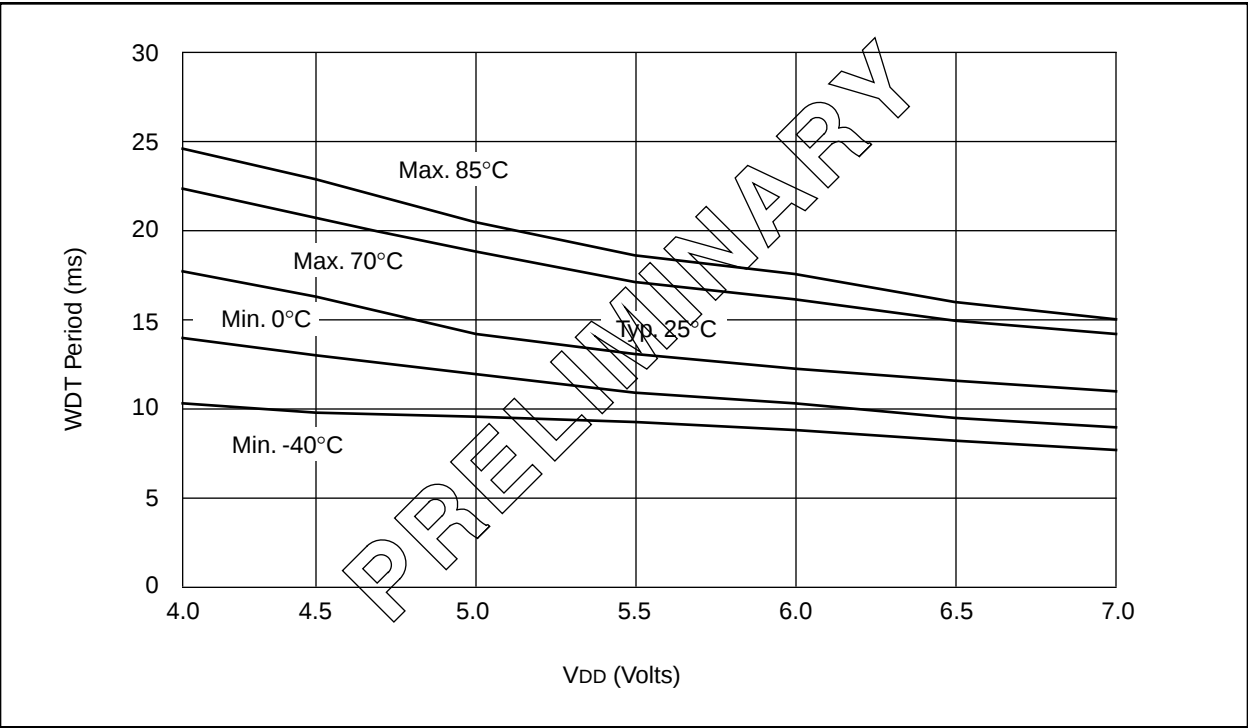
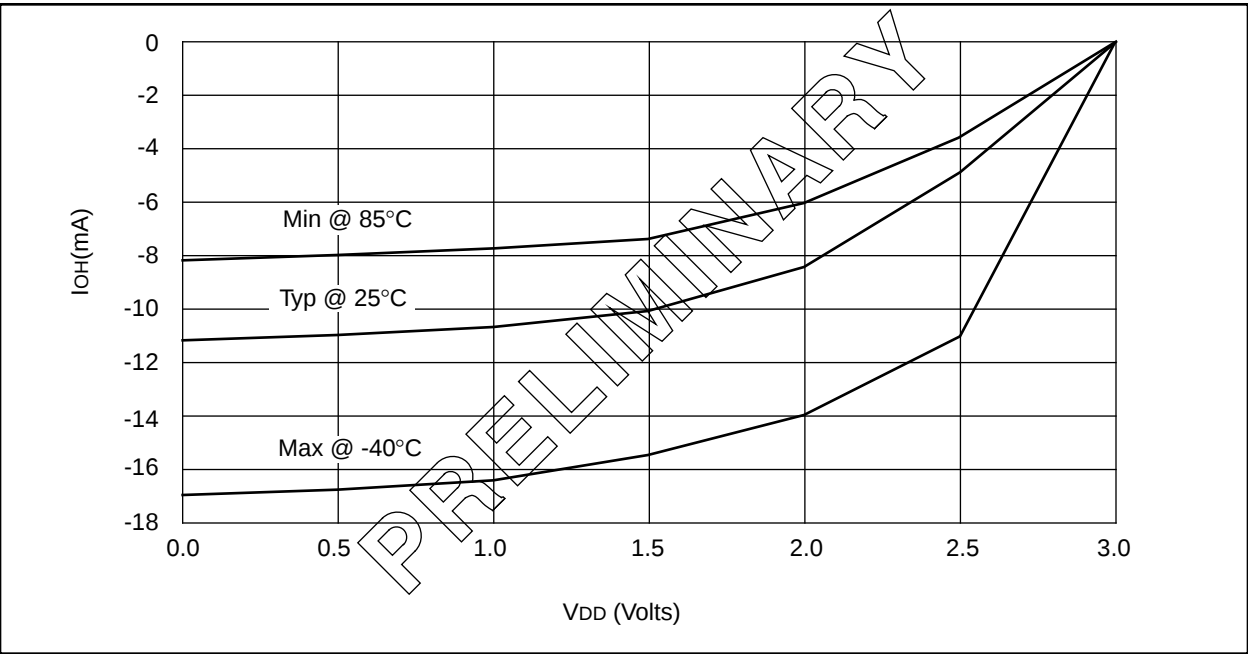


FIGURE 20-14: IOH vs. VOH, VDD = 3V



PIC17C4X

E.6 PIC16C8X Family of Devices

	Clock			Memory			Peripherals		Features		
	Maximum Frequency of Operation (MHz)			Program Memory							
	Flash	EEPROM	ROM	Data Memory (bytes)	Data EEPROM (bytes)	Timer Modules(s)	Interrupt Sources	I/O Pins		Voltage Range (Volts)	Packages
PIC16C84	10	—	1K	—	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC	
PIC16F84 ⁽¹⁾	10	1K	—	—	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC	
PIC16CR84 ⁽¹⁾	10	—	—	1K	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC	
PIC16F83 ⁽¹⁾	10	512	—	—	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC	
PIC16CR83 ⁽¹⁾	10	—	—	512	36	64	TMR0	4	13	2.0-6.0	18-pin DIP, SOIC

All PIC16/17 family devices have Power-on Reset, selectable Watchdog Timer, selectable code protect, and high I/O current capability.

All PIC16C8X family devices use serial programming with clock pin RB6 and data pin RB7.

Note 1: Please contact your local sales office for availability of these devices.

Figure 6-12:	Program Counter using The CALL and GOTO Instructions.....	41	Figure 14-3:	Crystal Operation, Overtone Crystals (XT OSC Configuration)	101
Figure 6-13:	BSR Operation (PIC17C43/R43/44)	42	Figure 14-4:	External Clock Input Operation (EC OSC Configuration)	101
Figure 7-1:	TLWT Instruction Operation.....	43	Figure 14-5:	External Parallel Resonant Crystal Oscillator Circuit	102
Figure 7-2:	TABLWT Instruction Operation.....	43	Figure 14-6:	External Series Resonant Crystal Oscillator Circuit	102
Figure 7-3:	TLRD Instruction Operation	44	Figure 14-7:	RC Oscillator Mode	102
Figure 7-4:	TABLRD Instruction Operation	44	Figure 14-8:	Watchdog Timer Block Diagram.....	104
Figure 7-5:	TABLWT Write Timing (External Memory)	46	Figure 14-9:	Wake-up From Sleep Through Interrupt... ..	105
Figure 7-6:	Consecutive TABLWT Write Timing (External Memory)	47	Figure 15-1:	General Format for Instructions	108
Figure 7-7:	TABLRD Timing	48	Figure 15-2:	Q Cycle Activity	109
Figure 7-8:	TABLRD Timing (Consecutive TABLRD Instructions)	48	Figure 17-1:	Parameter Measurement Information.....	154
Figure 9-1:	RA0 and RA1 Block Diagram	53	Figure 17-2:	External Clock Timing	155
Figure 9-2:	RA2 and RA3 Block Diagram	54	Figure 17-3:	CLKOUT and I/O Timing	156
Figure 9-3:	RA4 and RA5 Block Diagram	54	Figure 17-4:	Reset, Watchdog Timer, Oscillator Start-Up Timer and Power-Up Timer Timing	157
Figure 9-4:	Block Diagram of RB<7:4> and RB<1:0> Port Pins	55	Figure 17-5:	Timer0 Clock Timings.....	158
Figure 9-5:	Block Diagram of RB3 and RB2 Port Pins..	56	Figure 17-6:	Timer1, Timer2, And Timer3 Clock Timings.....	158
Figure 9-6:	Block Diagram of RC<7:0> Port Pins	58	Figure 17-7:	Capture Timings	159
Figure 9-7:	PORTD Block Diagram (in I/O Port Mode)	60	Figure 17-8:	PWM Timings	159
Figure 9-8:	PORTE Block Diagram (in I/O Port Mode)	62	Figure 17-9:	USART Module: Synchronous Transmission (Master/Slave) Timing	160
Figure 9-9:	Successive I/O Operation	64	Figure 17-10:	USART Module: Synchronous Receive (Master/Slave) Timing	160
Figure 11-1:	T0STA Register (Address: 05h, Unbanked)	67	Figure 17-11:	Memory Interface Write Timing	161
Figure 11-2:	Timer0 Module Block Diagram	68	Figure 17-12:	Memory Interface Read Timing	162
Figure 11-3:	TMR0 Timing with External Clock (Increment on Falling Edge)	68	Figure 18-1:	Typical RC Oscillator Frequency vs. Temperature	163
Figure 11-4:	TMR0 Timing: Write High or Low Byte	69	Figure 18-2:	Typical RC Oscillator Frequency vs. VDD	164
Figure 11-5:	TMR0 Read/Write in Timer Mode	70	Figure 18-3:	Typical RC Oscillator Frequency vs. VDD	164
Figure 12-1:	TCON1 Register (Address: 16h, Bank 3) ...	71	Figure 18-4:	Typical RC Oscillator Frequency vs. VDD	165
Figure 12-2:	TCON2 Register (Address: 17h, Bank 3) ...	72	Figure 18-5:	Transconductance (gm) of LF Oscillator vs. VDD	166
Figure 12-3:	Timer1 and Timer2 in Two 8-bit Timer/Counter Mode	73	Figure 18-6:	Transconductance (gm) of XT Oscillator vs. VDD	166
Figure 12-4:	TMR1 and TMR2 in 16-bit Timer/Counter Mode	74	Figure 18-7:	Typical IDD vs. Frequency (External Clock 25°C)	167
Figure 12-5:	Simplified PWM Block Diagram	75	Figure 18-8:	Maximum IDD vs. Frequency (External Clock 125°C to -40°C).....	167
Figure 12-6:	PWM Output	75	Figure 18-9:	Typical IPD vs. VDD Watchdog Disabled 25°C	168
Figure 12-7:	Timer3 with One Capture and One Period Register Block Diagram.....	78	Figure 18-10:	Maximum IPD vs. VDD Watchdog Disabled	168
Figure 12-8:	Timer3 with Two Capture Registers Block Diagram	79	Figure 18-11:	Typical IPD vs. VDD Watchdog Enabled 25°C	169
Figure 12-9:	TMR1, TMR2, and TMR3 Operation in External Clock Mode.....	80	Figure 18-12:	Maximum IPD vs. VDD Watchdog Enabled	169
Figure 12-10:	TMR1, TMR2, and TMR3 Operation in Timer Mode.....	81	Figure 18-13:	WDT Timer Time-Out Period vs. VDD	170
Figure 13-1:	TXSTA Register (Address: 15h, Bank 0)	83	Figure 18-14:	IOH vs. VOH, VDD = 3V.....	170
Figure 13-2:	RCSTA Register (Address: 13h, Bank 0) ...	84	Figure 18-15:	IOH vs. VOH, VDD = 5V.....	171
Figure 13-3:	USART Transmit.....	85	Figure 18-16:	IOL vs. VOL, VDD = 3V.....	171
Figure 13-4:	USART Receive.....	85	Figure 18-17:	IOL vs. VOL, VDD = 5V.....	172
Figure 13-5:	Asynchronous Master Transmission.....	90	Figure 18-18:	VTH (Input Threshold Voltage) of I/O Pins (TTL) vs. VDD	172
Figure 13-6:	Asynchronous Master Transmission (Back to Back)	90	Figure 18-19:	VTH, VIL of I/O Pins (Schmitt Trigger) vs. VDD	173
Figure 13-7:	RX Pin Sampling Scheme	91	Figure 18-20:	VTH (Input Threshold Voltage) of OSC1 Input (In XT and LF Modes) vs. VDD	173
Figure 13-8:	Asynchronous Reception.....	92	Figure 19-1:	Parameter Measurement Information.....	183
Figure 13-9:	Synchronous Transmission	94			
Figure 13-10:	Synchronous Transmission (Through TXEN)	94			
Figure 13-11:	Synchronous Reception (Master Mode, SREN).....	95			
Figure 14-1:	Configuration Word	99			
Figure 14-2:	Crystal or Ceramic Resonator Operation (XT or LF OSC Configuration)	100			