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	1.75KB (1K x 14)
Program Memory Type	FLASH
EEPROM Size	64 x 8
RAM Size	68 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 6V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16f84t-04-so

2.0   PIC16F8X 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 this section. When placing orders, please use the "PIC16F8X Product Identification System" at the back of this data sheet to specify the correct part number.

There are four device "types" as indicated in the device number.

1. **F**, as in PIC16**F**84. These devices have Flash program memory and operate over the standard voltage range.
2. **LF**, as in PIC16**LF**84. These devices have Flash program memory and operate over an extended voltage range.
3. **CR**, as in PIC16**CR**83. These devices have ROM program memory and operate over the standard voltage range.
4. **LCR**, as in PIC16**LCR**84. These devices have ROM program memory and operate over an extended voltage range.

When discussing memory maps and other architectural features, the use of **F** and **CR** also implies the **LF** and **LCR** versions.

2.1   Flash Devices

These devices are offered in the lower cost plastic package, even though the device can be erased and reprogrammed. This allows the same device to be used for prototype development and pilot programs as well as production.

A further advantage of the electrically-erasable Flash version is that it can be erased and reprogrammed in-circuit, or by device programmers, such as Microchip's PICSTART® Plus or PRO MATE® II programmers.

2.2   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 have all Flash locations and configuration options already programmed by the factory. Certain code and prototype verification procedures do apply before production shipments are available.

For information on submitting a QTP code, please contact your Microchip Regional Sales Office.

2.3   Serialized Quick-Turnaround-Production (SQTP^SM) Devices

Microchip offers the 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.

For information on submitting a SQTP code, please contact your Microchip Regional Sales Office.

2.4   ROM Devices

Some of Microchip's devices have a corresponding device where the program memory is a ROM. These devices give a cost savings over Microchip's traditional user programmed devices (EPROM, EEPROM).

ROM devices (PIC16CR8X) do not allow serialization information in the program memory space. The user may program this information into the Data EEPROM.

For information on submitting a ROM code, please contact your Microchip Regional Sales Office.

3.0   ARCHITECTURAL OVERVIEW

The high performance of the PIC16CXX family can be attributed to a number of architectural features commonly found in RISC microprocessors. To begin with, the PIC16CXX uses a Harvard architecture. This architecture has the program and data accessed from separate memories. So the device has a program memory bus and a data memory bus. This improves bandwidth over traditional von Neumann architecture where program and data are fetched from the same memory (accesses over the same bus). Separating program and data memory further allows instructions to be sized differently than the 8-bit wide data word. PIC16CXX opcodes are 14-bits wide, enabling single word instructions. The full 14-bit wide program memory bus fetches a 14-bit instruction in a single cycle. A two-stage pipeline overlaps fetch and execution of instructions (Example 3-1). Consequently, all instructions execute in a single cycle except for program branches.

The PIC16F83 and PIC16CR83 address 512 x 14 of program memory, and the PIC16F84 and PIC16CR84 address 1K x 14 program memory. All program memory is internal.

The PIC16CXX can directly or indirectly address its register files or data memory. All special function registers including the program counter are mapped in the data memory. An orthogonal (symmetrical) instruction set 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 PIC16CXX simple yet efficient. In addition, the learning curve is reduced significantly.

PIC16F8X

TABLE 5-1   PORTA FUNCTIONS

Name	Bit0	Buffer Type	Function
RA0	bit0	TTL	Input/output
RA1	bit1	TTL	Input/output
RA2	bit2	TTL	Input/output
RA3	bit3	TTL	Input/output
RA4/T0CKI	bit4	ST	Input/output or external clock input for TMR0. Output is open drain type.

Legend: TTL = TTL input, ST = Schmitt Trigger input

TABLE 5-2   SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

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
05h	PORTA	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	---u uuuu
85h	TRISA	—	—	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111

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

5.3   I/O Programming Considerations

5.3.1   BI-DIRECTIONAL I/O PORTS

Any instruction which writes, operates internally as a read followed by a write operation. The BCF and BSF instructions, for example, read the register into the CPU, execute the bit operation and write the result back to the register. Caution must be used when these instructions are applied to a port with both inputs and outputs defined. For example, a BSF operation on bit5 of PORTB will cause all eight bits of PORTB to be read into the CPU. Then the BSF operation takes place on bit5 and PORTB is written to the output latches. If another bit of PORTB is used as a bi-directional I/O pin (i.e., bit0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit0 is switched into output mode later on, the content of the data latch is unknown.

Reading the port register, reads the values of the port pins. Writing to the port register writes the value to the port latch. When using read-modify-write instructions (i.e., BCF, BSF, etc.) on a port, the value of the port pins is read, the desired operation is done to this value, and this value is then written to the port latch.

A pin actively outputting a Low or High should not be driven from external devices at the same time in order to change the level on this pin ("wired-or", "wired-and"). The resulting high output current may damage the chip.

5.3.2   SUCCESSIVE OPERATIONS ON I/O PORTS

The actual write to an I/O port happens at the end of an instruction cycle, whereas for reading, the data must be valid at the beginning of the instruction cycle (Figure 5-5). 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 that the pin voltage stabilizes (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 a NOP or another instruction not accessing this I/O port.

Example 5-1 shows the effect of two sequential read-modify-write instructions (e.g., BCF, BSF, etc.) on an I/O port.

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-ups and are
;not connected to other circuitry
;
;                  PORT latch   PORT pins
;                  ----------   ---------
  BCF PORTB, 7    ; 01pp ppp     11pp ppp
  BCF PORTB, 6    ; 10pp ppp     11pp ppp
  BSF STATUS, RP0 ;
  BCF TRISB, 7    ; 10pp ppp     11pp ppp
  BCF TRISB, 6    ; 10pp ppp     10pp ppp
;
;Note that the user may have expected the
;pin values to be 00pp ppp. The 2nd BCF
;caused RB7 to be latched as the pin value
;(high).
```

FIGURE 5-5:   SUCCESSIVE I/O OPERATION
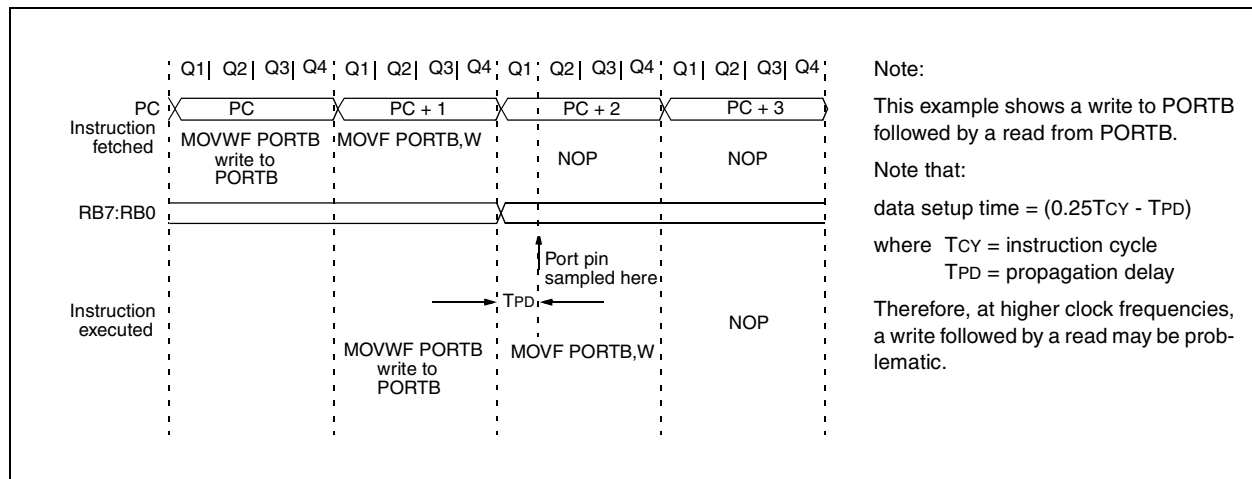
PIC16F8X

NOTES:

PIC16F8X

7.2 EECON1 and EECON2 Registers

EECON1 is the control register with five low order bits physically implemented. The upper-three bits are non-existent and read as '0's.

Control bits RD and WR initiate read and write, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a $\overline{MCLR}$ reset or a WDT time-out reset during normal operation. In these situations, following reset, the user can check the WRERR bit and rewrite the location. The data and address will be unchanged in the EEDATA and EEADR registers.

Interrupt flag bit EEIF is set when write is complete. It must be cleared in software.

EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the Data EEPROM write sequence.

7.3 Reading the EEPROM Data Memory

To read a data memory location, the user must write the address to the EEADR register and then set control bit RD (EECON1<0>). The data is available, in the very next cycle, in the EEDATA register; therefore it can be read in the next instruction. EEDATA will hold this value until another read or until it is written to by the user (during a write operation).

EXAMPLE 7-1: DATA EEPROM READ

```
BCF     STATUS, RP0  ; Bank 0
MOVLW   CONFIG_ADDR  ;
MOVWF   EEADR        ; Address to read
BSF     STATUS, RP0  ; Bank 1
BSF     EECON1, RD   ; EE Read
BCF     STATUS, RP0  ; Bank 0
MOVF    EEDATA, W    ; W = EEDATA
```

7.4 Writing to the EEPROM Data Memory

To write an EEPROM data location, the user must first write the address to the EEADR register and the data to the EEDATA register. Then the user must follow a specific sequence to initiate the write for each byte.

EXAMPLE 7-1: DATA EEPROM WRITE

```
            BSF     STATUS, RP0  ; Bank 1
            BCF     INTCON, GIE  ; Disable INTs.
            BSF     EECON1, WREN ; Enable Write
            MOVLW   55h          ;
Required    MOVWF   EECON2       ; Write 55h
Sequence    MOVLW   AAh          ;
            MOVWF   EECON2       ; Write AAh
            BSF     EECON1,WR    ; Set WR bit
                                 ;   begin write
            BSF     INTCON, GIE  ; Enable INTs.
```

The write will not initiate if the above sequence is not exactly followed (write 55h to EECON2, write AAh to EECON2, then set WR bit) for each byte. We strongly recommend that interrupts be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to errant (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. EEIF must be cleared by software.

7.5 Write Verify

Depending on the application, good programming practice may dictate that the value written to the Data EEPROM should be verified (Example 7-1) to the desired value to be written. This should be used in applications where an EEPROM bit will be stressed near the specification limit. The Total Endurance disk will help determine your comfort level.

Generally the EEPROM write failure will be a bit which was written as a '1', but reads back as a '0' (due to leakage off the bit).

EXAMPLE 7-1: WRITE VERIFY

```
    BCF    STATUS, RP0 ; Bank 0
    :                  ; Any code can go here
    :                  ;
    MOVF   EEDATA, W   ; Must be in Bank 0
    BSF    STATUS, RP0 ; Bank 1
READ
    BSF    EECON1, RD  ; YES, Read the
                       ;   value written
    BCF    STATUS, RP0 ; Bank 0
;
; Is the value written (in W reg) and
;   read (in EEDATA) the same?
;
    SUBWF  EEDATA, W   ;
    BTFSS  STATUS, Z   ; Is difference 0?
    GOTO   WRITE_ERR   ; NO, Write error
    :                  ; YES, Good write
    :                  ; Continue program
```

7.6 Protection Against Spurious Writes

There are conditions when the device may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built in. On power-up, WREN is cleared. Also, the Power-up Timer (72 ms duration) prevents EEPROM write.

The write initiate sequence and the WREN bit together help prevent an accidental write during brown-out, power glitch, or software malfunction.

7.7 Data EEPROM Operation during Code Protect

When the device is code protected, the CPU is able to read and write unscrambled data to the Data EEPROM.

For ROM devices, there are two code protection bits (Section 8.1). One for the ROM program memory and one for the Data EEPROM memory.

TABLE 7-1 REGISTERS/BITS ASSOCIATED WITH DATA EEPROM

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
08h	EEDATA	EEPROM data register								xxxx xxxx	uuuu uuuu
09h	EEADR	EEPROM address register								xxxx xxxx	uuuu uuuu
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	---0 q000
89h	EECON2	EEPROM control register 2								---- ----	---- ----

Legend: x = unknown, u = unchanged, - = unimplemented read as '0', q = value depends upon condition. Shaded cells are not used by Data EEPROM.

TABLE 8-3   RESET CONDITION FOR PROGRAM COUNTER AND THE STATUS REGISTER

Condition	Program Counter	STATUS Register
Power-on Reset	000h	`0001 1xxx`
$\overline{\text{MCLR}}$ Reset during normal operation	000h	`000u uuuu`
$\overline{\text{MCLR}}$ Reset during SLEEP	000h	`0001 0uuu`
WDT Reset (during normal operation)	000h	`0000 1uuu`
WDT Wake-up	PC + 1	`uuu0 0uuu`
Interrupt wake-up from SLEEP	PC + 1 [1]	`uuu1 0uuu`

Legend: `u` = unchanged, `x` = unknown.
Note 1: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).

TABLE 8-4   RESET CONDITIONS FOR ALL REGISTERS

Register	Address	Power-on Reset	$\overline{\text{MCLR}}$ Reset during: – normal operation – SLEEP WDT Reset during normal operation	Wake-up from SLEEP: – through interrupt – through WDT Time-out
W	—	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
INDF	00h	`---- ----`	`---- ----`	`---- ----`
TMR0	01h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
PCL	02h	`0000h`	`0000h`	PC + 1 [2]
STATUS	03h	`0001 1xxx`	`000q quuu` [3]	`uuuq quuu` [3]
FSR	04h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
PORTA	05h	`---x xxxx`	`---u uuuu`	`---u uuuu`
PORTB	06h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
EEDATA	08h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
EEADR	09h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
PCLATH	0Ah	`---0 0000`	`---0 0000`	`---u uuuu`
INTCON	0Bh	`0000 000x`	`0000 000u`	`uuuu uuuu` [1]
INDF	80h	`---- ----`	`---- ----`	`---- ----`
OPTION_REG	81h	`1111 1111`	`1111 1111`	`uuuu uuuu`
PCL	82h	`0000h`	`0000h`	PC + 1
STATUS	83h	`0001 1xxx`	`000q quuu` [3]	`uuuq quuu` [3]
FSR	84h	`xxxx xxxx`	`uuuu uuuu`	`uuuu uuuu`
TRISA	85h	`---1 1111`	`---1 1111`	`---u uuuu`
TRISB	86h	`1111 1111`	`1111 1111`	`uuuu uuuu`
EECON1	88h	`---0 x000`	`---0 q000`	`---0 uuuu`
EECON2	89h	`---- ----`	`---- ----`	`---- ----`
PCLATH	8Ah	`---0 0000`	`---0 0000`	`---u uuuu`
INTCON	8Bh	`0000 000x`	`0000 000u`	`uuuu uuuu` [1]

Legend: `u` = unchanged,   `x` = unknown,   `-` = unimplemented bit read as '0',
      `q` = value depends on condition.
Note 1: One or more bits in INTCON will be affected (to cause wake-up).
    2: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).
    3: Table 8-3 lists the reset value for each specific condition.

8.12.3 WAKE-UP USING INTERRUPTS

When global interrupts are disabled (GIE cleared) and any interrupt source has both its interrupt enable bit and interrupt flag bit set, one of the following will occur:

- If the interrupt occurs **before** the execution of a SLEEP instruction, the SLEEP instruction will complete as a NOP. Therefore, the WDT and WDT postscaler will not be cleared, the $\overline{TO}$ bit will not be set and $\overline{PD}$ bits will not be cleared.

- If the interrupt occurs **during or after** the execution of a SLEEP instruction, the device will immediately wake up from sleep. The SLEEP instruction will be completely executed before the wake-up. Therefore, the WDT and WDT postscaler will be cleared, the $\overline{TO}$ bit will be set and the $\overline{PD}$ bit will be cleared.

Even if the flag bits were checked before executing a SLEEP instruction, it may be possible for flag bits to become set before the SLEEP instruction completes. To determine whether a SLEEP instruction executed, test the $\overline{PD}$ bit. If the $\overline{PD}$ bit is set, the SLEEP instruction was executed as a NOP.

To ensure that the WDT is cleared, a CLRWDT instruction should be executed before a SLEEP instruction.

8.13   Program Verification/Code Protection

If the code protection bit(s) have not been programmed, the on-chip program memory can be read out for verification purposes.

> **Note:** Microchip does not recommend code protecting widowed devices.

8.14   ID Locations

Four memory locations (2000h - 2003h) are designated as ID locations to store checksum or other code identification numbers. These locations are not accessible during normal execution but are readable and writable only during program/verify. Only the 4 least significant bits of ID location are usable.

For ROM devices, these values are submitted along with the ROM code.

8.15   In-Circuit Serial Programming

PIC16F8X microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data, and three other lines for power, ground, and the programming voltage. Customers can manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product, allowing the most recent firmware or custom firmware to be programmed.

The device is placed into a program/verify mode by holding the RB6 and RB7 pins low, while raising the $\overline{MCLR}$ pin from V_{IL} to V_{IHH} (see programming specification). RB6 becomes the programming clock and RB7 becomes the programming data. Both RB6 and RB7 are Schmitt Trigger inputs in this mode.

After reset, to place the device into programming/verify mode, the program counter (PC) points to location 00h. A 6-bit command is then supplied to the device, 14-bits of program data is then supplied to or from the device, using load or read-type instructions. For complete details of serial programming, please refer to the PIC16CXX Programming Specifications (Literature #DS30189).

FIGURE 8-20:   TYPICAL IN-SYSTEM SERIAL PROGRAMMING CONNECTION
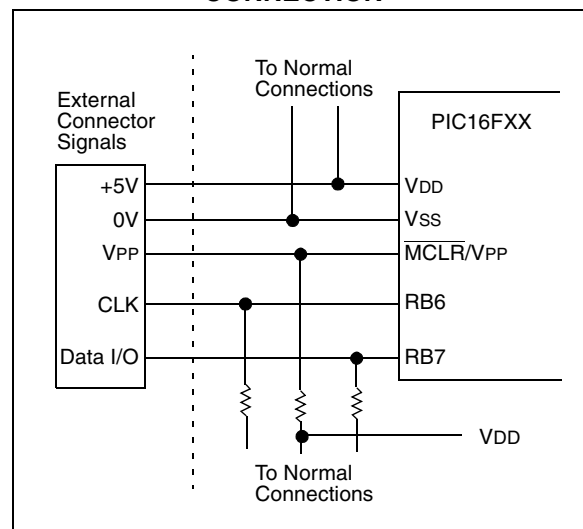
For ROM devices, both the program memory and Data EEPROM memory may be read, but only the Data EEPROM memory may be programmed.

INCFSZ	**Increment f, Skip if 0**
Syntax:	[*label*] INCFSZ f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	(f) + 1 $\rightarrow$ (destination), skip if result = 0
Status Affected:	None

Encoding:

00	1111	dfff	ffff

Description:	The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'. If the result is 1, the next instruction is executed. If the result is 0, a NOP is executed instead making it a 2TCY instruction.
Words:	1
Cycles:	1(2)

Q Cycle Activity:

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

If Skip: (2nd Cycle)

Q1	Q2	Q3	Q4
No-Operation	No-Operation	No-Operation	No-Operation

Example

```
HERE      INCFSZ      CNT, 1
          GOTO        LOOP
CONTINUE  •
          •
          •
```

Before Instruction
    PC   =   address HERE
After Instruction
    CNT   =   CNT + 1
    if CNT=   0,
    PC   =   address CONTINUE
    if CNT≠   0,
    PC   =   address HERE +1

IORLW	**Inclusive OR Literal with W**
Syntax:	[*label*] IORLW k
Operands:	$0 \leq k \leq 255$
Operation:	(W) .OR. k $\rightarrow$ (W)
Status Affected:	Z

Encoding:

11	1000	kkkk	kkkk

Description:	The contents of the W register is OR'ed with the eight bit literal 'k'. The result is placed in the W register.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal 'k'	Process data	Write to W

Example

```
IORLW   0x35
```
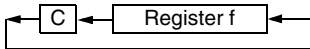
Before Instruction
    W   =   0x9A
After Instruction
    W   =   0xBF
    Z   =   1

RLF	**Rotate Left f through Carry**

Syntax:	[*label*] RLF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
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

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	Write to destination
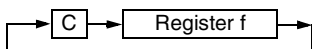
Example   RLF   REG1,0

Before Instruction
REG1	=	1110 0110
C	=	0

After Instruction
REG1	=	1110 0110
W	=	1100 1100
C	=	1

RRF	**Rotate Right f through Carry**

Syntax:	[*label*] RRF f,d
Operands:	$0 \leq f \leq 127$ $d \in [0,1]$
Operation:	See description below
Status Affected:	C

Encoding:

00	1100	dfff	ffff

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

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example   RRF   REG1,0

Before Instruction
REG1	=	1110 0110
C	=	0

After Instruction
REG1	=	1110 0110
W	=	0111 0011
C	=	0

SUBWF	Subtract W from f
Syntax:	[*label*] SUBWF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f) - (W) $\rightarrow$ (destination)
Status Affected:	C, DC, Z
Encoding:	`00` `0010` `dfff` `ffff`
Description:	Subtract (2's complement method) W register from register 'f'. If 'd' is 0 the result is stored in the W register. If 'd' is 1 the result is stored back in register 'f'.
Words:	1
Cycles:	1

Q Cycle Activity:	Q1	Q2	Q3	Q4
	Decode	Read register 'f'	Process data	Write to destination

Example 1:   SUBWF   REG1,1

Before Instruction

    REG1 = 3
    W   = 2
    C   = ?
    Z   = ?

After Instruction

    REG1 = 1
    W   = 2
    C   = 1; result is positive
    Z   = 0

Example 2:   Before Instruction

    REG1 = 2
    W   = 2
    C   = ?
    Z   = ?

After Instruction

    REG1 = 0
    W   = 2
    C   = 1; result is zero
    Z   = 1

Example 3:   Before Instruction

    REG1 = 1
    W   = 2
    C   = ?
    Z   = ?

After Instruction

    REG1 = 0xFF
    W   = 2
    C   = 0; result is negative
    Z   = 0

SWAPF	Swap Nibbles in f
Syntax:	[*label*] SWAPF f,d
Operands:	$0 \le f \le 127$ $d \in [0,1]$
Operation:	(f<3:0>) $\rightarrow$ (destination<7:4>), (f<7:4>) $\rightarrow$ (destination<3:0>)
Status Affected:	None
Encoding:	`00` `1110` `dfff` `ffff`
Description:	The upper and lower nibbles of register 'f' are exchanged. If 'd' is 0 the result is placed in W register. If 'd' is 1 the result is placed in register 'f'.
Words:	1
Cycles:	1

Q Cycle Activity:	Q1	Q2	Q3	Q4
	Decode	Read register 'f'	Process data	Write to destination

Example   SWAPF REG,   0

Before Instruction

    REG1 = 0xA5

After Instruction

    REG1 = 0xA5
    W   = 0x5A

TRIS	Load TRIS Register
Syntax:	[*label*] TRIS f
Operands:	$5 \le f \le 7$
Operation:	(W) $\rightarrow$ TRIS register f;
Status Affected:	None
Encoding:	`00` `0000` `0110` `0fff`
Description:	The instruction is supported for code compatibility with the PIC16C5X products. Since TRIS registers are readable and writable, the user can directly address them.
Words:	1
Cycles:	1
Example	

To maintain upward compatibility with future PIC16CXX products, do not use this instruction.

TABLE 10-1   CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	PIC16F84-04 PIC16F83-04	PIC16F84-10 PIC16F83-10	PIC16LF84-04 PIC16LF83-04
RC	VDD: 4.0V to 6.0V IDD: 4.5 mA max. at 5.5V IPD: 14 μA max. at 4V WDT dis Freq: 4.0 MHz max.	VDD: 4.5V to 5.5V IDD: 1.8 mA typ. at 5.5V IPD: 1.0 μA typ. at 5.5V WDT dis Freq: 4..0 MHz max.	VDD: 2.0V to 6.0V IDD: 4.5 mA max. at 5.5V IPD: 7.0 μA max. at 2V WDT dis Freq: 2.0 MHz max.
XT	VDD: 4.0V to 6.0V IDD: 4.5 mA max. at 5.5V IPD: 14 μA max. at 4V WDT dis Freq: 4.0 MHz max.	VDD: 4.5V to 5.5V IDD: 1.8 mA typ. at 5.5V IPD: 1.0 μA typ. at 5.5V WDT dis Freq: 4.0 MHz max.	VDD: 2.0V to 6.0V IDD: 4.5 mA max. at 5.5V IPD: 7.0 μA max. at 2V WDT dis Freq: 2.0 MHz max.
HS	VDD: 4.5V to 5.5V IDD: 4.5 mA typ. at 5.5V IPD: 1.0 μA typ. at 4.5V WDT dis Freq: 4.0 MHz max.	VDD: 4.5V to 5.5V IDD: 10 mA max. at 5.5V typ. IPD: 1.0 μA typ. at 4.5V WDT dis Freq: 10 MHz max.	Do not use in HS mode
LP	VDD: 4.0V to 6.0V IDD: 48 μA typ. at 32 kHz, 2.0V IPD: 0.6 μA typ. at 3.0V WDT dis Freq: 200 kHz max.	Do not use in LP mode	VDD: 2.0V to 6.0V IDD: 45 μA max. at 32 kHz, 2.0V IPD: 7 μA max. at 2.0V WDT dis Freq: 200 kHz max.

The shaded sections indicate oscillator selections which are tested for functionality, but not for MIN/MAX specifications. It is recommended that the user select the device type that ensures the specifications required.

11.0   ELECTRICAL CHARACTERISTICS FOR PIC16CR83 AND PIC16CR84

Absolute Maximum Ratings †

Ambient temperature under bias...-55°C to +125°C

Storage temperature .... -65°C to +150°C

Voltage on VDD with respect to VSS   .... -0.3 to +7.5V

Voltage on $\overline{\text{MCLR}}$ with respect to VSS**(2)** ....-0.3 to +14V

Voltage on any pin with respect to VSS (except VDD and $\overline{\text{MCLR}}$)... -0.6V to (VDD + 0.6V)

Total power dissipation**(1)** ....800 mW

Maximum current out of VSS pin ....150 mA

Maximum current into VDD pin ....100 mA

Input clamp current, IIK (VI < 0 or VI > VDD)..± 20 mA

Output clamp current, IOK (VO < 0 or VO > VDD) ....± 20 mA

Maximum output current sunk by any I/O pin...25 mA

Maximum output current sourced by any I/O pin ....20 mA

Maximum current sunk by PORTA ....80 mA

Maximum current sourced by PORTA..50 mA

Maximum current sunk by PORTB...150 mA

Maximum current sourced by PORTB..100 mA

> **Note 1:** Power dissipation is calculated as follows: Pdis = VDD x {IDD - ∑ IOH} + ∑ {(VDD-VOH) x IOH} + ∑(VOl x IOL)
>
> **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.
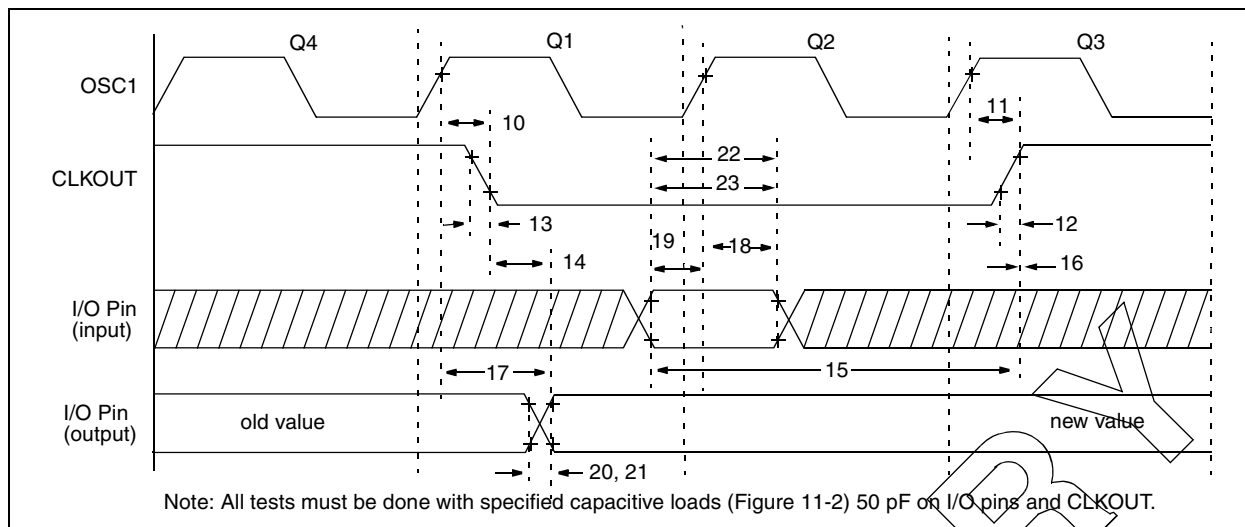
FIGURE 11-4: CLKOUT AND I/O TIMING

Note: All tests must be done with specified capacitive loads (Figure 11-2) 50 pF on I/O pins and CLKOUT.

TABLE 11-4   CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
10	TosH2ckL	OSC1↑ to CLKOUT↓	PIC16CR8X	—	15	30 *	ns	Note 1
10A			PIC16LCR8X	—	15	120 *	ns	Note 1
11	TosH2ckH	OSC1↑ to CLKOUT↑	PIC16CR8X	—	15	30 *	ns	Note 1
11A			PIC16LCR8X	—	15	120 *	ns	Note 1
12	TckR	CLKOUT rise time	PIC16CR8X	—	15	30 *	ns	Note 1
12A			PIC16LCR8X	—	15	100 *	ns	Note 1
13	TckF	CLKOUT fall time	PIC16CR8X	—	15	30 *	ns	Note 1
13A			PIC16LCR8X	—	15	100 *	ns	Note 1
14	TckL2ioV	CLKOUT ↓ to Port out valid		—	—	0.5TCY +20 *	ns	Note 1
15	TioV2ckH	Port in valid before	PIC16CR8X	0.30TCY + 30 *	—	—	ns	Note 1
		CLKOUT↑	PIC16LCR8X	0.30TCY + 80 *	—	—	ns	Note 1
16	TckH2ioI	Port in hold after CLKOUT ↑		0 *	—	—	ns	Note 1
17	TosH2ioV	OSC1↑ (Q1 cycle) to	PIC16CR8X	—	—	125 *	ns	
		Port out valid	PIC16LCR8X	—	—	250 *	ns	
18	TosH2ioI	OSC1↑ (Q2 cycle) to	PIC16CR8X	10 *	—	—	ns	
		Port input invalid (I/O in hold time)	PIC16LCR8X	10 *	—	—	ns	
19	TioV2osH	Port input valid to	PIC16CR8X	-75 *	—	—	ns	
		OSC1↑ (I/O in setup time)	PIC16LCR8X	-175 *	—	—	ns	
20	TioR	Port output rise time	PIC16CR8X	—	10	35 *	ns	
20A			PIC16LCR8X	—	10	70 *	ns	
21	TioF	Port output fall time	PIC16CR8X	—	10	35 *	ns	
21A			PIC16LCR8X	—	10	70 *	ns	
22	Tinp	INT pin high	PIC16CR8X	20 *	—	—	ns	
22A		or low time	PIC16LCR8X	55 *	—	—	ns	
23	Trbp	RB7:RB4 change INT	PIC16CR8X	TOSC §	—	—	ns	
23A		high or low time	PIC16LCR8X	TOSC §	—	—	ns	

*   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.

§   By design

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x TOSC.

PIC16F8X

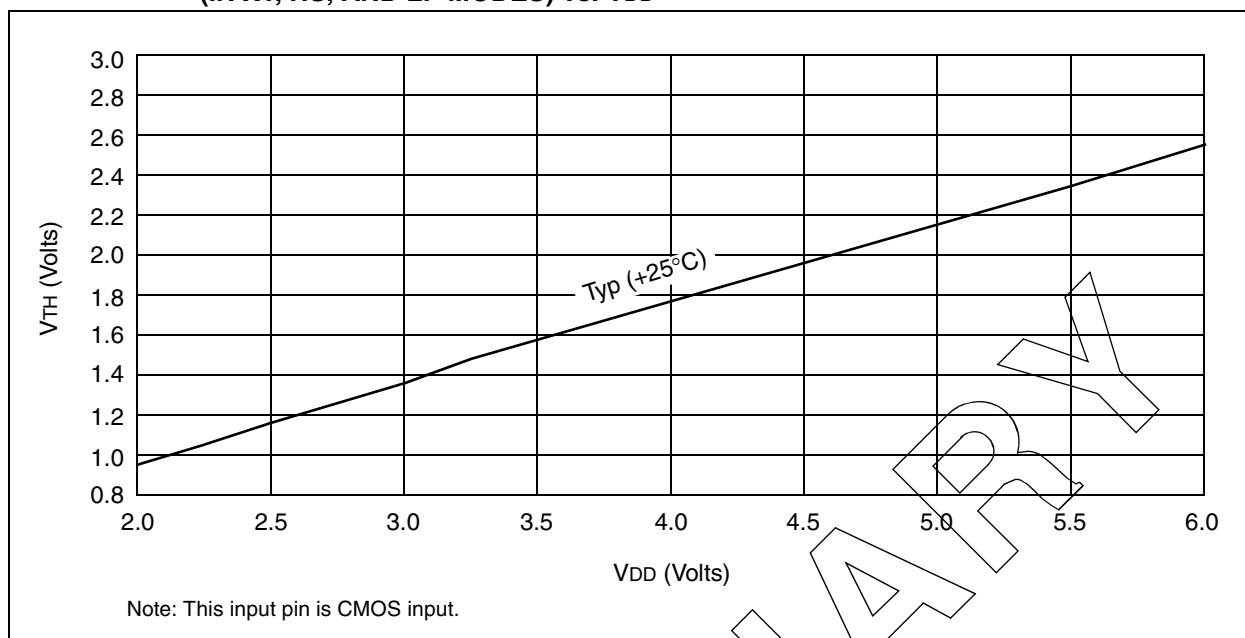**FIGURE 12-8:** Vᴛʜ (INPUT THRESHOLD VOLTAGE) OF OSC1 INPUT (IN XT, HS, AND LP MODES) vs. Vᴅᴅ

Note: This input pin is CMOS input.

FIGURE 12-9: Vɪʜ, Vɪʟ OF M̄C̄L̄R̄, T0CKI AND OSC1 (IN RC MODE) vs. Vᴅᴅ

Note: These input pins have Schmitt Trigger input buffers.
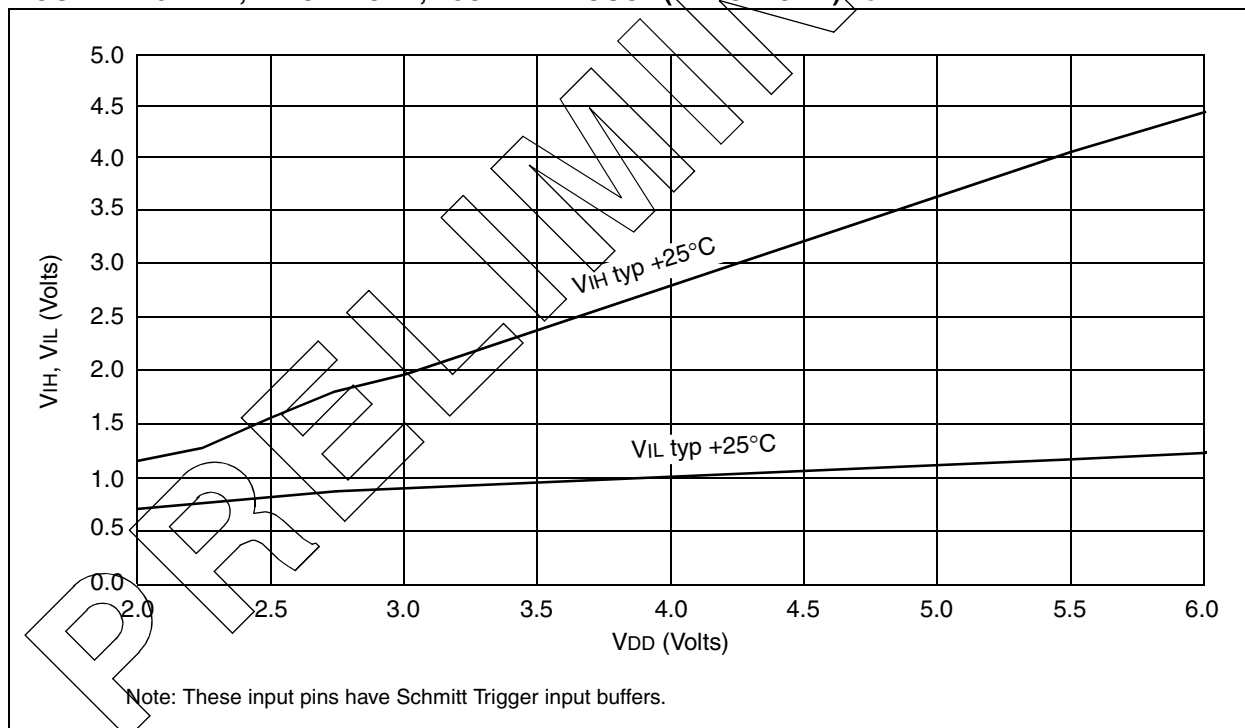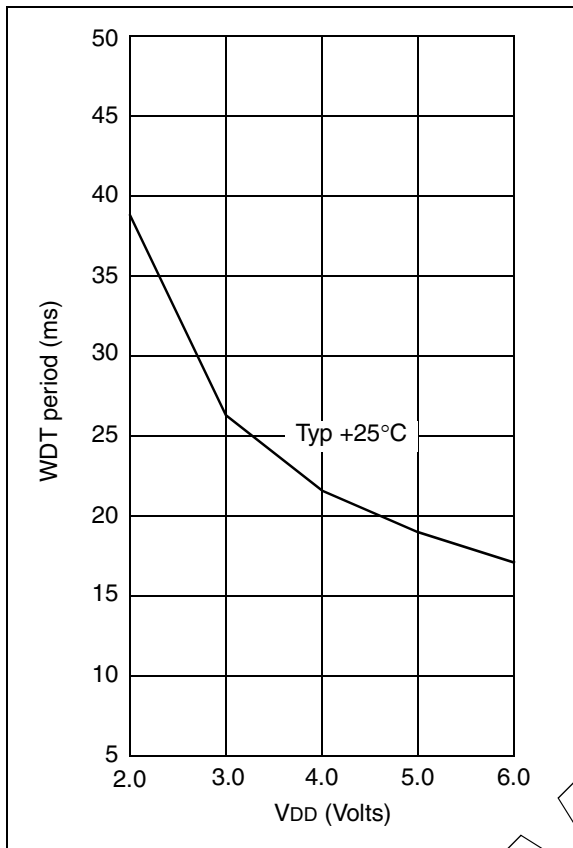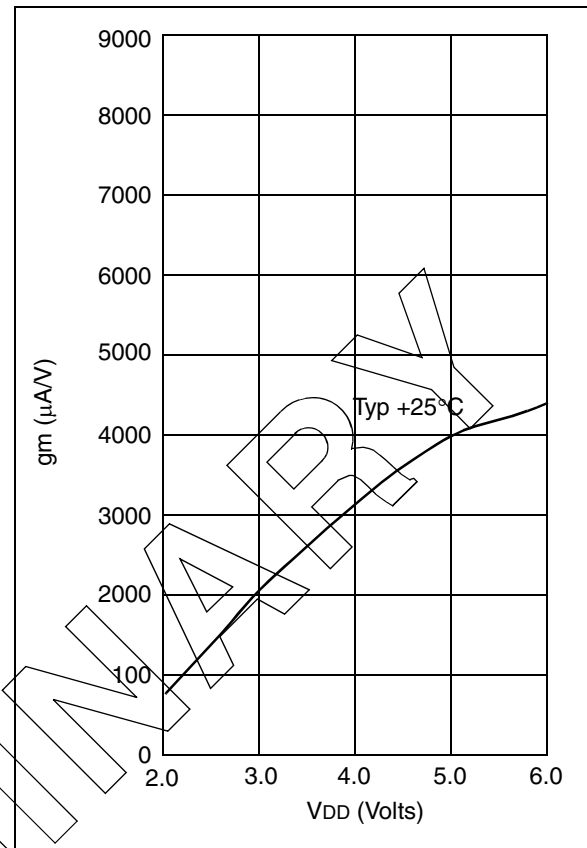
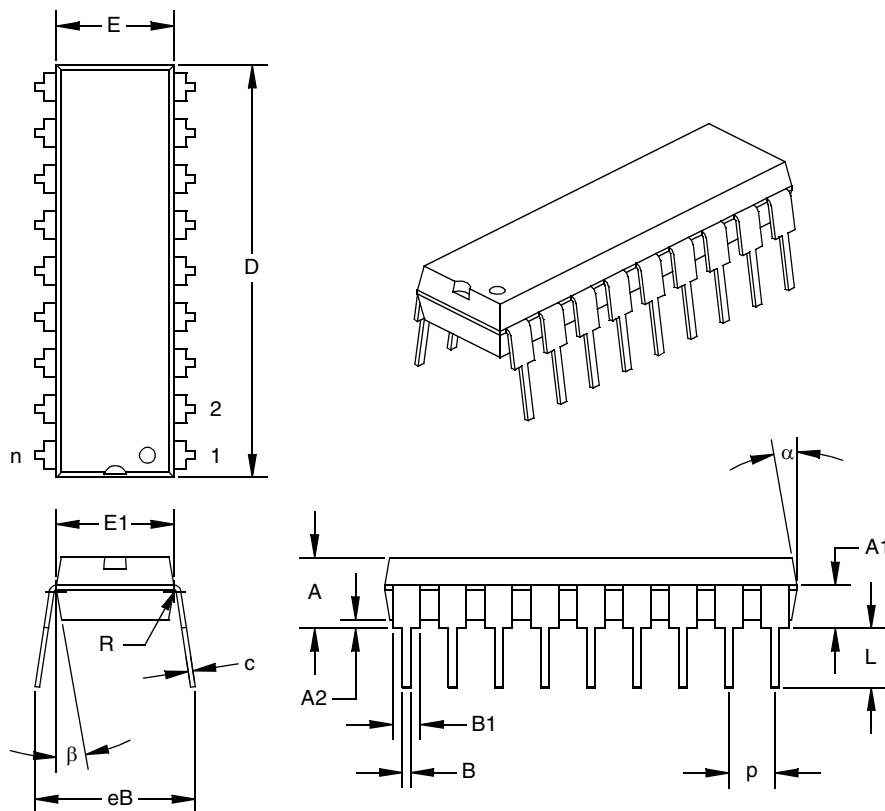**FIGURE 12-13: WDT TIMER TIME-OUT PERIOD vs. VDD**

FIGURE 12-14: TRANSCONDUCTANCE (gm) OF HS OSCILLATOR vs. VDD

PIC16F8X

Package Type: K04-007 18-Lead Plastic Dual In-line (P) – 300 mil

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
PCB Row Spacing			0.300			7.62	
Number of Pins	n		18			18	
Pitch	p		0.100			2.54	
Lower Lead Width	B	0.013	0.018	0.023	0.33	0.46	0.58
Upper Lead Width	B1†	0.055	0.060	0.065	1.40	1.52	1.65
Shoulder Radius	R	0.000	0.005	0.010	0.00	0.13	0.25
Lead Thickness	c	0.005	0.010	0.015	0.13	0.25	0.38
Top to Seating Plane	A	0.110	0.155	0.155	2.79	3.94	3.94
Top of Lead to Seating Plane	A1	0.075	0.095	0.115	1.91	2.41	2.92
Base to Seating Plane	A2	0.000	0.020	0.020	0.00	0.51	0.51
Tip to Seating Plane	L	0.125	0.130	0.135	3.18	3.30	3.43
Package Length	D‡	0.890	0.895	0.900	22.61	22.73	22.86
Molded Package Width	E‡	0.245	0.255	0.265	6.22	6.48	6.73
Radius to Radius Width	E1	0.230	0.250	0.270	5.84	6.35	6.86
Overall Row Spacing	eB	0.310	0.349	0.387	7.87	8.85	9.83
Mold Draft Angle Top	α	5	10	15	5	10	15
Mold Draft Angle Bottom	β	5	10	15	5	10	15

* Controlling Parameter.

† Dimension "B1" does not include dam-bar protrusions. Dam-bar protrusions shall not exceed 0.003" (0.076 mm) per side or 0.006" (0.152 mm) more than dimension "B1."

‡ Dimensions "D" and "E" do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.010" (0.254 mm) per side or 0.020" (0.508 mm) more than dimensions "D" or "E."

NOTES:

APPENDIX E: CONVERSION CONSIDERATIONS - PIC16C84 TO PIC16F83/F84 AND PIC16CR83/CR84

Considerations for converting from the PIC16C84 to the PIC16F84 are listed in the table below. These considerations apply to converting from the PIC16C84 to the PIC16F83 (same as PIC16F84 except for program and data RAM memory sizes) and the PIC16CR84 and PIC16CR83 (ROM versions of Flash devices). Development Systems support is available for all of the PIC16X8X devices.

Difference	PIC16C84	PIC16F84
The polarity of the PWRTE bit has been reversed. Ensure that the programmer has this bit correctly set before programming.	PWRTE	$\overline{\text{PWRTE}}$
The PIC16F84 (and PIC16CR84) have larger RAM sizes. Ensure that this does not cause an issue with your program.	RAM = 36 bytes	RAM = 68 bytes
The $\overline{\text{MCLR}}$ pin now has an on-chip filter. The input signal on the $\overline{\text{MCLR}}$ pin will require a longer low pulse to generate an interrupt.	$\overline{\text{MCLR}}$ pulse width (low) = 350ns; 2.0V ≤ VDD ≤ 3.0V = 150ns; 3.0V ≤ VDD ≤ 6.0V	$\overline{\text{MCLR}}$ pulse width (low) = 1000ns; 2.0V ≤ VDD ≤ 6.0V
Some electrical specifications have been improved (see IPD example). Compare the electrical specifications of the two devices to ensure that this will not cause a compatibility issue.	IPD (typ @ 2V) = 26μA IPD (max @ 4V, WDT disabled) =100μA (PIC16C84) =100μA (PIC16LC84)	IPD (typ @ 2V) < 1μA IPD (max @ 4V, WDT disabled) =14μA (PIC16F84) =7μA (PIC16LF84)
PORTA and crystal oscillator values less than 500kHz	For crystal oscillator configurations operating below 500kHz, the device may generate a spurious internal Q-clock when PORTA<0> switches state.	N/A
RB0/INT pin	TTL	TTL/ST* (* This buffer is a Schmitt Trigger input when configured as the external interrupt.)
EEADR<7:6> and IDD	It is recommended that the EEADR<7:6> bits be cleared. When either of these bits is set, the maximum IDD for the device is higher than when both are cleared.	N/A
Code Protect	1 CP bit	9 CP bits
Recommended value of REXT for RC oscillator circuits	REXT = 3kΩ - 100kΩ	REXT = 5kΩ - 100kΩ
GIE bit unintentional enable	If an interrupt occurs while the Global Interrupt Enable (GIE) bit is being cleared, the GIE bit may unintentionally be re-enabled by the user's Interrupt Service Routine (the RETFIE instruction).	N/A