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	Brown-out Detect/Reset, PWM, WDT
Number of I/O	13
Program Memory Size	1.75KB (1K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	68 x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 6V
Data Converters	A/D 4x8b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°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/pic16lc711-04i-so

FIGURE 4-5:   PIC16C711 REGISTER FILE MAP

File Address	Bank 0	Bank 1	File Address
00h	INDF[1]	INDF[1]	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h		PCON	87h
08h	ADCON0	ADCON1	88h
09h	ADRES	ADRES	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	General Purpose Register	General Purpose Register Mapped in Bank 0[2]	8Ch
4Fh			CFh
50h			D0h
7Fh			FFh

Unimplemented data memory locations, read as '0'.

Note 1:   Not a physical register.
  2:   These locations are unimplemented in Bank 1. Any access to these locations will access the corresponding Bank 0 register.

FIGURE 4-6:   PIC16C715 REGISTER FILE MAP

File Address	Bank 0	Bank 1	File Address
00h	INDF[1]	INDF[1]	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h			87h
08h			88h
09h			89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	PIR1	PIE1	8Ch
0Dh			8Dh
0Eh		PCON	8Eh
0Fh			8Fh
10h			90h
11h			91h
12h			92h
13h			93h
14h			94h
15h			95h
16h			96h
17h			97h
18h			98h
19h			99h
1Ah			9Ah
1Bh			9Bh
1Ch			9Ch
1Dh			9Dh
1Eh	ADRES		9Eh
1Fh	ADCON0	ADCON1	9Fh
20h	General Purpose Register	General Purpose Register	A0h
			BFh
			C0h
7Fh			FFh

Unimplemented data memory locations, read as '0'.

Note 1:   Not a physical register.

4.2.2.3   INTCON REGISTER

Applicable Devices	710	71	711	715

The INTCON Register is a readable and writable register which contains various enable and flag bits for the TMR0 register overflow, RB Port change and External RB0/INT pin interrupts.

> **Note:** Interrupt flag bits get set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>).
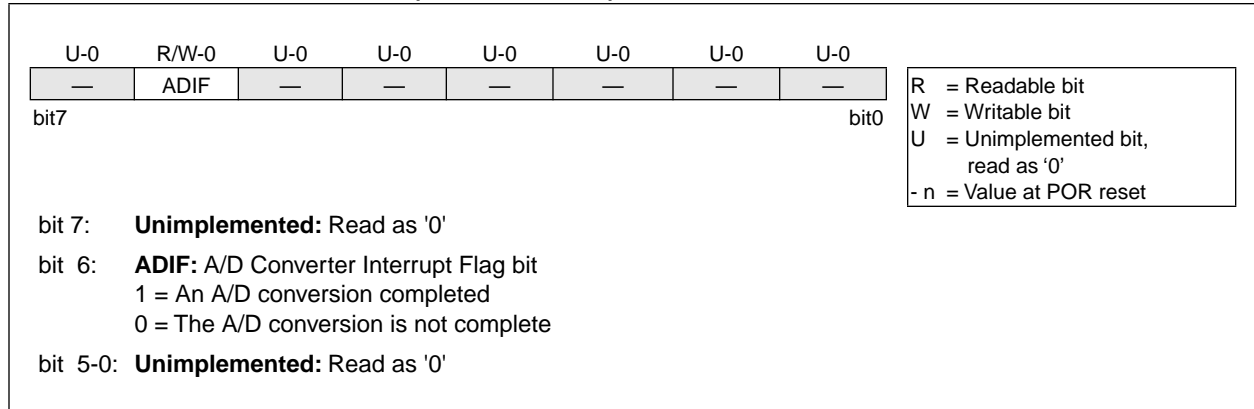
FIGURE 4-9:   INTCON REGISTER (ADDRESS 0Bh, 8Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	ADIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF

bit7   bit0

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

bit 7:   **GIE:**[1] Global Interrupt Enable bit
1 = Enables all un-masked interrupts
0 = Disables all interrupts

bit 6:   **ADIE:** A/D Converter Interrupt Enable bit
1 = Enables A/D interrupt
0 = Disables A/D interrupt

bit 5:   **T0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

bit 4:   **INTE:** RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt

bit 3:   **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt

bit 2:   **T0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1:   **INTF:** RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

bit 0:   **RBIF:** RB Port Change Interrupt Flag bit
1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

Note 1:   For the PIC16C71, if an interrupt occurs while the GIE bit is being cleared, the GIE bit may be unintentionally re-enabled by the RETFIE instruction in the user's Interrupt Service Routine. Refer to Section 8.5 for a detailed description.

Interrupt flag bits get set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

4.2.2.5 PIR1 REGISTER

| Applicable Devices | 710 | 71 | 711 | 715 |

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

Note: Interrupt flag bits get set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global enable bit, GIE (INTCON<7>). User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

FIGURE 4-11: PIR1 REGISTER (ADDRESS 0Ch)

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	ADIF	—	—	—	—	—	—

bit7 .... bit0

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

bit 7: **Unimplemented:** Read as '0'

bit 6: **ADIF:** A/D Converter Interrupt Flag bit
1 = An A/D conversion completed
0 = The A/D conversion is not complete

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

PIC16C71X

| **Applicable Devices** | 710 | 71 | 711 | 715 |

The Power Control (PCON) register contains a flag bit to allow differentiation between a Power-on Reset (POR) to an external $\overline{\text{MCLR}}$ Reset or WDT Reset. Those devices with brown-out detection circuitry contain an additional bit to differentiate a Brown-out Reset (BOR) condition from a Power-on Reset condition. For the PIC16C715 the PCON register also contains status bits MPEEN and PER. MPEEN reflects the value of the MPEEN bit in the configuration word. PER indicates a parity error reset has occurred.

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

FIGURE 4-12:   PCON REGISTER (ADDRESS 8Eh), PIC16C710/711

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-q
—	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$

bit7 .... bit0

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

bit 7-2:   **Unimplemented:** Read as '0'

bit 1: $\overline{\text{POR}}$**:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0: $\overline{\text{BOR}}$**:** Brown-out Reset Status bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

FIGURE 4-13:   PCON REGISTER (ADDRESS 8Eh), PIC16C715

R-U	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-q
MPEEN	—	—	—	—	$\overline{\text{PER}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$[1]

bit7 .... bit0

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

bit 7:   **MPEEN:** Memory Parity Error Circuitry Status bit
Reflects the value of configuration word bit, MPEEN

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

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

bit 1: $\overline{\text{POR}}$**:** Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0: $\overline{\text{BOR}}$**:** Brown-out Reset Status bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

PIC16C71X

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 (e.g., bit0) and it is defined as an input at this time, the input signal present on the pin itself would be read into the CPU and rewritten to the data latch of this particular pin, overwriting the previous content. As long as the pin stays in the input mode, no problem occurs. However, if bit0 is switched to an output, the content of the data latch may now be unknown.

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

Example 5-3 shows the effect of two sequential read-modify-write instructions on an I/O port.

EXAMPLE 5-3: READ-MODIFY-WRITE INSTRUCTIONS ON AN I/O PORT

```
;Initial PORT settings: PORTB<7:4> Inputs
;                       PORTB<3:0> Outputs
;PORTB<7:6> have external pull-ups and are
;not connected to other circuitry
;
;                   PORT latch  PORT pins
;                   ----------  ---------
  BCF PORTB, 7  ; 01pp pppp     11pp pppp
  BCF PORTB, 6  ; 10pp pppp     11pp pppp
  BSF STATUS, RP0 ;
  BCF TRISB, 7  ; 10pp pppp     11pp pppp
  BCF TRISB, 6  ; 10pp pppp     10pp pppp
;
;Note that the user may have expected the
;pin values to be 00pp ppp. The 2nd BCF
;caused RB7 to be latched as the pin value
;(high).
```

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

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 5-6). Therefore, care must be exercised if a write followed by a read operation is carried out on the same I/O port. The sequence of instructions should be such to allow the pin voltage to stabilize (load dependent) before the next instruction which causes that file to be read into the CPU is executed. Otherwise, the previous state of that pin may be read into the CPU rather than the new state. When in doubt, it is better to separate these instructions with a NOP or another instruction not accessing this I/O port.

FIGURE 5-6: SUCCESSIVE I/O OPERATION

Note:

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

Note that:

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

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

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

The ADRES register contains the result of the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRES register, the GO/$\overline{\text{DONE}}$ bit (ADCON0<2>) is cleared, and A/D interrupt flag bit ADIF is set. The block diagram of the A/D module is shown in Figure 7-4.

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as an input. To determine acquisition time, see Section 7.1. After this acquisition time has elapsed the A/D conversion can be started. The following steps should be followed for doing an A/D conversion:

1. Configure the A/D module:
   - Configure analog pins / voltage reference / and digital I/O (ADCON1)
   - Select A/D input channel (ADCON0)
   - Select A/D conversion clock (ADCON0)
   - Turn on A/D module (ADCON0)

2. Configure A/D interrupt (if desired):
   - Clear ADIF bit
   - Set ADIE bit
   - Set GIE bit
3. Wait the required acquisition time.
4. Start conversion:
   - Set GO/$\overline{\text{DONE}}$ bit (ADCON0)
5. Wait for A/D conversion to complete, by either:
   - Polling for the GO/$\overline{\text{DONE}}$ bit to be cleared

   OR
   - Waiting for the A/D interrupt
6. Read A/D Result register (ADRES), clear bit ADIF if required.
7. For next conversion, go to step 1 or step 2 as required. The A/D conversion time per bit is defined as T_{AD}. A minimum wait of $2T_{AD}$ is required before next acquisition starts.

FIGURE 7-4: **A/D BLOCK DIAGRAM**

7.9 Transfer Function

The ideal transfer function of the A/D converter is as follows: the first transition occurs when the analog input voltage (VAIN) is Analog VREF/256 (Figure 7-6).

7.10 References

A very good reference for understanding A/D converters is the "Analog-Digital Conversion Handbook" third edition, published by Prentice Hall (ISBN 0-13-03-2848-0).

FIGURE 7-6:   A/D TRANSFER FUNCTION
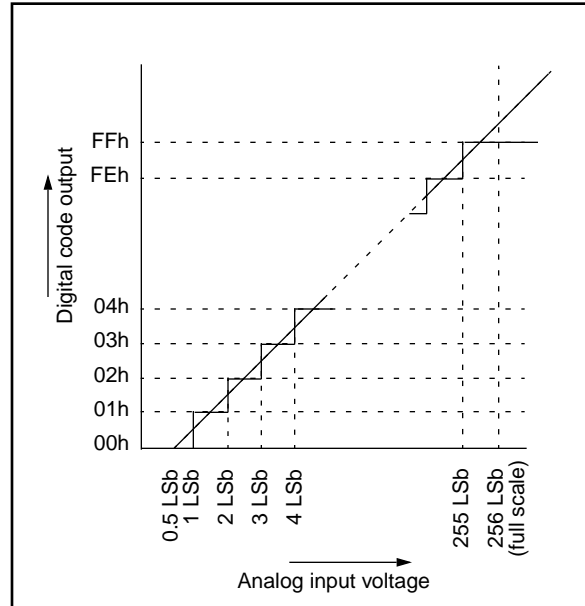
FIGURE 7-7:   FLOWCHART OF A/D OPERATION
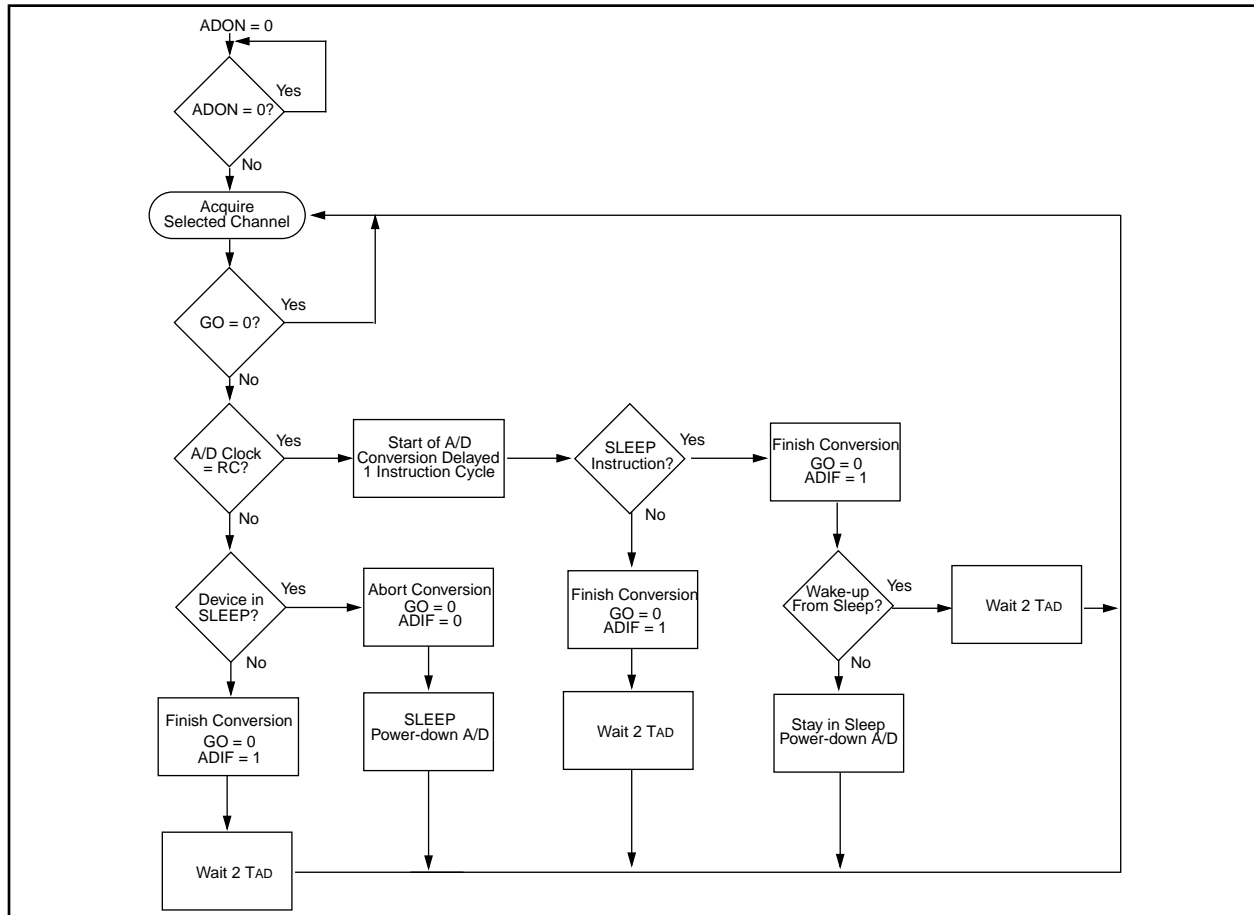
8.2 Oscillator Configurations

8.2.1 OSCILLATOR TYPES

The PIC16CXX can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

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

8.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In XT, LP or HS modes a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 8-4). The PIC16CXX Oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1/CLKIN pin (Figure 8-5).

FIGURE 8-4: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)

See Table 8-1 and Table 8-1 for recommended values of C1 and C2.

Note 1: A series resistor may be required for AT strip cut crystals.

2: The buffer is on the OSC2 pin.

FIGURE 8-5: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

TABLE 8-1: CERAMIC RESONATORS, PIC16C71

Ranges Tested:			
Mode	**Freq**	**OSC1**	**OSC2**
XT	455 kHz	47 - 100 pF	47 - 100 pF
	2.0 MHz	15 - 68 pF	15 - 68 pF
	4.0 MHz	15 - 68 pF	15 - 68 pF
HS	8.0 MHz	15 - 68 pF	15 - 68 pF
	16.0 MHz	10 - 47 pF	10 - 47 pF
These values are for design guidance only. See notes at bottom of page.			
Resonators Used:			
455 kHz	Panasonic EFO-A455K04B	± 0.3%	
2.0 MHz	Murata Erie CSA2.00MG	± 0.5%	
4.0 MHz	Murata Erie CSA4.00MG	± 0.5%	
8.0 MHz	Murata Erie CSA8.00MT	± 0.5%	
16.0 MHz	Murata Erie CSA16.00MX	± 0.5%	
All resonators used did not have built-in capacitors.			

TABLE 8-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR, PIC16C71

Mode	**Freq**	**OSC1**	**OSC2**
LP	32 kHz	33 - 68 pF	33 - 68 pF
	200 kHz	15 - 47 pF	15 - 47 pF
XT	100 kHz	47 - 100 pF	47 - 100 pF
	500 kHz	20 - 68 pF	20 - 68 pF
	1 MHz	15 - 68 pF	15 - 68 pF
	2 MHz	15 - 47 pF	15 - 47 pF
	4 MHz	15 - 33 pF	15 - 33 pF
HS	8 MHz	15 - 47 pF	15 - 47 pF
	20 MHz	15 - 47 pF	15 - 47 pF
These values are for design guidance only. See notes at bottom of page.			

8.5   Interrupts

Applicable Devices | 710 | 71 | 711 | 715 |

The PIC16C71X family has 4 sources of interrupt.

Interrupt Sources
External interrupt RB0/INT
TMR0 overflow interrupt
PORTB change interrupts (pins RB7:RB4)
A/D Interrupt

The interrupt control register (INTCON) records individual interrupt requests in flag bits. It also has individual and global interrupt enable bits.

> **Note:** Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

A global interrupt enable bit, GIE (INTCON<7>) enables (if set) all un-masked interrupts or disables (if cleared) all interrupts. When bit GIE is enabled, and an interrupt's flag bit and mask bit are set, the interrupt will vector immediately. Individual interrupts can be disabled through their corresponding enable bits in various registers. Individual interrupt bits are set regardless of the status of the GIE bit. The GIE bit is cleared on reset.

The "return from interrupt" instruction, RETFIE, exits the interrupt routine as well as sets the GIE bit, which re-enables interrupts.

The RB0/INT pin interrupt, the RB port change interrupt and the TMR0 overflow interrupt flags are contained in the INTCON register.

The peripheral interrupt flags are contained in the special function registers PIR1 and PIR2. The corresponding interrupt enable bits are contained in special function registers PIE1 and PIE2, and the peripheral interrupt enable bit is contained in special function register INTCON.

When an interrupt is responded to, the GIE bit is cleared to disable any further interrupt, the return address is pushed onto the stack and the PC is loaded with 0004h. Once in the interrupt service routine the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bit(s) must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

For external interrupt events, such as the INT pin or PORTB change interrupt, the interrupt latency will be three or four instruction cycles. The exact latency depends when the interrupt event occurs (Figure 8-19). The latency is the same for one or two cycle instructions. Individual interrupt flag bits are set regardless of the status of their corresponding mask bit or the GIE bit.

> **Note:** For the PIC16C71
> 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). The events that would cause this to occur are:
>
> 1. An instruction clears the GIE bit while an interrupt is acknowledged.
>
> 2. The program branches to the Interrupt vector and executes the Interrupt Service Routine.
>
> 3. The Interrupt Service Routine completes with the execution of the RETFIE instruction. This causes the GIE bit to be set (enables interrupts), and the program returns to the instruction after the one which was meant to disable interrupts.
>
> Perform the following to ensure that interrupts are globally disabled:
>
> ```
> LOOP BCF   INTCON, GIE  ; Disable global
>                         ;  interrupt bit
>      BTFSC INTCON, GIE  ; Global interrupt
>                         ;   disabled?
>      GOTO  LOOP         ; NO, try again
>         :               ;   Yes, continue
>                         ;   with program
>                         ;   flow
> ```

FIGURE 8-22:   WAKE-UP FROM SLEEP THROUGH INTERRUPT

Note   1:   XT, HS or LP oscillator mode assumed.
    2:   T$_{OST}$ = 1024T$_{OSC}$ (drawing not to scale) This delay will not be there for RC osc mode.
    3:   GIE = '1' assumed. In this case after wake- up, the processor jumps to the interrupt routine. If GIE = '0', execution will continue in-line.
    4:   CLKOUT is not available in these osc modes, but shown here for timing reference.

8.9   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 windowed devices.

8.10   ID Locations

Four memory locations (2000h - 2003h) are designated as ID locations where the user can store checksum or other code-identification numbers. These locations are not accessible during normal execution but are readable and writable during program/verify. It is recommended that only the 4 least significant bits of the ID location are used.

8.11   In-Circuit Serial Programming

PIC16CXX 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. This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a 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 MCLR (VPP) pin from VIL to VIHH (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) is at location 00h. A 6-bit command is then supplied to the device. Depending on the command, 14-bits of program data are then supplied to or from the device, depending if the command was a load or a read. For complete details of serial programming, please refer to the PIC16C6X/7X Programming Specifications (Literature #DS30228).

FIGURE 8-23:   TYPICAL IN-CIRCUIT SERIAL PROGRAMMING CONNECTION

PIC16C71X

GOTO	Unconditional Branch
Syntax:	[*label*] GOTO k
Operands:	0 ≤ k ≤ 2047
Operation:	k → PC<10:0> PCLATH<4:3> → PC<12:11>
Status Affected:	None

Encoding:

10	1kkk	kkkk	kkkk

Description: GOTO is an unconditional branch. The eleven bit immediate value is loaded into PC bits <10:0>. The upper bits of PC are loaded from PCLATH<4:3>. GOTO is a two cycle instruction.

Words: 1

Cycles: 2

Q Cycle Activity:

	Q1	Q2	Q3	Q4
1st Cycle	Decode	Read literal 'k'	Process data	Write to PC
2nd Cycle	NOP	NOP	NOP	NOP

Example   GOTO THERE

After Instruction

PC	=	Address THERE

INCF	Increment f
Syntax:	[*label*] INCF f,d
Operands:	0 ≤ f ≤ 127 d ∈ [0,1]
Operation:	(f) + 1 → (dest)
Status Affected:	Z

Encoding:

00	1010	dfff	ffff

Description: The contents of register 'f' are incremented. If 'd' is 0 the result is placed in the W register. If 'd' is 1 the result is placed back in register 'f'.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example   INCF   CNT, 1

Before Instruction

CNT	=	0xFF
Z	=	0

After Instruction

CNT	=	0x00
Z	=	1

10.6 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-Plus 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.

10.7 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-Plus, 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 I^{2}C bus and separate headers for connection to an LCD module and a keypad.

10.8 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.

10.9 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.

10.10 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.

MPASM allows full symbolic debugging from PICMASTER, Microchip's Universal Emulator System.

PIC16C71X

DC CHARACTERISTICS	Standard Operating Conditions (unless otherwise stated) Operating temperature $\quad 0°C \quad \leq T_A \leq +70°C$ (commercial) $\quad\quad\quad\quad\quad\quad\quad\quad -40°C \quad \leq T_A \leq +85°C$ (industrial) $\quad\quad\quad\quad\quad\quad\quad\quad -40°C \quad \leq T_A \leq +125°C$ (extended) Operating voltage V_{DD} range as described in DC spec Section 11.1 and Section 11.2.						

Param No.	Characteristic	Sym	Min	Typ †	Max	Units	Conditions
	Output Low Voltage						
D080	I/O ports	V_{OL}	-	-	0.6	V	I_{OL} = 8.5 mA, V_{DD} = 4.5V, -40°C to +85°C
D080A			-	-	0.6	V	I_{OL} = 7.0 mA, V_{DD} = 4.5V, -40°C to +125°C
D083	OSC2/CLKOUT (RC osc config)		-	-	0.6	V	I_{OL} = 1.6 mA, V_{DD} = 4.5V, -40°C to +85°C
D083A			-	-	0.6	V	I_{OL} = 1.2 mA, V_{DD} = 4.5V, -40°C to +125°C
	Output High Voltage						
D090	I/O ports (Note 3)	V_{OH}	V_{DD} - 0.7	-	-	V	I_{OH} = -3.0 mA, V_{DD} = 4.5V, -40°C to +85°C
D090A			V_{DD} - 0.7	-	-	V	I_{OH} = -2.5 mA, V_{DD} = 4.5V, -40°C to +125°C
D092	OSC2/CLKOUT (RC osc config)		V_{DD} - 0.7	-	-	V	I_{OH} = -1.3 mA, V_{DD} = 4.5V, -40°C to +85°C
D092A			V_{DD} - 0.7	-	-	V	I_{OH} = -1.0 mA, V_{DD} = 4.5V, -40°C to +125°C
D130*	**Open-Drain High Voltage**	V_{OD}	-	-	14	V	RA4 pin
	Capacitive Loading Specs on Output Pins						
D100	OSC2 pin	C_{OSC2}	-	-	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1.
D101	All I/O pins and OSC2 (in RC mode)	C_{IO}	-	-	50	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.

Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16C7X be driven with external clock in RC mode.

  2: The leakage current on the $\overline{MCLR}$ 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 current sourced by the pin.

PIC16C71X

FIGURE 12-3: TYPICAL I_{PD} vs. V_{DD} @ 25°C (WDT ENABLED, RC MODE)
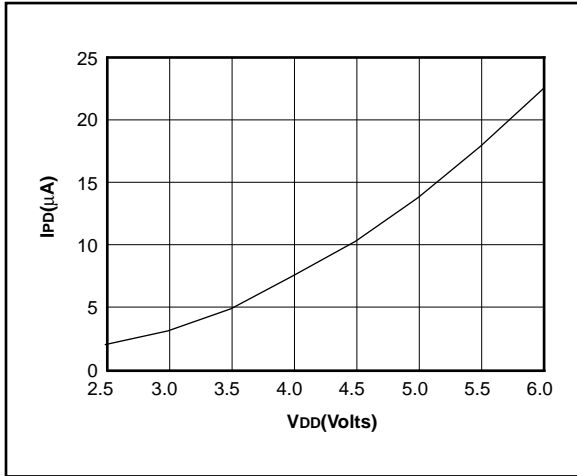
FIGURE 12-4: MAXIMUM I_{PD} vs. V_{DD} (WDT ENABLED, RC MODE)
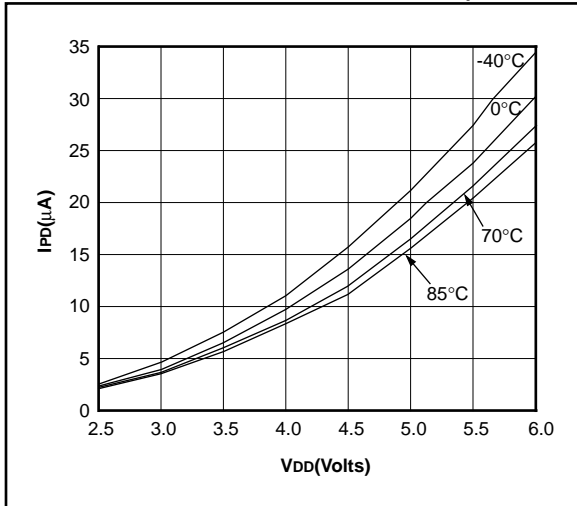
FIGURE 12-5: TYPICAL RC OSCILLATOR FREQUENCY vs. V_{DD}

Shaded area is beyond recommended range.
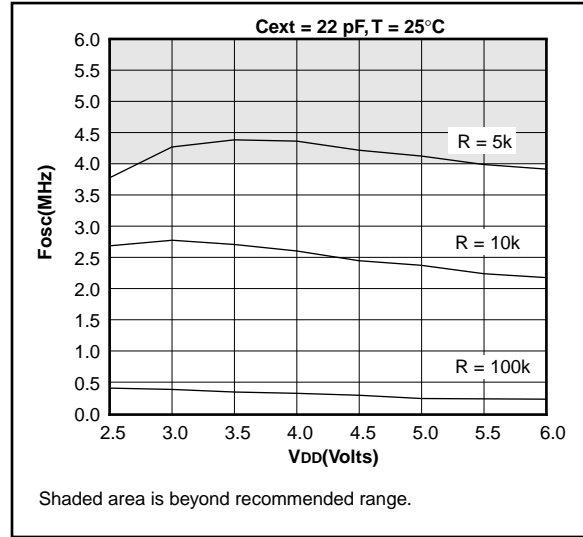
FIGURE 12-6: TYPICAL RC OSCILLATOR FREQUENCY vs. V_{DD}
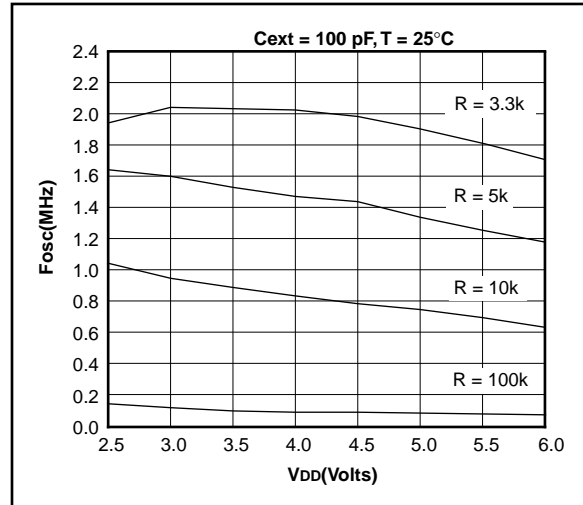
FIGURE 12-7: TYPICAL RC OSCILLATOR FREQUENCY vs. V_{DD}

PIC16C71X
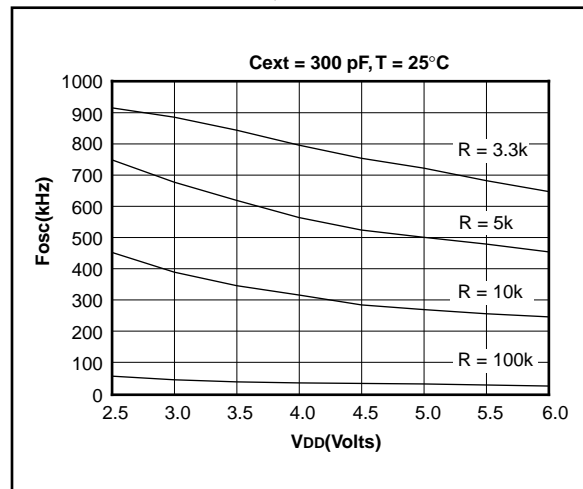
FIGURE 12-18: TYPICAL I$_{DD}$ vs. CAPACITANCE @ 500 kHz (RC MODE)
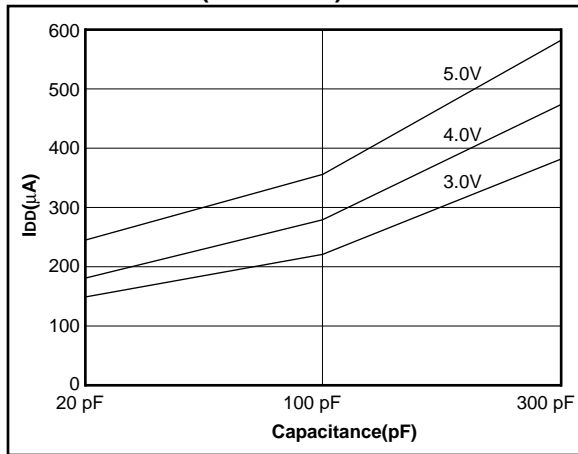
TABLE 12-1: RC OSCILLATOR FREQUENCIES

Cext	Rext	Average	
		Fosc @ 5V, 25°C	
22 pF	5k	4.12 MHz	± 1.4%
	10k	2.35 MHz	± 1.4%
	100k	268 kHz	± 1.1%
100 pF	3.3k	1.80 MHz	± 1.0%
	5k	1.27 MHz	± 1.0%
	10k	688 kHz	± 1.2%
	100k	77.2 kHz	± 1.0%
300 pF	3.3k	707 kHz	± 1.4%
	5k	501 kHz	± 1.2%
	10k	269 kHz	± 1.6%
	100k	28.3 kHz	± 1.1%

The percentage variation indicated here is part to part variation due to normal process distribution. The variation indicated is ±3 standard deviation from average value for V$_{DD}$ = 5V.

FIGURE 12-19: TRANSCONDUCTANCE(gm) OF HS OSCILLATOR vs. V$_{DD}$
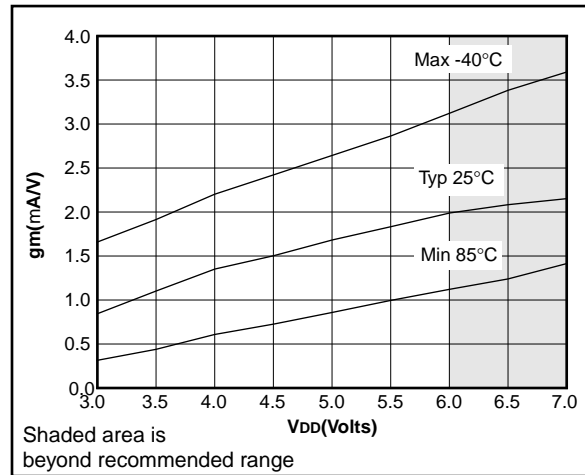
FIGURE 12-20: TRANSCONDUCTANCE(gm) OF LP OSCILLATOR vs. V$_{DD}$
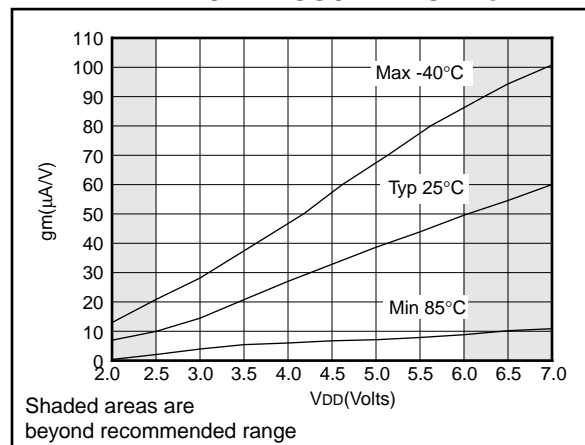
FIGURE 12-21: TRANSCONDUCTANCE(gm) OF XT OSCILLATOR vs. V$_{DD}$
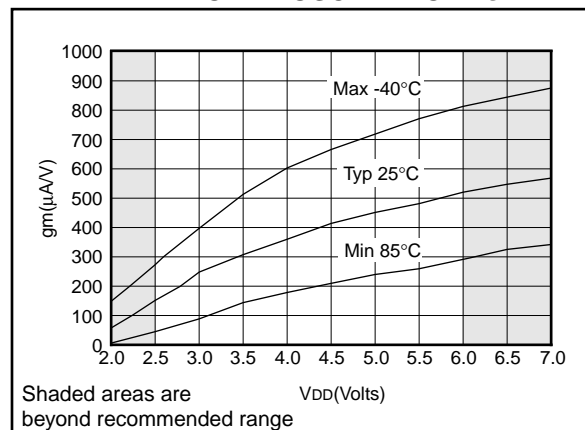
PIC16C71X

Applicable Devices | 710 | 71 | 711 | 715

13.1   DC Characteristics: PIC16C715-04 (Commercial, Industrial, Extended)
PIC16C715-10 (Commercial, Industrial, Extended)
PIC16C715-20 (Commercial, Industrial, Extended))

DC CHARACTERISTICS							Standard Operating Conditions (unless otherwise stated) Operating temperature 0˚C ≤ TA ≤ +70˚C (commercial) -40˚C ≤ TA ≤ +85˚C (industrial) -40˚C ≤ TA ≤ +125˚C (extended)
Param. No.	Characteristic	Sym	Min	Typ†	Max	Units	Conditions
D001 D001A	Supply Voltage	VDD	4.0 4.5	- -	5.5 5.5	V V	XT, RC and LP osc configuration HS osc configuration
D002*	RAM Data Retention Voltage (Note 1)	VDR	-	1.5	-	V	Device in SLEEP mode
D003	VDD start voltage to ensure internal Power-on Reset signal	VPOR	-	VSS	-	V	See section on Power-on Reset for details
D004*	VDD rise rate to ensure internal Power-on Reset signal	SVDD	0.05	-	-	V/ms	See section on Power-on Reset for details
D005	Brown-out Reset Voltage	BVDD	3.7	4.0	4.3	V	BODEN configuration bit is enabled
D010	Supply Current (Note 2)	IDD	-	2.7	5	mA	XT, RC osc configuration (PIC16C715-04) FOSC = 4 MHz, VDD = 5.5V (Note 4)
D013			-	13.5	30	mA	HS osc configuration (PIC16C715-20) FOSC = 20 MHz, VDD = 5.5V
D015	Brown-out Reset Current (Note 5)	∆IBOR	-	300*	500	µA	BOR enabled VDD = 5.0V
D020 D021 D021A D021B	Power-down Current (Note 3)	IPD	- - - -	10.5 1.5 1.5 1.5	42 21 24 30	µA µA µA µA	VDD = 4.0V, WDT enabled, -40°C to +85°C VDD = 4.0V, WDT disabled, -0°C to +70°C VDD = 4.0V, WDT disabled, -40°C to +85°C VDD = 4.0V, WDT disabled, -40°C to +125°C
D023	Brown-out Reset Current (Note 5)	∆IBOR	-	300*	500	µA	BOR enabled VDD = 5.0V

*   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.

Note 1:   This is the limit to which VDD can be lowered in SLEEP mode without losing RAM data.

2:   The supply current is mainly a function of the operating voltage and frequency. Other factors such as I/O pin loading and switching rate, oscillator type, internal code execution pattern, and temperature also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail to rail; all I/O pins tristated, pulled to VDD
MCLR = VDD; WDT enabled/disabled as specified.

3:   The power-down current in SLEEP mode does not depend on the oscillator type. Power-down current is measured with the part in SLEEP mode, with all I/O pins in hi-impedance state and tied to VDD and VSS.

4:   For RC osc configuration, current through Rext is not included. The current through the resistor can be estimated by the formula Ir = VDD/2Rext (mA) with Rext in kOhm.

5:   The ∆ current is the additional current consumed when this peripheral is enabled. This current should be added to the base IDD or IPD measurement.

© 1997 Microchip Technology Inc.   DS30272A-page 113

PIC16C71X

FIGURE 14-16: TYPICAL I$_{DD}$ vs. FREQUENCY (RC MODE @ 300 pF, 25°C)

FIGURE 14-17: MAXIMUM I$_{DD}$ vs. FREQUENCY (RC MODE @ 300 pF, -40°C TO 85°C)

PIC16C71X

FIGURE 15-3: CLKOUT AND I/O TIMING

Note: Refer to Figure 15-1 for load conditions.
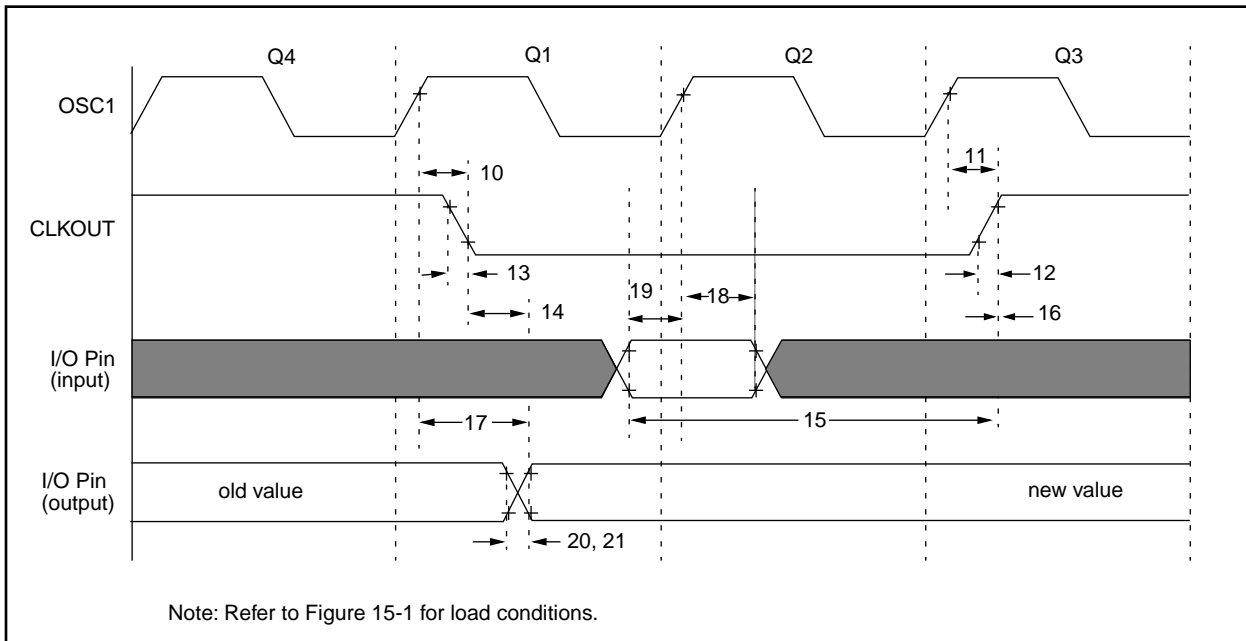
TABLE 15-3: CLKOUT AND I/O TIMING REQUIREMENTS

Parameter No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
10*	TosH2ckL	OSC1↑ to CLKOUT↓		—	15	30	ns	Note 1
11*	TosH2ckH	OSC1↑ to CLKOUT↑		—	15	30	ns	Note 1
12*	TckR	CLKOUT rise time		—	5	15	ns	Note 1
13*	TckF	CLKOUT fall time		—	5	15	ns	Note 1
14*	TckL2ioV	CLKOUT ↓ to Port out valid		—	—	0.5T$_{CY}$ + 20	ns	Note 1
15*	TioV2ckH	Port in valid before CLKOUT ↑		0.25T$_{CY}$ + 25	—	—	ns	Note 1
16*	TckH2ioI	Port in hold after CLKOUT ↑		0	—	—	ns	Note 1
17*	TosH2ioV	OSC1↑ (Q1 cycle) to Port out valid		—	—	80 - 100	ns	
18*	TosH2ioI	OSC1↑ (Q2 cycle) to Port input invalid (I/O in hold time)	PIC16**C**71	100	—	—	ns	
			PIC16**LC**71	200	—	—	ns	
19*	TioV2osH	Port input valid to OSC1↑ (I/O in setup time)		0	—	—	ns	
20*	TioR	Port output rise time	PIC16**C**71	—	10	25	ns	
			PIC16**LC**71	—	—	60	ns	
21*	TioF	Port output fall time	PIC16**C**71	—	10	25	ns	
			PIC16**LC**71	—	—	60	ns	
22††*	Tinp	INT pin high or low time		20	—	—	ns	
23††*	Trbp	RB7:RB4 change INT high or low time		20	—	—	ns	

* 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 asynchronous events not related to any internal clock edges.

Note 1: Measurements are taken in RC Mode where CLKOUT output is 4 x T$_{OSC}$.

LIST OF TABLES

NOTES: