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	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	20-SSOP (0.209", 5.30mm Width)
Supplier Device Package	20-SSOP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16lc711t-04i-ss

TABLE 3-1: PIC16C710/71/711/715 PINOUT DESCRIPTION

Pin Name	DIP Pin#	SSOP Pin#(4)	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	18	16	I	ST/CMOS(3)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	17	15	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	4	4	4	I/P	ST	Master clear (reset) input or programming voltage input. This pin is an active low reset to the device.
						PORTA is a bi-directional I/O port.
RA0/AN0	17	19	17	I/O	TTL	RA0 can also be analog input0
RA1/AN1	18	20	18	I/O	TTL	RA1 can also be analog input1
RA2/AN2	1	1	1	I/O	TTL	RA2 can also be analog input2
RA3/AN3/VREF	2	2	2	I/O	TTL	RA3 can also be analog input3 or analog reference voltage
RA4/T0CKI	3	3	3	I/O	ST	RA4 can also be the clock input to the Timer0 module. Output is open drain type.
						PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	6	7	6	I/O	TTL/ST(1)	RB0 can also be the external interrupt pin.
RB1	7	8	7	I/O	TTL	
RB2	8	9	8	I/O	TTL	
RB3	9	10	9	I/O	TTL	
RB4	10	11	10	I/O	TTL	Interrupt on change pin.
RB5	11	12	11	I/O	TTL	Interrupt on change pin.
RB6	12	13	12	I/O	TTL/ST(2)	Interrupt on change pin. Serial programming clock.
RB7	13	14	13	I/O	TTL/ST(2)	Interrupt on change pin. Serial programming data.
VSS	5	4, 6	5	P	—	Ground reference for logic and I/O pins.
VDD	14	15, 16	14	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in serial programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.
4: The PIC16C71 is not available in SSOP package.

4.2.2.4 PIE1 REGISTER

Applicable Devices	710	71	711	715

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

Note:	Bit PEIE (INTCON<6>) must be set to enable any peripheral interrupt.

FIGURE 4-10: PIE1 REGISTER (ADDRESS 8Ch)

U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0
—	ADIE	—	—	—	—	—	—
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: **ADIE:** A/D Converter Interrupt Enable bit
1 = Enables the A/D interrupt
0 = Disables the A/D interrupt

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

TABLE 5-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other resets
06h, 106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h, 186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
81h, 181h	OPTION	$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

6.0 TIMER0 MODULE

Applicable Devices	710	71	711	715

The Timer0 module timer/counter has the following features:

- 8-bit timer/counter
- Readable and writable
- 8-bit software programmable prescaler
- Internal or external clock select
- Interrupt on overflow from FFh to 00h
- Edge select for external clock

Figure 6-1 is a simplified block diagram of the Timer0 module.

Timer mode is selected by clearing bit T0CS (OPTION<5>). In timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If the TMR0 register is written, the increment is inhibited for the following two instruction cycles (Figure 6-2 and Figure 6-3). The user can work around this by writing an adjusted value to the TMR0 register.

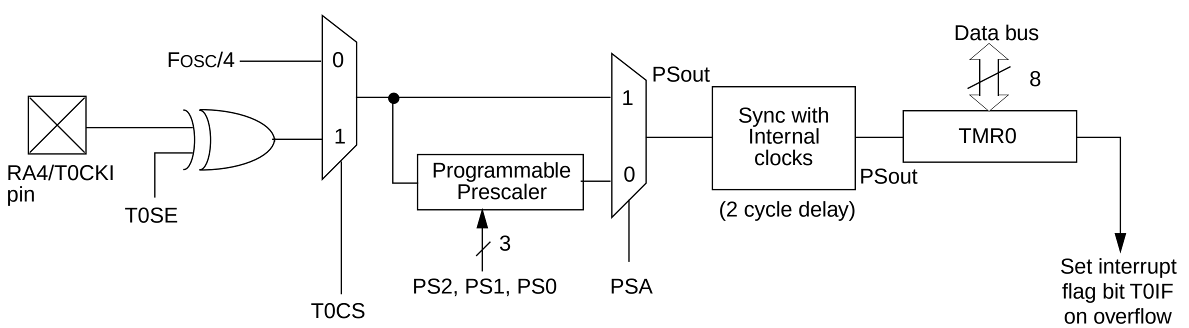
Counter mode is selected by setting bit T0CS (OPTION<5>). In counter mode, Timer0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit T0SE (OPTION<4>). Clearing bit T0SE selects the rising edge. Restrictions on the external clock input are discussed in detail in Section 6.2.

The prescaler is mutually exclusively shared between the Timer0 module and the Watchdog Timer. The prescaler assignment is controlled in software by control bit PSA (OPTION<3>). Clearing bit PSA will assign the prescaler to the Timer0 module. The prescaler is not readable or writable. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4, ..., 1:256 are selectable. Section 6.3 details the operation of the prescaler.

6.1 Timer0 Interrupt

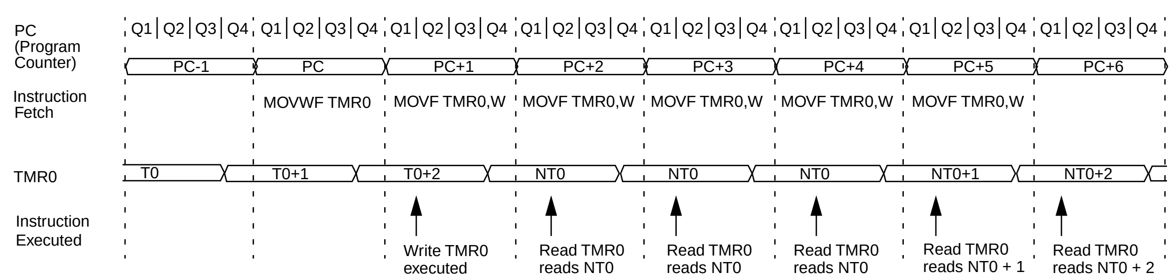
The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h. This overflow sets bit T0IF (INTCON<2>). The interrupt can be masked by clearing bit T0IE (INTCON<5>). Bit T0IF must be cleared in software by the Timer0 module interrupt service routine before re-enabling this interrupt. The TMR0 interrupt cannot awaken the processor from SLEEP since the timer is shut off during SLEEP. See Figure 6-4 for Timer0 interrupt timing.

FIGURE 6-1: TIMER0 BLOCK DIAGRAM



Note 1: T0CS, T0SE, PSA, PS2:PS0 (OPTION<5:0>).
2: The prescaler is shared with Watchdog Timer (refer to Figure 6-6 for detailed block diagram).

FIGURE 6-2: TIMER0 TIMING: INTERNAL CLOCK/NO PRESCALE



8.3 Reset

Applicable Devices	710	71	711	715

The PIC16CXX differentiates between various kinds of reset:

- Power-on Reset (POR)
- $\overline{MCLR}$ reset during normal operation
- $\overline{MCLR}$ reset during SLEEP
- WDT Reset (normal operation)
- Brown-out Reset (BOR) (PIC16C710/711/715)
- Parity Error Reset (PIC16C715)

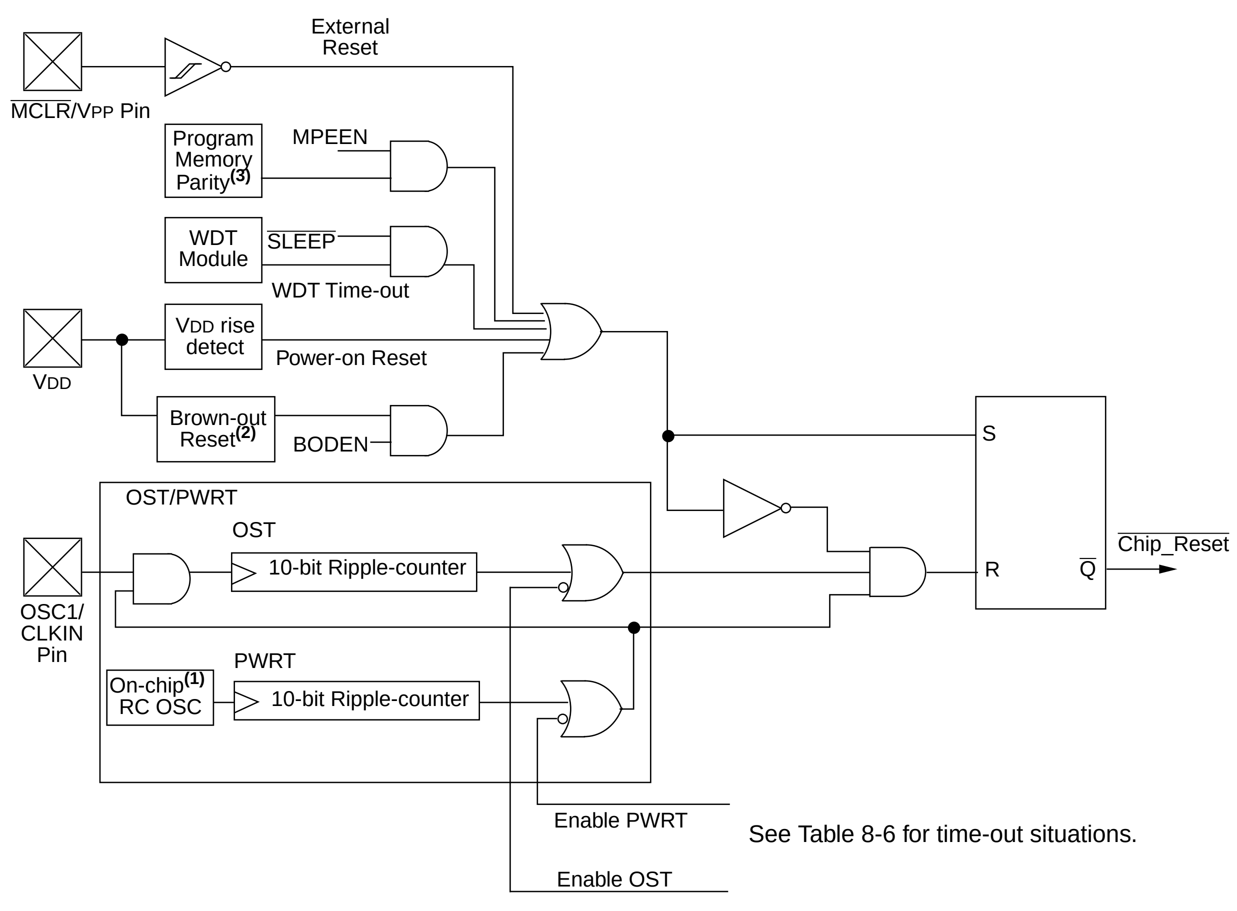
Some registers are not affected in any reset condition; their status is unknown on POR and unchanged in any other reset. Most other registers are reset to a “reset state” on Power-on Reset (POR), on the $\overline{MCLR}$ and WDT Reset, on $\overline{MCLR}$ reset during SLEEP, and Brown-out Reset (BOR). They are not affected by a WDT Wake-up, which is viewed as the resumption of normal operation. The $\overline{TO}$ and $\overline{PD}$ bits are set or cleared differently in different reset situations as indicated in Table 8-7, Table 8-8 and Table 8-9. These bits are used in software to determine the nature of the reset. See Table 8-10 and Table 8-11 for a full description of reset states of all registers.

A simplified block diagram of the on-chip reset circuit is shown in Figure 8-9.

The PIC16C710/711/715 have a $\overline{MCLR}$ noise filter in the $\overline{MCLR}$ reset path. The filter will detect and ignore small pulses.

It should be noted that a WDT Reset does not drive $\overline{MCLR}$ pin low.

FIGURE 8-9: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



Note 1: This is a separate oscillator from the RC oscillator of the CLKIN pin.
2: Brown-out Reset is implemented on the PIC16C710/711/715.
3: Parity Error Reset is implemented on the PIC16C715.

TABLE 8-12: INITIALIZATION CONDITIONS FOR ALL REGISTERS, PIC16C710/71/711

Register	Power-on Reset, Brown-out Reset(5)	$\overline{\text{MCLR}}$ Resets WDT Reset	Wake-up via WDT or Interrupt
W	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	N/A	N/A	N/A
TMR0	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	0000h	0000h	PC + 1(2)
STATUS	0001 1xxx	000q quuu(3)	uuuq quuu(3)
FSR	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	---x 0000	---u 0000	---u uuuu
PORTB	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCLATH	---0 0000	---0 0000	---u uuuu
INTCON	0000 000x	0000 000u	uuuu uuuu(1)
ADRES	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	00-0 0000	00-0 0000	uu-u uuuu
OPTION	1111 1111	1111 1111	uuuu uuuu
TRISA	---1 1111	---1 1111	---u uuuu
TRISB	1111 1111	1111 1111	uuuu uuuu
PCON(4)	---- --0u	---- --uu	---- --uu
ADCON1	---- --00	---- --00	---- --uu

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: See Table 8-10 for reset value for specific condition.
4: The PCON register is not implemented on the PIC16C71.
5: Brown-out reset is not implemented on the PIC16C71.

FIGURE 8-17: INTERRUPT LOGIC, PIC16C710, 71, 711

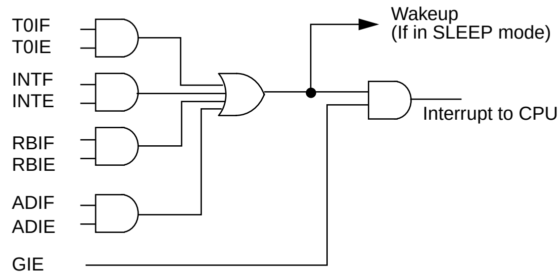
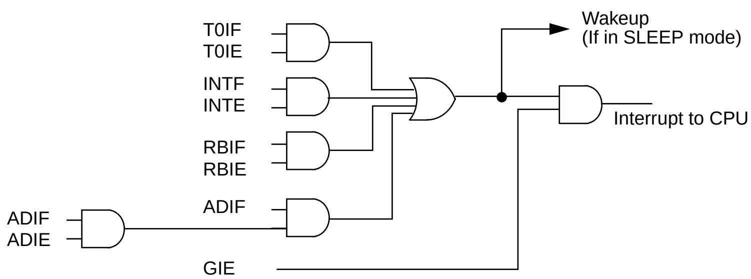


FIGURE 8-18: INTERRUPT LOGIC, PIC16C715



8.8 Power-down Mode (SLEEP)

Power-down mode is entered by executing a SLEEP instruction.

If enabled, the Watchdog Timer will be cleared but keeps running, the $\overline{\text{PD}}$ bit (STATUS<3>) is cleared, the $\overline{\text{TO}}$ (STATUS<4>) bit is set, and the oscillator driver is turned off. The I/O ports maintain the status they had, before the SLEEP instruction was executed (driving high, low, or hi-impedance).

For lowest current consumption in this mode, place all I/O pins at either V_{DD}, or V_{SS}, ensure no external circuitry is drawing current from the I/O pin, power-down the A/D, disable external clocks. Pull all I/O pins, that are hi-impedance inputs, high or low externally to avoid switching currents caused by floating inputs. The T0CKI input should also be at V_{DD} or V_{SS} for lowest current consumption. The contribution from on-chip pull-ups on PORTB should be considered.

The $\overline{\text{MCLR}}$ pin must be at a logic high level (V_{IHMC}).

8.8.1 WAKE-UP FROM SLEEP

The device can wake up from SLEEP through one of the following events:

1. External reset input on $\overline{\text{MCLR}}$ pin.
2. Watchdog Timer Wake-up (if WDT was enabled).
3. Interrupt from INT pin, RB port change, or some Peripheral Interrupts.

External $\overline{\text{MCLR}}$ Reset will cause a device reset. All other events are considered a continuation of program execution and cause a "wake-up". The $\overline{\text{TO}}$ and $\overline{\text{PD}}$ bits in the STATUS register can be used to determine the cause of device reset. The $\overline{\text{PD}}$ bit, which is set on power-up, is cleared when SLEEP is invoked. The $\overline{\text{TO}}$ bit is cleared if a WDT time-out occurred (and caused wake-up).

The following peripheral interrupts can wake the device from SLEEP:

1. TMR1 interrupt. Timer1 must be operating as an asynchronous counter.
2. A/D conversion (when A/D clock source is RC).

Other peripherals cannot generate interrupts since during SLEEP, no on-chip Q clocks are present.

When the SLEEP instruction is being executed, the next instruction (PC + 1) is pre-fetched. For the device to wake-up through an interrupt event, the corresponding interrupt enable bit must be set (enabled). Wake-up is regardless of the state of the GIE bit. If the GIE bit is clear (disabled), the device continues execution at the instruction after the SLEEP instruction. If the GIE bit is set (enabled), the device executes the instruction after the SLEEP instruction and then branches to the interrupt address (0004h). In cases where the execution of the instruction following SLEEP is not desirable, the user should have a NOP after the SLEEP instruction.

8.8.2 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 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 $\overline{\text{TO}}$ bit will not be set and $\overline{\text{PD}}$ 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 $\overline{\text{TO}}$ bit will be set and the $\overline{\text{PD}}$ 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 $\overline{\text{PD}}$ bit. If the $\overline{\text{PD}}$ 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.

TABLE 9-2: PIC16CXX INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	**f, d**	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	**f, d**	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	**f**	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	**-**	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	**f, d**	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	**f, d**	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	**f, d**	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	**f, d**	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	**f, d**	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	**f, d**	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	**f, d**	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	**f**	Move W to f	1	00	0000	lfff	ffff		
NOP	**-**	No Operation	1	00	0000	0xx0	0000		
RLF	**f, d**	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	**f, d**	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	**f, d**	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	**f, d**	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	**f, d**	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	**f, b**	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	**f, b**	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	**f, b**	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	**f, b**	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	**k**	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	**k**	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	**k**	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	**-**	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	**k**	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	**k**	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	**k**	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	**-**	Return from interrupt	2	00	0000	0000	1001		
RETLW	**k**	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	**-**	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	**-**	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	**k**	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	**k**	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

BCF — Bit Clear f

Syntax: [*label*] BCF f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: $0 \rightarrow$ (f<b>)

Status Affected: None

Encoding:

`01`	`00bb`	`bfff`	`ffff`

Description: Bit 'b' in register 'f' is cleared.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example

```
BCF      FLAG_REG, 7
```

Before Instruction
FLAG_REG = 0xC7
After Instruction
FLAG_REG = 0x47

BSF — Bit Set f

Syntax: [*label*] BSF f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: $1 \rightarrow$ (f<b>)

Status Affected: None

Encoding:

`01`	`01bb`	`bfff`	`ffff`

Description: Bit 'b' in register 'f' is set.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example

```
BSF      FLAG_REG,    7
```

Before Instruction
FLAG_REG = 0x0A
After Instruction
FLAG_REG = 0x8A

BTFSC — Bit Test, Skip if Clear

Syntax: [*label*] BTFSC f,b

Operands: $0 \le f \le 127$
$0 \le b \le 7$

Operation: skip if (f<b>) = 0

Status Affected: None

Encoding:

`01`	`10bb`	`bfff`	`ffff`

Description: If bit 'b' in register 'f' is '1' then the next instruction is executed.
If bit 'b', in register 'f', is '0' then the next instruction is discarded, and a NOP is executed instead, making this a $2T_{CY}$ instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	NOP

If Skip: (2nd Cycle)

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP

Example

```
HERE     BTFSC   FLAG,1
FALSE    GOTO    PROCESS_CODE
TRUE     •
         •
         •
```

Before Instruction
PC = address `HERE`
After Instruction
if FLAG<1> = 0,
PC = address `TRUE`
if FLAG<1>=1,
PC = address `FALSE`

BTFSS — Bit Test f, Skip if Set

Syntax: [*label*] BTFSS f,b

Operands: $0 \le f \le 127$
$0 \le b < 7$

Operation: skip if (f<b>) = 1

Status Affected: None

Encoding:

01	11bb	bfff	ffff

Description: If bit 'b' in register 'f' is '0' then the next instruction is executed.
If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2TCY instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process data	NOP

If Skip: (2nd Cycle)

Q1	Q2	Q3	Q4
NOP	NOP	NOP	NOP

Example

```
HERE    BTFSC  FLAG,1
FALSE   GOTO   PROCESS_CODE
TRUE    •
        •
        •
```

Before Instruction
PC = address HERE
After Instruction
if FLAG<1> = 0,
PC = address FALSE
if FLAG<1> = 1,
PC = address TRUE

CALL — Call Subroutine

Syntax: [*label*] CALL k

Operands: $0 \le k \le 2047$

Operation: (PC)+ 1→ TOS,
k → PC<10:0>,
(PCLATH<4:3>) → PC<12:11>

Status Affected: None

Encoding:

10	0kkk	kkkk	kkkk

Description: Call Subroutine. First, return address (PC+1) is pushed onto the stack. The eleven bit immediate address is loaded into PC bits <10:0>. The upper bits of the PC are loaded from PCLATH. CALL is a two cycle instruction.

Words: 1

Cycles: 2

Q Cycle Activity:

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

Example

```
HERE    CALL   THERE
```

Before Instruction
PC = Address HERE
After Instruction
PC = Address THERE
TOS = Address HERE+1

RETLW — Return with Literal in W

Syntax:	[*label*] RETLW k
Operands:	$0 \le k \le 255$
Operation:	$k \rightarrow (W)$; $TOS \rightarrow PC$
Status Affected:	None
Encoding:	`11` `01xx` `kkkk` `kkkk`
Description:	The W register is loaded with the eight bit literal 'k'. The program counter is loaded from the top of the stack (the return address). This is a two cycle instruction.
Words:	1
Cycles:	2

Q Cycle Activity:

	Q1	Q2	Q3	Q4
1st Cycle	Decode	Read literal 'k'	NOP	Write to W, Pop from the Stack
2nd Cycle	NOP	NOP	NOP	NOP

Example

```
      CALL TABLE  ;W contains table
                  ;offset value
      •             ;W now has table value
      •
      •
TABLE ADDWF PC   ;W = offset
      RETLW k1   ;Begin table
      RETLW k2   ;
      •
      •
      •
      RETLW kn   ; End of table
```

Before Instruction
W = 0x07

After Instruction
W = value of k8

RETURN — Return from Subroutine

Syntax:	[*label*] RETURN
Operands:	None
Operation:	$TOS \rightarrow PC$
Status Affected:	None
Encoding:	`00` `0000` `0000` `1000`
Description:	Return from subroutine. The stack is POPed and the top of the stack (TOS) is loaded into the program counter. This is a two cycle instruction.
Words:	1
Cycles:	2

Q Cycle Activity:

	Q1	Q2	Q3	Q4
1st Cycle	Decode	NOP	NOP	Pop from the Stack
2nd Cycle	NOP	NOP	NOP	NOP

Example

```
RETURN
```

After Interrupt
PC = TOS

SLEEP

Syntax: [*label*] SLEEP

Operands: None

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

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

Encoding:

00	0000	0110	0011

Description: The power-down status bit, $\overline{PD}$ is cleared. Time-out status bit, $\overline{TO}$ is set. Watchdog Timer and its prescaler are cleared.
The processor is put into SLEEP mode with the oscillator stopped. See Section 8.8 for more details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	NOP	NOP	Go to Sleep

Example: SLEEP

SUBLW Subtract W from Literal

Syntax: [*label*] SUBLW k

Operands: $0 \leq k \leq 255$

Operation: k - (W) → (W)

Status Affected: C, DC, Z

Encoding:

11	110x	kkkk	kkkk

Description: The W register is subtracted (2's complement method) from 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 1: SUBLW 0x02

Before Instruction
W = 1
C = ?
Z = ?

After Instruction
W = 1
C = 1; result is positive
Z = 0

Example 2: Before Instruction
W = 2
C = ?
Z = ?

After Instruction
W = 0
C = 1; result is zero
Z = 1

Example 3: Before Instruction
W = 3
C = ?
Z = ?

After Instruction
W = 0xFF
C = 0; result is negative
Z = 0

MPASM has the following features to assist in developing software for specific use applications.

- Provides translation of Assembler source code to object code for all Microchip microcontrollers.
- Macro assembly capability.
- Produces all the files (Object, Listing, Symbol, and special) required for symbolic debug with Microchip's emulator systems.
- Supports Hex (default), Decimal and Octal source and listing formats.

MPASM provides a rich directive language to support programming of the PIC16/17. Directives are helpful in making the development of your assemble source code shorter and more maintainable.

10.11 Software Simulator (MPLAB-SIM)

The MPLAB-SIM Software Simulator allows code development in a PC host environment. It allows the user to simulate the PIC16/17 series microcontrollers on an instruction level. On any given instruction, the user may examine or modify any of the data areas or provide external stimulus to any of the pins. The input/output radix can be set by the user and the execution can be performed in; single step, execute until break, or in a trace mode.

MPLAB-SIM fully supports symbolic debugging using MPLAB-C and MPASM. The Software Simulator offers the low cost flexibility to develop and debug code outside of the laboratory environment making it an excellent multi-project software development tool.

10.12 C Compiler (MPLAB-C)

The MPLAB-C Code Development System is a complete 'C' compiler and integrated development environment for Microchip's PIC16/17 family of microcontrollers. The compiler provides powerful integration capabilities and ease of use not found with other compilers.

For easier source level debugging, the compiler provides symbol information that is compatible with the MPLAB IDE memory display.

10.13 Fuzzy Logic Development System (*fuzzy*TECH-MP)

*fuzzy*TECH-MP fuzzy logic development tool is available in two versions - a low cost introductory version, MP Explorer, for designers to gain a comprehensive working knowledge of fuzzy logic system design; and a full-featured version, *fuzzy*TECH-MP, edition for implementing more complex systems.

Both versions include Microchip's *fuzzy*LAB™ demonstration board for hands-on experience with fuzzy logic systems implementation.

10.14 MP-DriveWay™ – Application Code Generator

MP-DriveWay is an easy-to-use Windows-based Application Code Generator. With MP-DriveWay you can visually configure all the peripherals in a PIC16/17 device and, with a click of the mouse, generate all the initialization and many functional code modules in C language. The output is fully compatible with Microchip's MPLAB-C C compiler. The code produced is highly modular and allows easy integration of your own code. MP-DriveWay is intelligent enough to maintain your code through subsequent code generation.

10.15 SEEVAL® Evaluation and Programming System

The SEEVAL SEEPROM Designer's Kit supports all Microchip 2-wire and 3-wire Serial EEPROMs. The kit includes everything necessary to read, write, erase or program special features of any Microchip SEEPROM product including Smart Serials™ and secure serials. The Total Endurance™ Disk is included to aid in trade-off analysis and reliability calculations. The total kit can significantly reduce time-to-market and result in an optimized system.

10.16 KEELOQ® Evaluation and Programming Tools

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

Applicable Devices	710	71	711	715

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) Operating voltage VDD range as described in DC spec Section 13.1 and Section 13.2.				
Param No.	**Characteristic**	**Sym**	**Min**	**Typ †**	**Max**	**Units**	**Conditions**
	Output High Voltage						
D090	I/O ports (Note 3)	VOH	VDD - 0.7	-	-	V	IOH = -3.0 mA, VDD = 4.5V, -40°C to +85°C
D090A			VDD - 0.7	-	-	V	IOH = -2.5 mA, VDD = 4.5V, -40°C to +125°C
D092	OSC2/CLKOUT (RC osc config)		VDD - 0.7	-	-	V	IOH = -1.3 mA, VDD = 4.5V, -40°C to +85°C
D092A			VDD - 0.7	-	-	V	IOH = -1.0 mA, VDD = 4.5V, -40°C to +125°C
	Capacitive Loading Specs on Output Pins						
D100	OSC2 pin	COSC2	-	-	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)	CIO	-	-	50	pF	

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

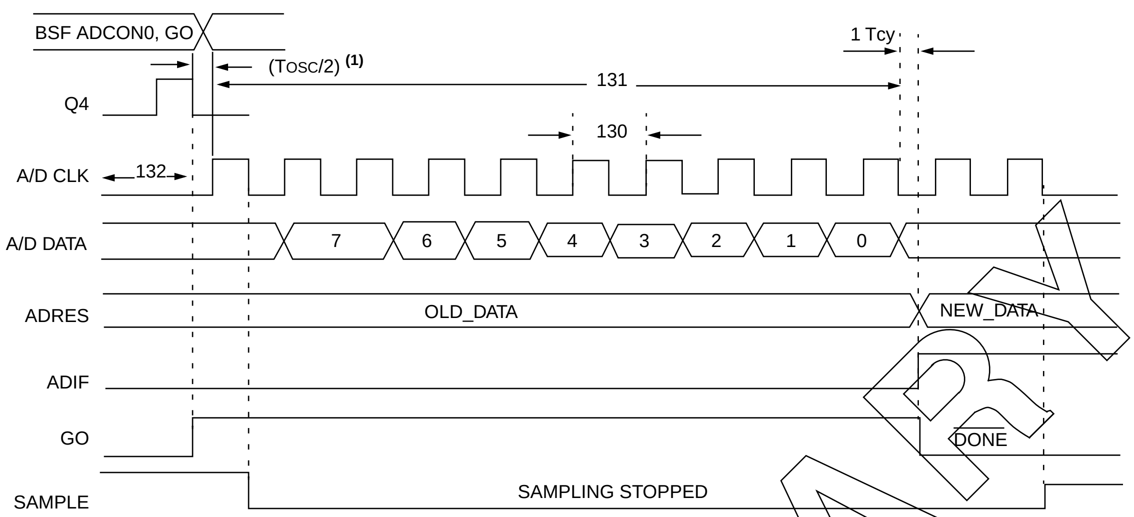
Note 1: In RC oscillator configuration, the OSC1/CLKIN pin is a Schmitt Trigger input. It is not recommended that the PIC16C7X be driven with external clock in RC mode.

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

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

Applicable Devices	710	71	711	715

FIGURE 13-7: A/D CONVERSION TIMING



Note 1: If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

TABLE 13-8: A/D CONVERSION REQUIREMENTS

Parameter No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
130	TAD	A/D clock period	1.6 2.0		— —	µs µs	VREF ≥ 3.0V VREF full range
130	TAD	A/D Internal RC Oscillator source					ADCS1:ADCS0 = 11 (RC oscillator source)
			3.0	6.0	9.0	µs	PIC16LC715, VDD = 3.0V
			2.0	4.0	6.0	µs	PIC16C715
131	TCNV	Conversion time (not including S/H time). Note 1	—	9.5TAD	—	—	
132	TACQ	Acquisition time	Note 2	20	—	µs	

* 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: ADRES register may be read on the following TCY cycle.

Applicable Devices	710	71	711	715

FIGURE 14-12: TYPICAL IDD vs. FREQUENCY (RC MODE @ 22 pF, 25°C)

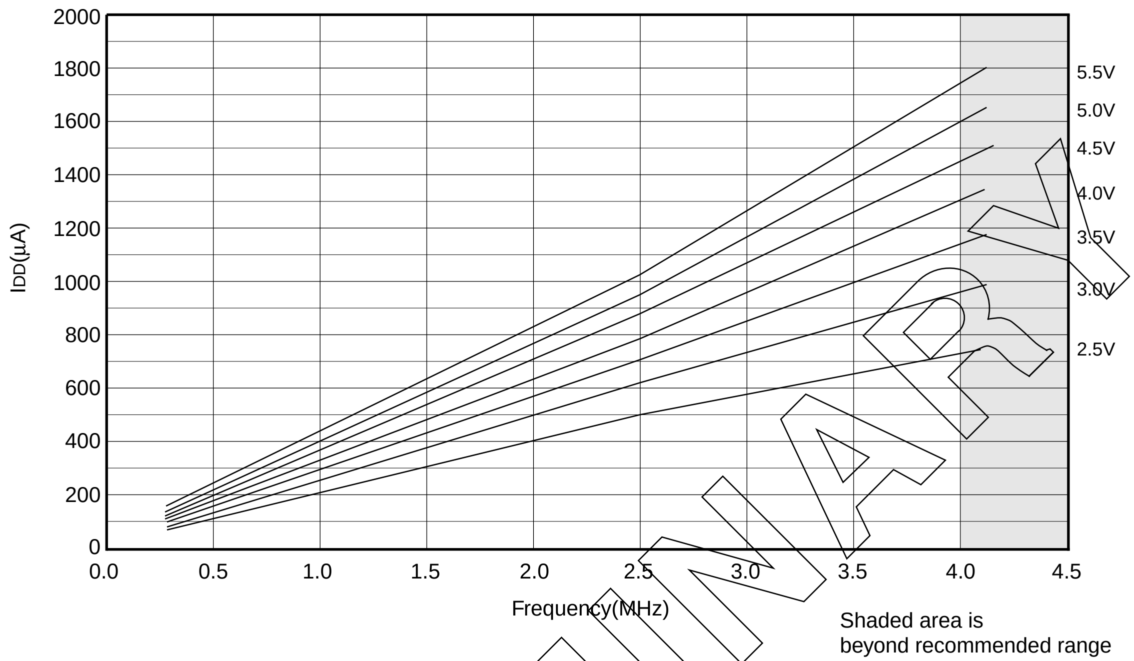
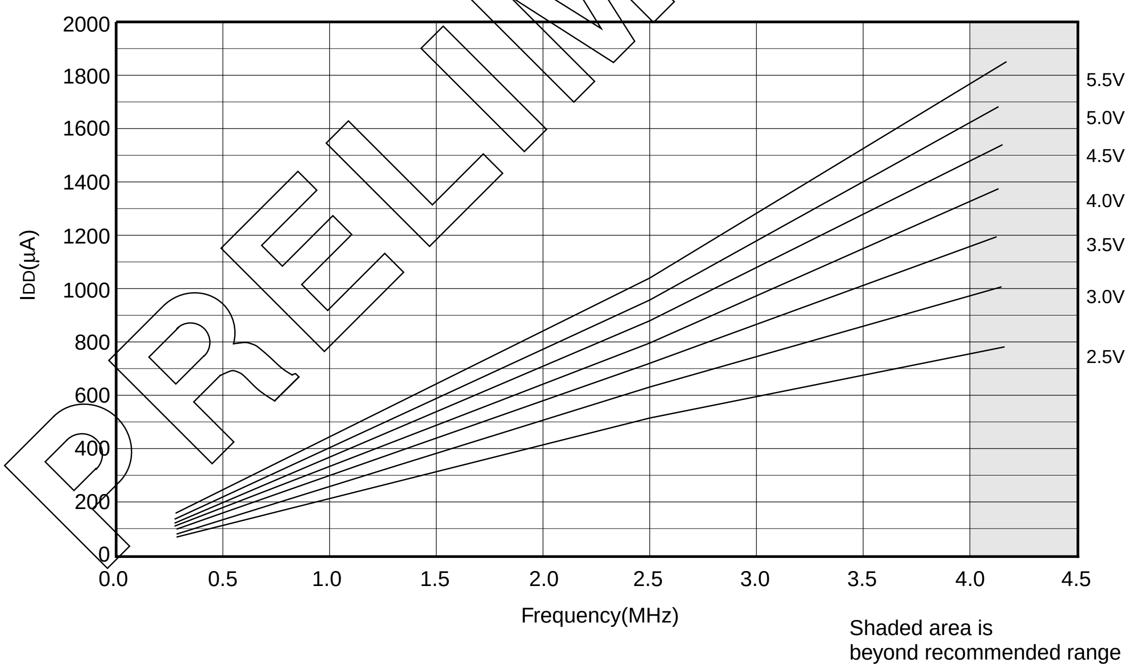


FIGURE 14-13: MAXIMUM IDD vs. FREQUENCY (RC MODE @ 22 pF, -40°C TO 85°C)



Applicable Devices	710	71	711	715

15.0 ELECTRICAL CHARACTERISTICS FOR PIC16C71

Absolute Maximum Ratings †

Ambient temperature under bias .... -55 to +125˚C
Storage temperature .... -65˚C to +150˚C
Voltage on any pin with respect to V_{SS} (except V_{DD}, $\overline{MCLR}$, and RA4) .... -0.3V to (V_{DD} + 0.3V)
Voltage on V_{DD} with respect to V_{SS} .... -0.3 to +7.5V
Voltage on $\overline{MCLR}$ with respect to V_{SS} (Note 2) .... 0 to +14V
Voltage on RA4 with respect to Vss .... 0 to +14V
Total power dissipation (Note 1) .... 800 mW
Maximum current out of V_{SS} pin .... 150 mA
Maximum current into V_{DD} pin .... 100 mA
Input clamp current, I_{IK} (V_I < 0 or V_I > V_{DD}) .... ± 20 mA
Output clamp current, I_{OK} (V_O < 0 or V_O > V_{DD}) .... ± 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 = V_{DD} x {I_{DD} - $\sum$ I_{OH}} + $\sum$ {(V_{DD}-V_{OH}) x I_{OH}} + $\sum$(V_{OI} x I_{OL})

Note 2: Voltage spikes below V_{SS} at the $\overline{MCLR}$ 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 $\overline{MCLR}$ pin rather than pulling this pin directly to V_{SS}.

† 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.

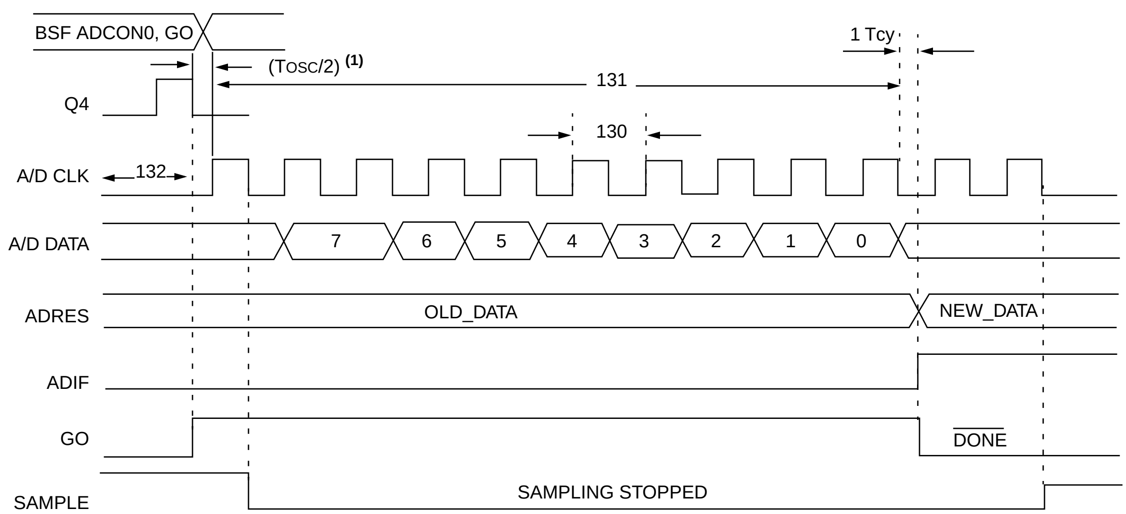
TABLE 15-1: CROSS REFERENCE OF DEVICE SPECS FOR OSCILLATOR CONFIGURATIONS AND FREQUENCIES OF OPERATION (COMMERCIAL DEVICES)

OSC	PIC16C71-04	PIC16C71-20	PIC16LC71-04	JW Devices
RC	V_{DD}: 4.0V to 6.0V I_{DD}: 3.3 mA max. at 5.5V I_{PD}: 14 µA max. at 4V Freq:4 MHz max.	V_{DD}: 4.5V to 5.5V I_{DD}: 1.8 mA typ. at 5.5V I_{PD}: 1.0 µA typ. at 4V Freq: 4 MHz max.	V_{DD}: 3.0V to 6.0V I_{DD}: 1.4 mA typ. at 3.0V I_{PD}: 0.6 µA typ. at 3V Freq: 4 MHz max.	V_{DD}: 4.0V to 6.0V I_{DD}: 3.3 mA max. at 5.5V I_{PD}: 14 µA max. at 4V Freq:4 MHz max.
XT	V_{DD}: 4.0V to 6.0V I_{DD}: 3.3 mA max. at 5.5V I_{PD}: 14 µA max. at 4V Freq: 4 MHz max.	V_{DD}: 4.5V to 5.5V I_{DD}: 1.8 mA typ. at 5.5V I_{PD}: 1.0 µA typ. at 4V Freq: 4 MHz max.	V_{DD}: 3.0V to 6.0V I_{DD}: 1.4 mA typ. at 3.0V I_{PD}: 0.6 µA typ. at 3V Freq: 4 MHz max.	V_{DD}: 4.0V to 6.0V I_{DD}: 3.3 mA max. at 5.5V I_{PD}: 14 µA max. at 4V Freq: 4 MHz max.
HS	V_{DD}: 4.5V to 5.5V I_{DD}: 13.5 mA typ. at 5.5V I_{PD}: 1.0 µA typ. at 4.5V Freq: 4 MHz max.	V_{DD}: 4.5V to 5.5V I_{DD}: 30 mA max. at 5.5V I_{PD}: 1.0 µA typ. at 4.5V Freq: 20 MHz max.	Not recommended for use in HS mode	V_{DD}: 4.5V to 5.5V I_{DD}: 30 mA max. at 5.5V I_{PD}: 1.0 µA typ. at 4.5V Freq: 20 MHz max.
LP	V_{DD}: 4.0V to 6.0V I_{DD}: 15 µA typ. at 32 kHz, 4.0V I_{PD}: 0.6 µA typ. at 4.0V Freq: 200 kHz max.	Not recommended for use in LP mode	V_{DD}: 3.0V to 6.0V I_{DD}: 32 µA max. at 32 kHz, 3.0V I_{PD}: 9 µA max. at 3.0V Freq: 200 kHz max.	V_{DD}: 3.0V to 6.0V I_{DD}: 32 µA max. at 32 kHz, 3.0V I_{PD}: 9 µA max. at 3.0V 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.

Applicable Devices	710	71	711	715

FIGURE 15-6: A/D CONVERSION TIMING



Note 1: If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

TABLE 15-7: A/D CONVERSION REQUIREMENTS

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
130	TAD	A/D clock period	PIC16**C**71	2.0	—	—	μs	TOSC based, VREF ≥ 3.0V
			PIC16**LC**71	2.0	—	—	μs	TOSC based, VREF full range
			PIC16**C**71	2.0	4.0	6.0	μs	A/D RC Mode
			PIC16**LC**71	3.0	6.0	9.0	μs	A/D RC Mode
131	TCNV	Conversion time (not including S/H time) (Note 1)		—	9.5	—	TAD	
132	TACQ	Acquisition time		Note 2	20	—	μs	
				5*	—	—	μs	The minimum time is the amplifier settling time. This may be used if the "new" input voltage has not changed by more than 1 LSb (i.e., 19.5 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).
134	TGO	Q4 to A/D clock start		—	Tosc/2§	—	—	If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.
135	TSWC	Switching from convert → sample time		1.5§	—	—	TAD	

* 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 specifications ensured by design.

Note 1: ADRES register may be read on the following TCY cycle.

2: See Section 7.1 for min conditions.

NOTES: